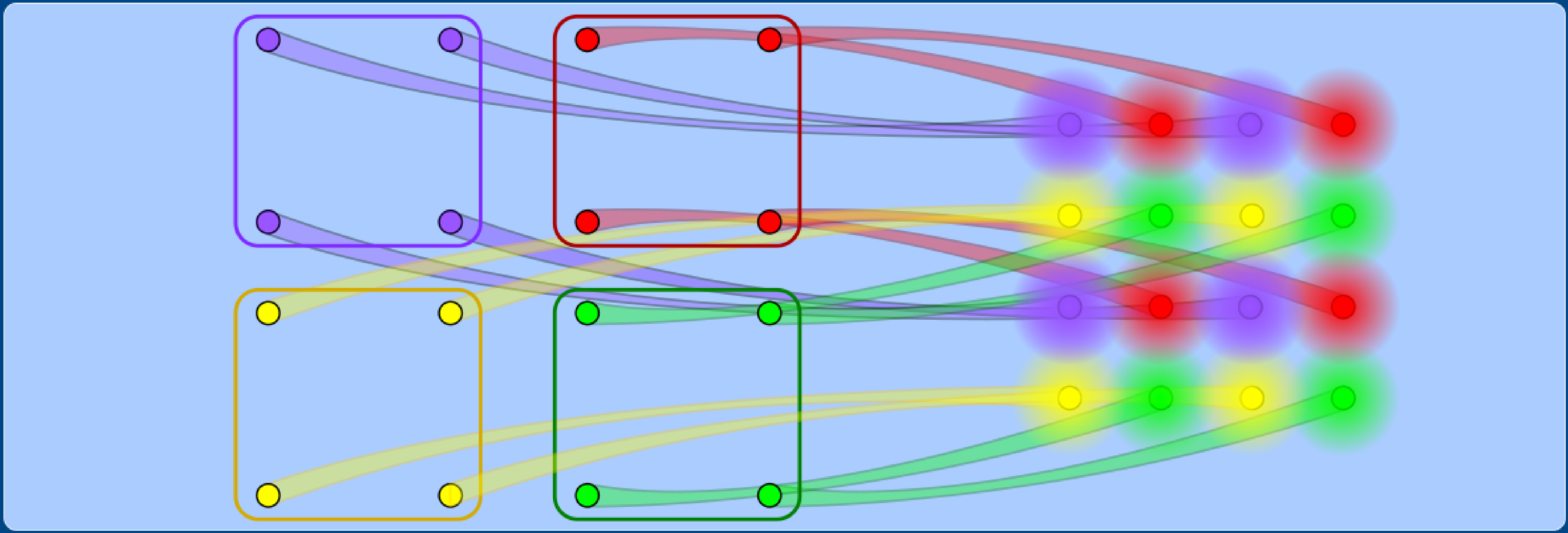
IFIC INSTITUT DE FÍSICA CORPUSCULAR | VNIVERSITAT DE VALÈNCIA

Machine Learning Trivializing Flows



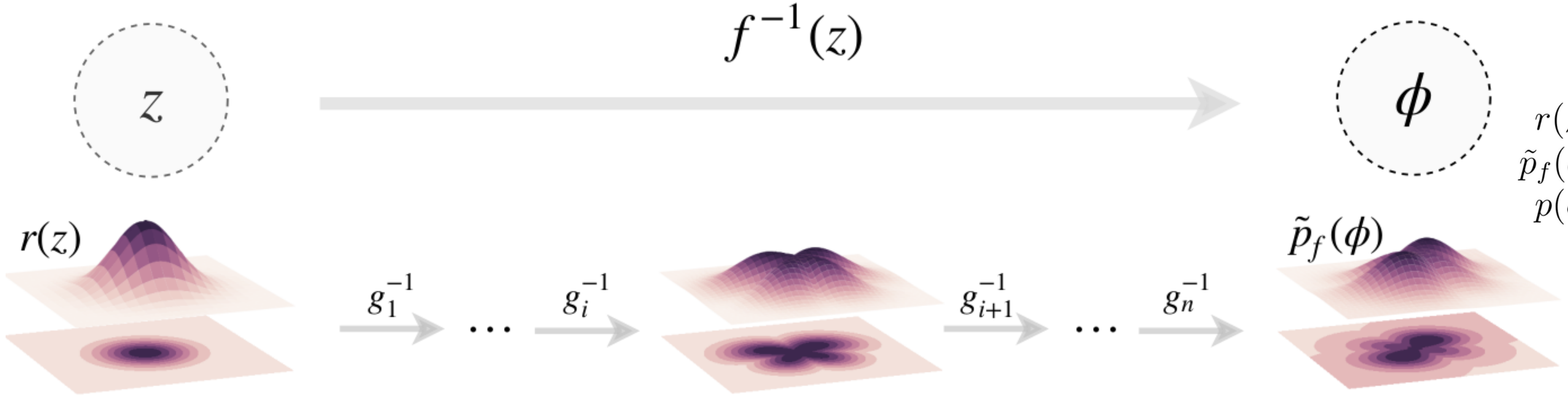
2023 LATTICE

David Albandea

Luigi Del Debbio
Richard Kenway
Joe Marsh Rossney

David Albandea
Pilar Hernández
Alberto Ramos

Normalizing flows



(a) Normalizing flow between prior and output distributions

M. S. Albergo, G. Kanwar and P. E. Shanahan, Phys. Rev. D 100, 034515 (2019), 1904.12072

$r(z)$: input distribution
$\tilde{p}_f(\phi)$: output distribution
$p(\phi)$: target distribution

⇨ *f(z)* is a network trained to minimize the Kullbach-Leibler divergence:

$$D_{\mathrm{KL}}(\tilde{p}_f \,||\, p) = \int \mathcal{D}\phi \, \tilde{p}_f(\phi) \log \frac{\tilde{p}_f(\phi)}{p(\phi)}$$

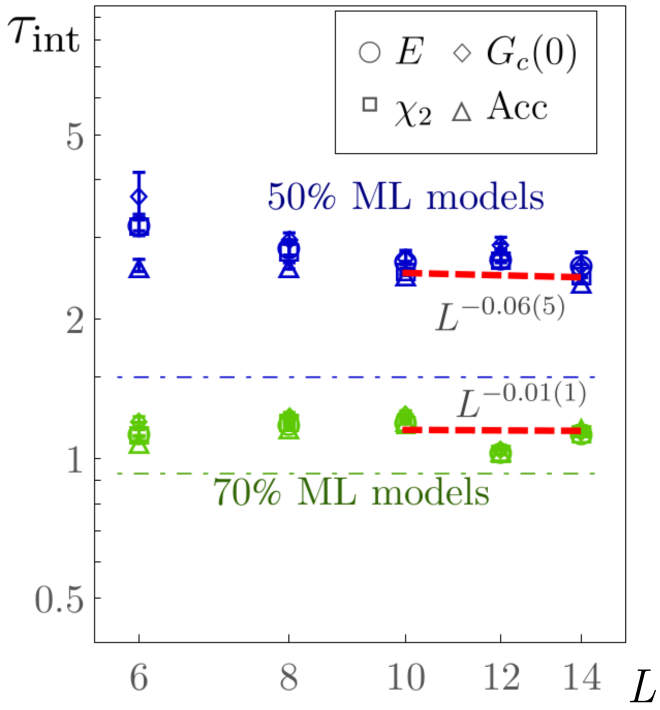
☆ $D_{\mathrm{KL}}(\tilde{p}_f \,||\, p) \geq 0$

☆ $D_{\mathrm{KL}}(\tilde{p}_f \,||\, p) = 0 \iff \tilde{p}_f = p$ ~ Trivializing map

⇨ Once *f* is trained, build a Markov chain with Metropoils-Hastings reweighting

Exploding training costs

M. S. Albergo, G. Kanwar and P. E. Shanahan,
Phys. Rev. D 100, 034515 (2019), 1904.12072



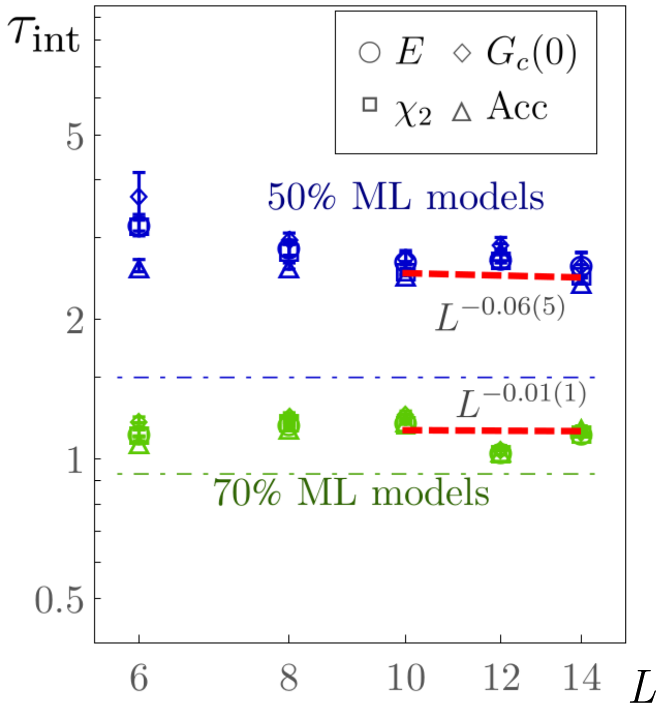
For equal acceptance, autocorrelation times do not scale towards the continuum

vs HMC: $\sim\xi^{2}$

Exploding training costs

Total cost = configuration production cost + network training cost

M. S. Albergo, G. Kanwar and P. E. Shanahan,
Phys. Rev. D 100, 034515 (2019), 1904.12072



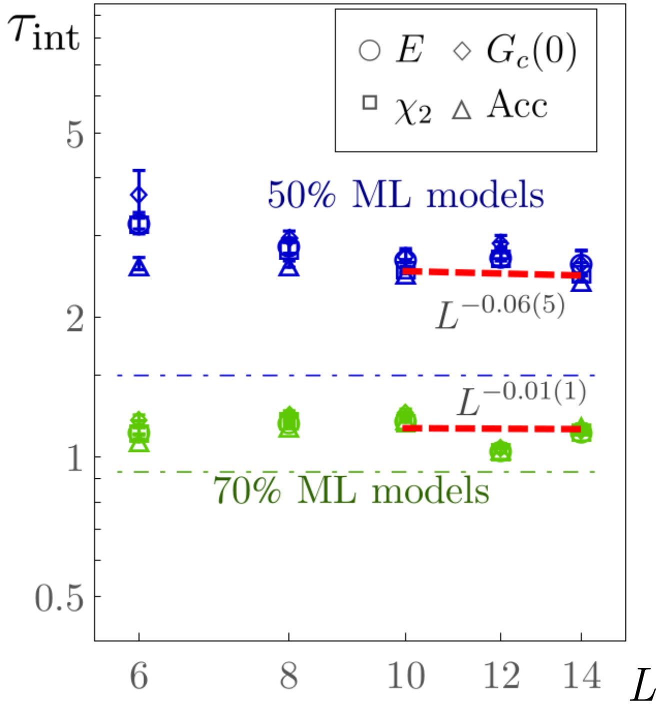
For equal acceptance, autocorrelation times do not scale towards the continuum

vs HMC: $\sim\xi^{2}$

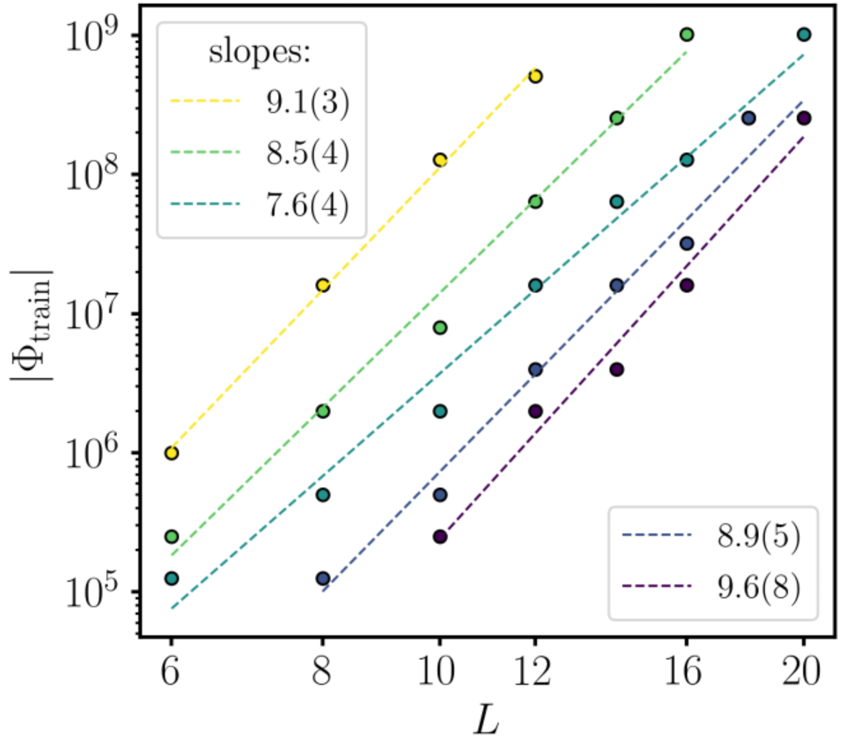
Exploding training costs

Total cost = configuration production cost + network training cost

M. S. Albergo, G. Kanwar and P. E. Shanahan, Phys. Rev. D 100, 034515 (2019), 1904.12072



Luigi Del Debbio, Joe Marsh Rossney, and Michael Wilson Phys. Rev. D 104, 094507



For equal acceptance, autocorrelation times do not scale towards the continuum

vs HMC: $\sim\xi^2$

Training costs to achieve equal acceptance explode towards the continuum as $\sim\xi^8$

Can we benefit from normalizing flows keeping training costs low?

Learning trivializing flows

Idea: use the normalizing flow f to **help** HMC sampling

$$Z = \int D\phi \, e^{-S(\phi)} \xrightarrow{\tilde{\phi} = f(\phi)} \int D\tilde{\phi} \, e^{-S(f^{-1}(\tilde{\phi})) + \log \det J[f]} \equiv \int D\tilde{\phi} \, e^{-\tilde{S}(\tilde{\phi})}$$

⇒ $\tilde{S}$ might be easier to sample from using HMC

↳ lower autocorrelation times!

Learning trivializing flows

Idea: use the normalizing flow f to **help** HMC sampling

$$Z = \int D\phi \, e^{-S(\phi)} \xrightarrow{\tilde{\phi} = f(\phi)} \int D\tilde{\phi} \, e^{-S(f^{-1}(\tilde{\phi})) + \log \det J[f]} \equiv \int D\tilde{\phi} \, e^{-\tilde{S}(\tilde{\phi})}$$

$\Rightarrow$ $\tilde{S}$ might be easier to sample from using HMC

↳ lower autocorrelation times!

The algorithm

1. Train the network f minimizing the KL divergence.
2. Use HMC to build a Markov chain following $\tilde{p} = e^{-\tilde{S}(\tilde{\phi})}$

$$\{\tilde{\phi}_1, \, \tilde{\phi}_2, \, \tilde{\phi}_3, \, \ldots, \, \tilde{\phi}_N\} \sim e^{-\tilde{S}(\tilde{\phi})}$$

3. Apply f^{-1} to the Markov chain to obtain configurations following $p(\phi) = e^{-S(\phi)}$

$$\{f^{-1}(\tilde{\phi}_1), \, f^{-1}(\tilde{\phi}_2), \, f^{-1}(\tilde{\phi}_3), \, \ldots, \, f^{-1}(\tilde{\phi}_N)\} = \{\phi_1, \, \phi_2, \, \phi_3, \, \ldots, \, \phi_N\} \sim e^{-S(\phi)}$$

$\Rightarrow$ The acceptance of HMC with the new action $\tilde{S}$ **does not depend on** f!

Learning trivializing flows

☆ Lüscher: an exact trivializing flow is not known, but can be constructed via power series (Wilson flow) Lüscher, M. Trivializing Maps, the Wilson Flow and the HMC Algorithm. Commun. Math. Phys. 293, 899 (2010)

⇒ It was not good enough to improve autocorrelation scaling towards the continuum on a CP(N) theory G. P. Engel, S. Schaefer, Testing trivializing maps in the Hybrid Monte Carlo algorithm, Comput.Phys.Commun. 182 (2011) 2107-2114. See also S. Bacchio et al. Phys.Rev.D 107 (2023) 5

⇒ Can normalizing flows be helpful as trivializing flows for HMC?

Xiao-Yong Jin, Neural Network Field Transformation and Its Application in HMC, PoS LATTICE2021 (2022) 600. Also X. Jin Thu 2:30PM

S. Foreman *et al.*, HMC with Normalizing Flows, PoS LATTICE2021 (2022) 073. Also Mon 1:50PM

The algorithm

1. Train the network f minimizing the KL divergence.
2. Use HMC to build a Markov chain following $\tilde{p} = e^{-\tilde{S}(\tilde{\phi})}$

$$\{\tilde{\phi}_1,\ \tilde{\phi}_2,\ \tilde{\phi}_3,\ \ldots,\ \tilde{\phi}_N\} \sim e^{-\tilde{S}(\tilde{\phi})}$$

3. Apply f^{-1} to the Markov chain to obtain configurations following $p(\phi) = e^{-S(\phi)}$

$$\{f^{-1}(\tilde{\phi}_1),\ f^{-1}(\tilde{\phi}_2),\ f^{-1}(\tilde{\phi}_3),\ \ldots,\ f^{-1}(\tilde{\phi}_N)\} = \{\phi_1,\ \phi_2,\ \phi_3,\ \ldots,\ \phi_N\} \sim e^{-S(\phi)}$$

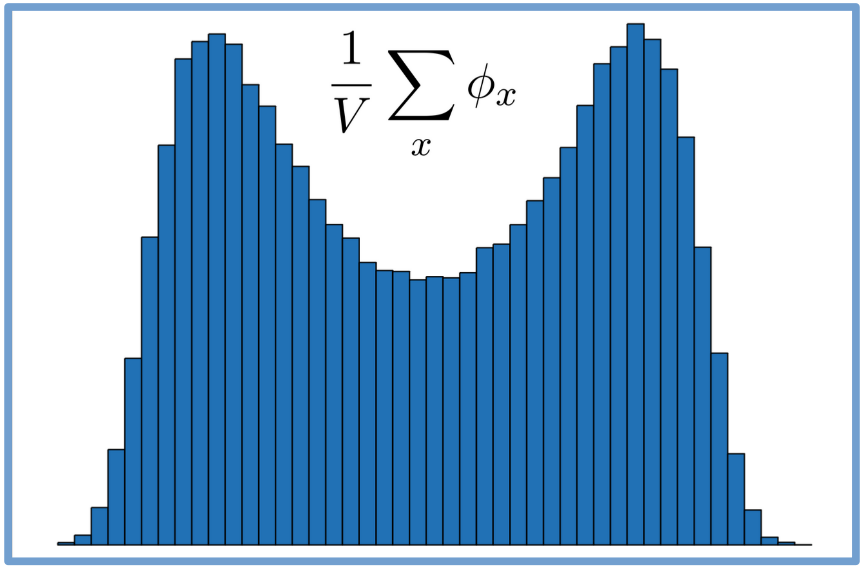
⇒ The acceptance of HMC with the new action $\tilde{S}$ **does not depend on** f!

The model

We study a ϕ^4 theory in 2 dimensions

$$S(\phi) = \sum_x \left[-\beta \sum_{\mu=1}^{2} \phi_{x+\mu}\phi_x + \phi_x^2 + \lambda(\phi_x^2 - 1)^2 \right]$$

- $\mathbb{Z}_2$ symmetry: action invariant under $\phi \to -\phi$
- Bimodal probability density
- Non-trivial correlation length ξ
  - HMC scaling: $\tau_{\text{int}} \propto \xi^2$

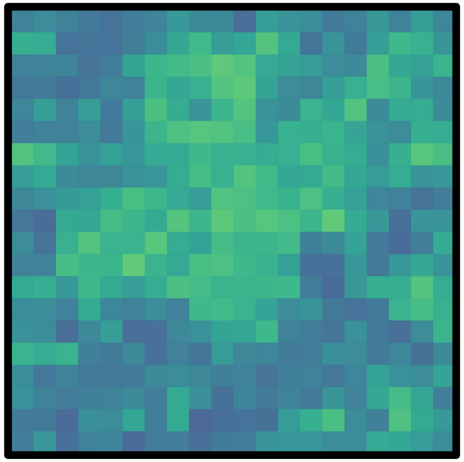


Keeping training costs low

Total cost ≈ configuration production cost

☆ Translational symmetry ⇨ use convolutional networks

configuration

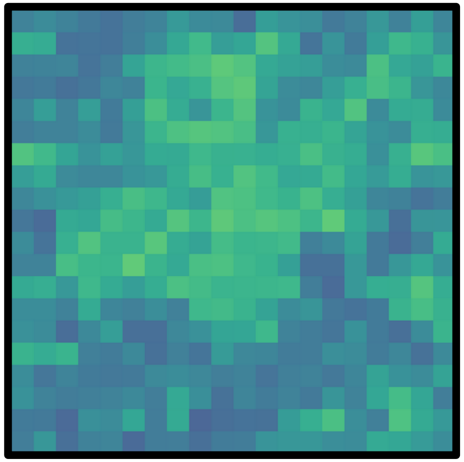


Keeping training costs low

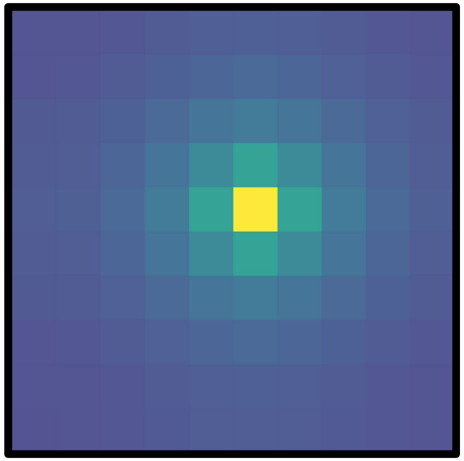
Total cost ≈ configuration production cost

- ☆ Translational symmetry ⇨ use convolutional networks
- ☆ Information within correlation length ⇨ control network footprint

configuration



2-point correlation



Keeping training costs low

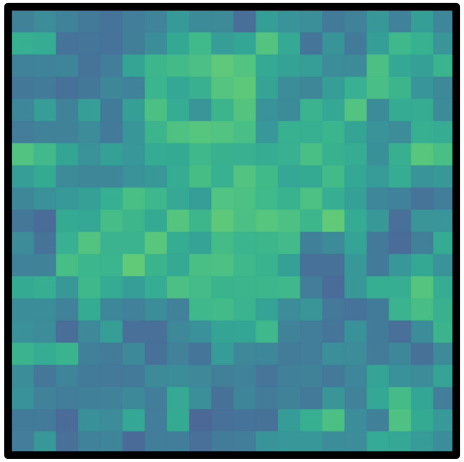
Total cost ≈ configuration production cost

- ☆ Translational symmetry ⇨ use convolutional networks
- ☆ Information within correlation length ⇨ control network footprint
  - ↳ simple affine coupling layer with no hidden layers

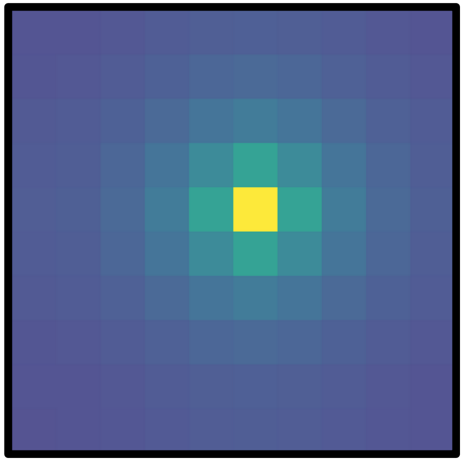
$$\phi_x \to e^{s(\phi)} \phi_x + t(\phi)$$

  - ↳ footprint can be controlled with the kernel size k of the CNNs s and t

configuration



2-point correlation



Keeping training costs low

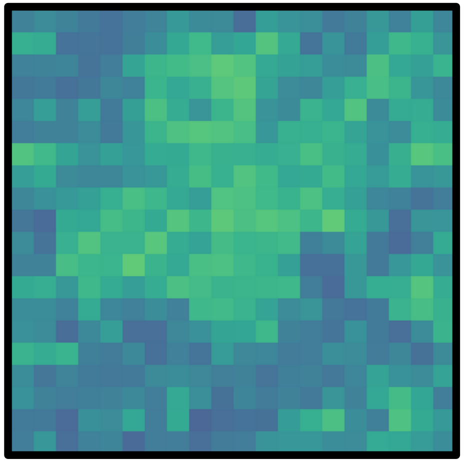
Total cost ≈ configuration production cost

- Translational symmetry ⇨ use convolutional networks
- Information within correlation length ⇨ control network footprint
  - simple affine coupling layer with no hidden layers

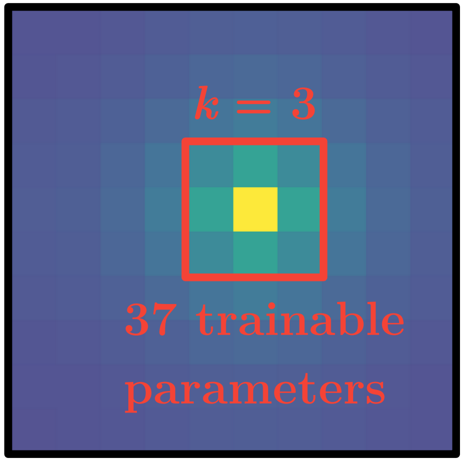
$$\phi_x \to e^{s(\phi)} \phi_x + t(\phi)$$

  - footprint can be controlled with the kernel size k of the CNNs s and t

configuration



2-point correlation

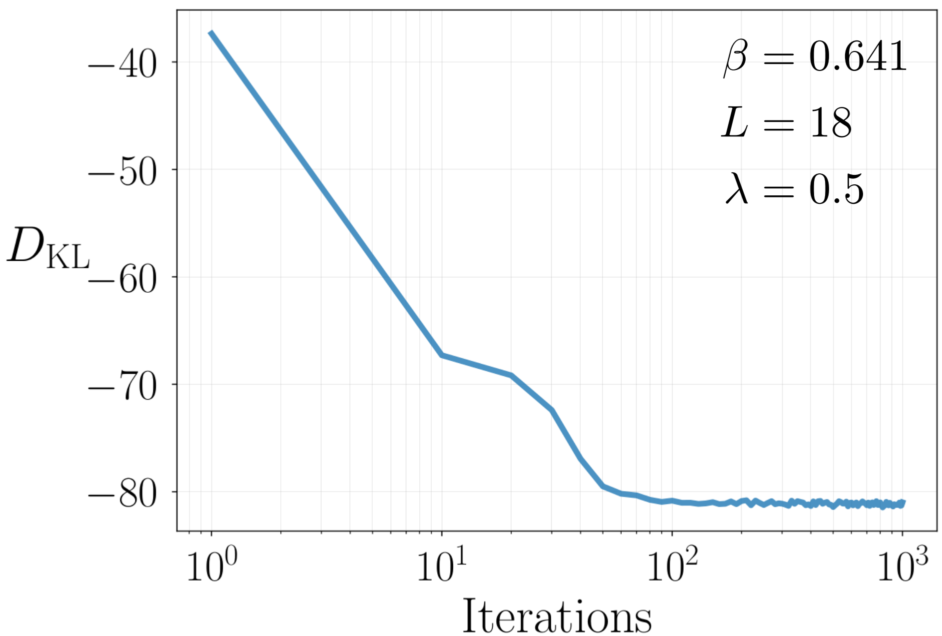


Can this simple network learn something?

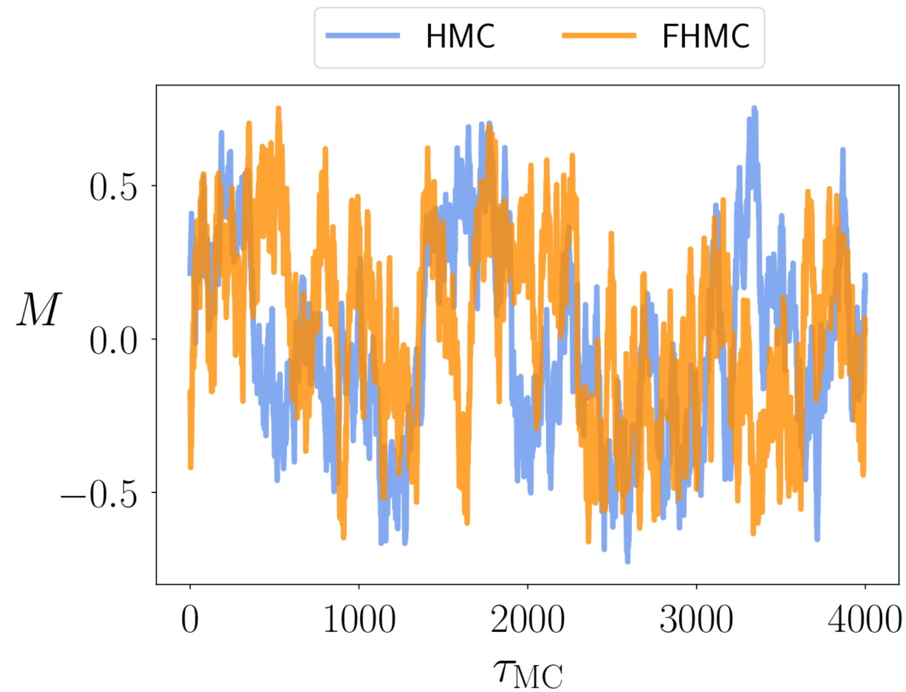
Check 1: minimal network

Minimal architecture
- 1 affine coupling layer
- $k = 3$

1. Train network minimizing KL



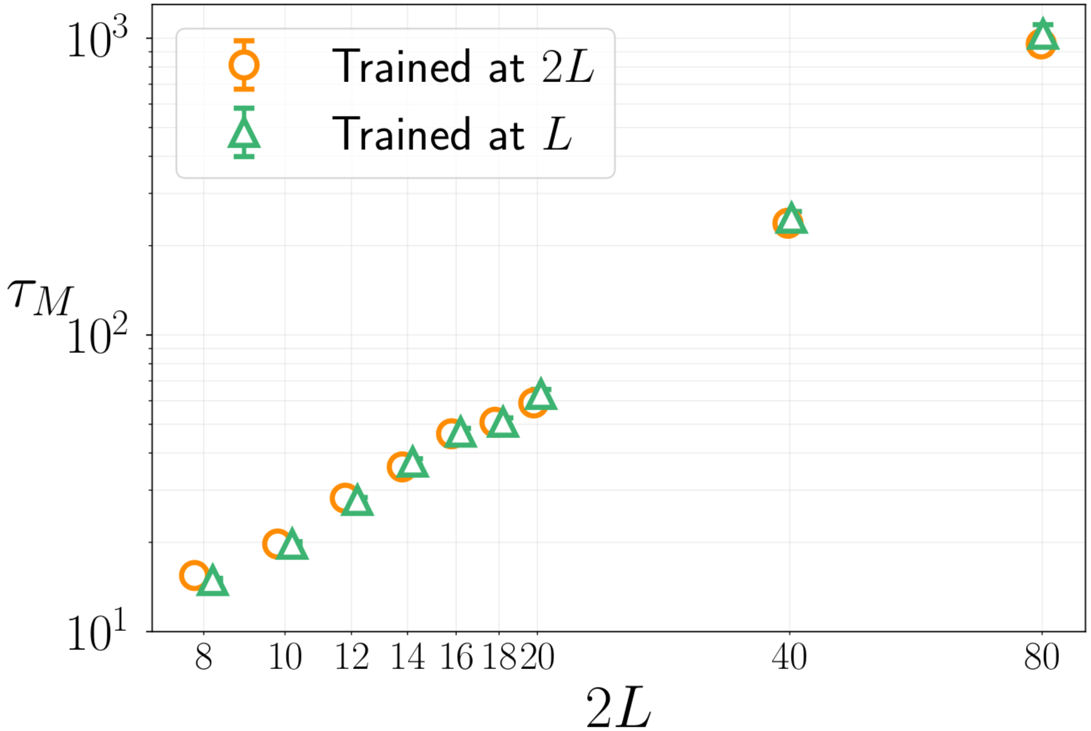
2. Compare magnetization history with HMC



Algorithm	τ_M
HMC	100.4(2)
FHMC	74.4(3)

- KL divergence saturates fast
- Results from both algorithms are consistent with each other
- Learned trivializing flow reduces autocorrelations even with simple architectures

Convolutional networks can be reused for bigger volumes



Autocorrelation times remain the same on bigger volumes

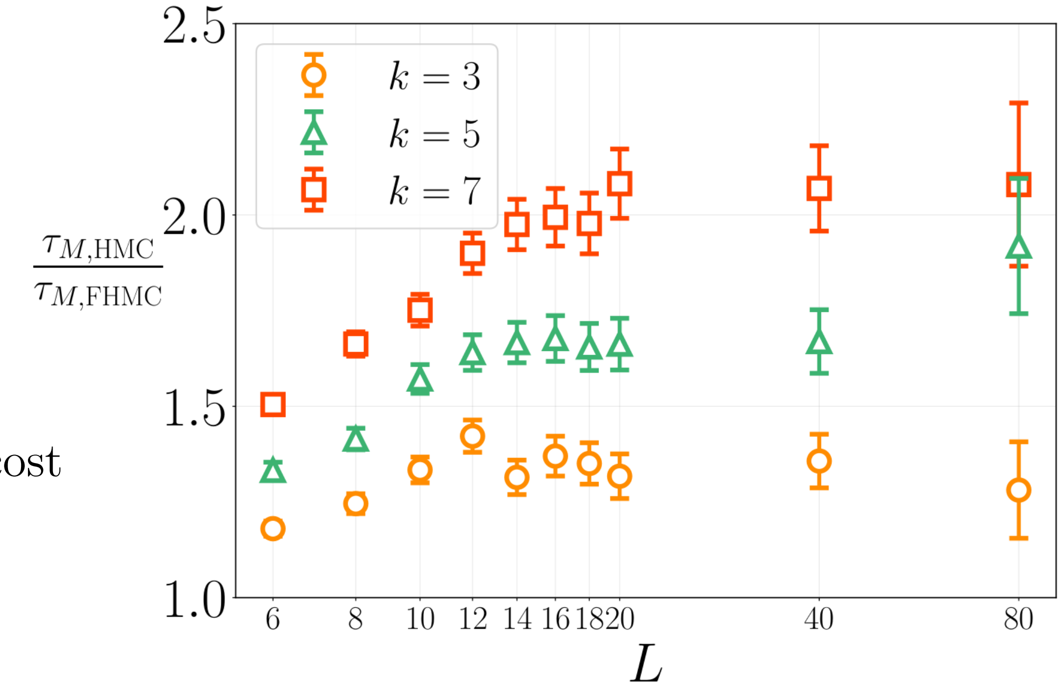
Training should be done at the correlation length

Scaling of the computational cost

Magnetization: $M = \frac{1}{V}\sum_x \phi_x$

- Lattice with fixed physical size
- Simple network architectures: 1 affine layer
- Networks trained until saturation
- Training cost negligible w.r.t. production cost

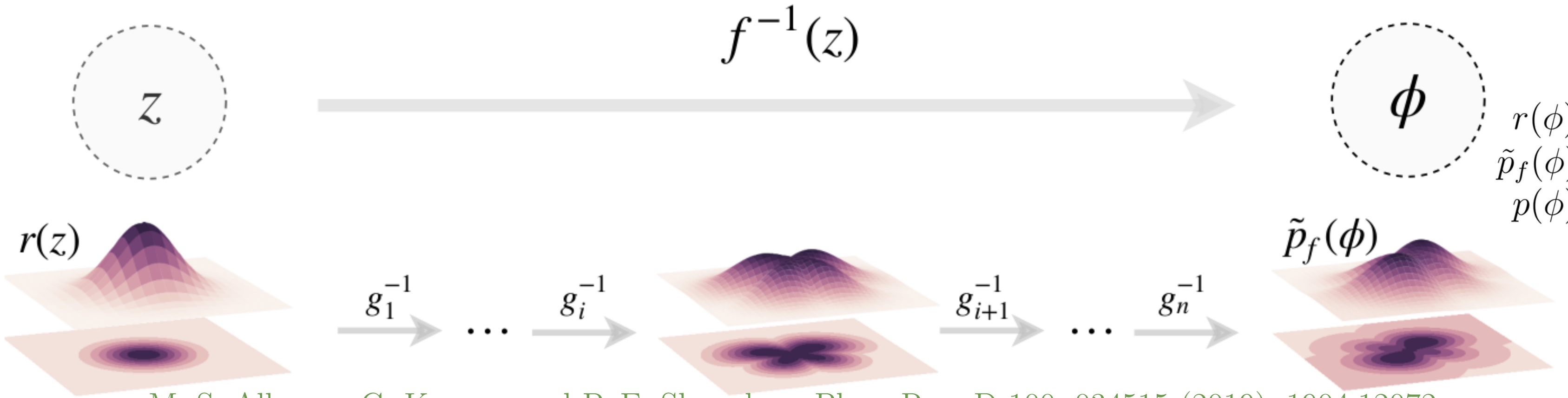
Total cost ≈ configuration production cost



- Autocorrelation times are decreased compared to HMC
- For a fixed architecture the scaling does not improve

⇨ Can this change with a different input theory?

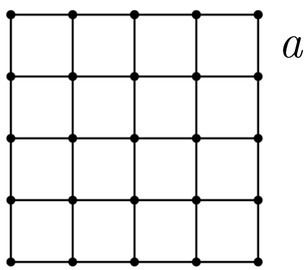
Training from a coarser theory

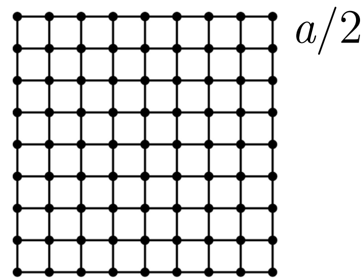


$r(\phi)$: input distribution
$\tilde{p}_f(\phi)$: output distribution
$p(\phi)$: target distribution

M. S. Albergo, G. Kanwar and P. E. Shanahan, Phys. Rev. D 100, 034515 (2019), 1904.12072

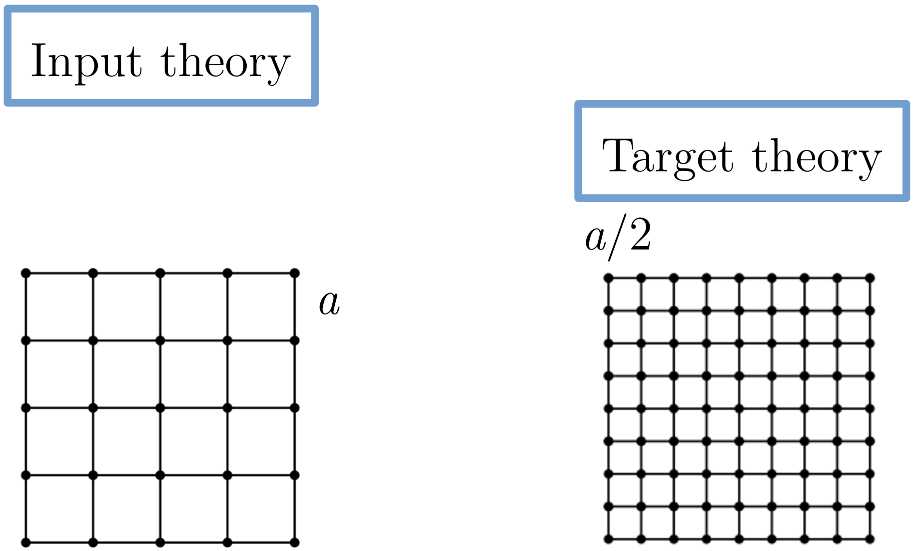
Input theory	Target theory
$r(\phi) \equiv p_{\beta'}(\phi) = \frac{1}{\mathcal{Z}_{\beta'}} e^{-S_{\beta'}[\phi]}$	$p_\beta(\phi) = \frac{1}{\mathcal{Z}_\beta} e^{-S_\beta[\phi]}$

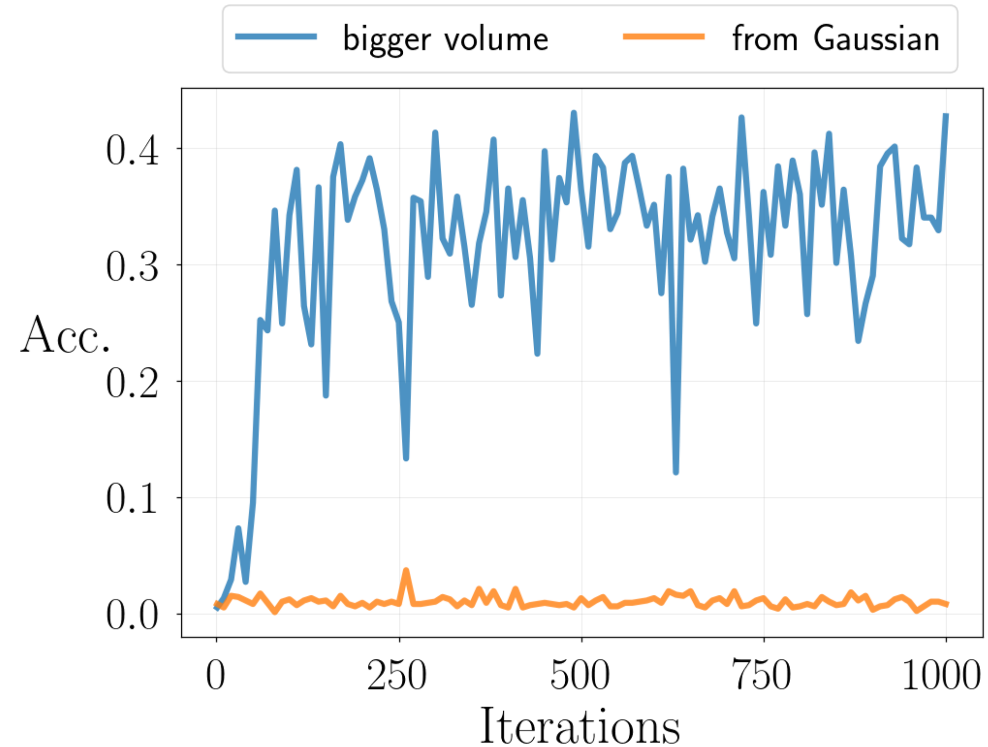




- Longest correlation length is already captured in the coarsest lattice
- Problem: the number of degrees of freedom is not the same at same physical volume
  - Studied two possible workarounds

Training from a coarser theory: bigger volume



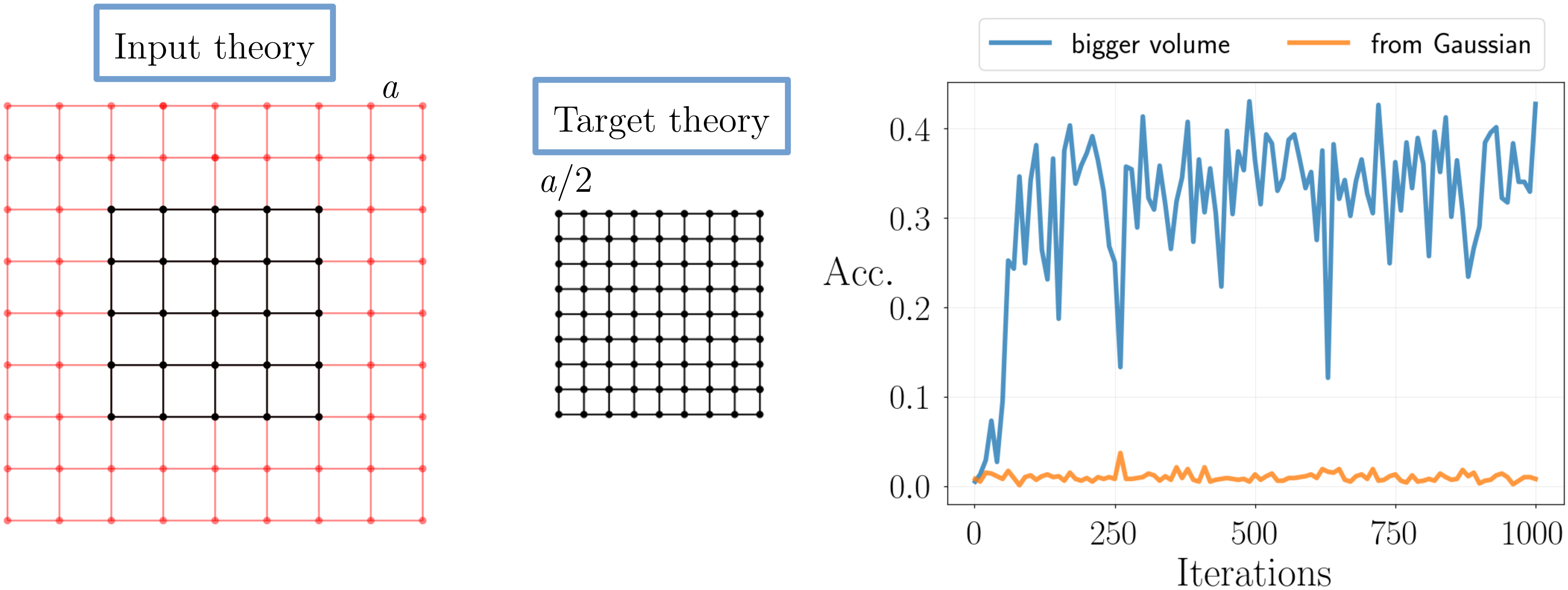


⇨ Training from larger to smaller lattice spacing keeping same degrees of freedom

Algorithm	τ_M
HMC	77.9(1.5)
FHMC (coarse theory, bigger volume)	63.6(2.2)
FHMC (from Gaussian)	56.9(1.8)

⇨ Higher Metropolis-Hastings acceptance does not imply lower autocorrelation times

Training from a coarser theory: bigger volume



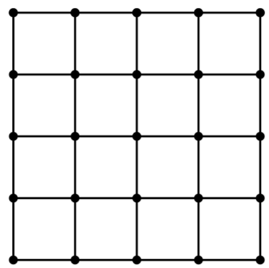
⇨ Training from larger to smaller lattice spacing keeping same degrees of freedom

Algorithm	τ_M
HMC	77.9(1.5)
FHMC (coarse theory, bigger volume)	63.6(2.2)
FHMC (from Gaussian)	56.9(1.8)

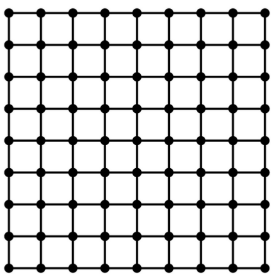
⇨ Higher Metropolis-Hastings acceptance does not imply lower autocorrelation times

Training from a coarser theory: interpolation

Input theory



Target theory



[N. Matsumoto, Mon 4:00PM]

[R. Abbott, Mon 4:20PM]

☆ Some information of correlation length already there

	Algorithm	τ_M
	HMC	77.9(1.5)
FHMC	● ● $\sim \mathcal{N}(0,1)$	56.9(1.8)
FHMC	● $\sim p_{\beta'}(\phi)$ ● $\sim \mathcal{N}(0,1)$	55.2(1.8)

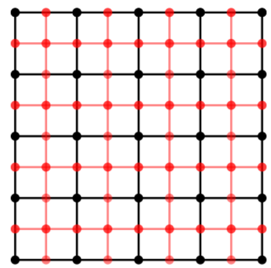
⇨ Not better than training directly from normal numbers

Training from a coarser theory: interpolation

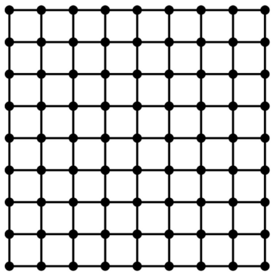
interpolated **Input theory** **Target theory**

[N. Matsumoto, Mon 4:00PM]

[R. Abbott, Mon 4:20PM]



a



$a/2$

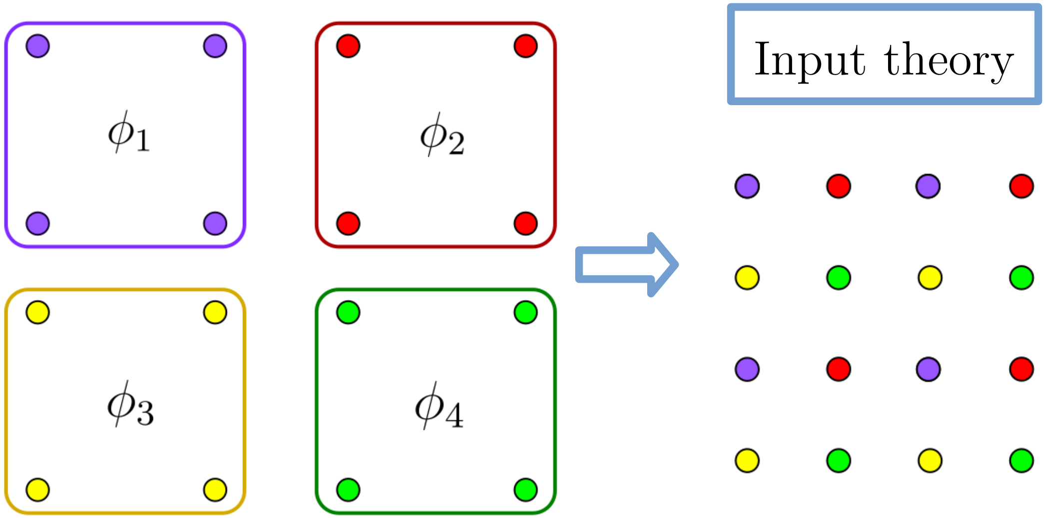
Some information of correlation length already there

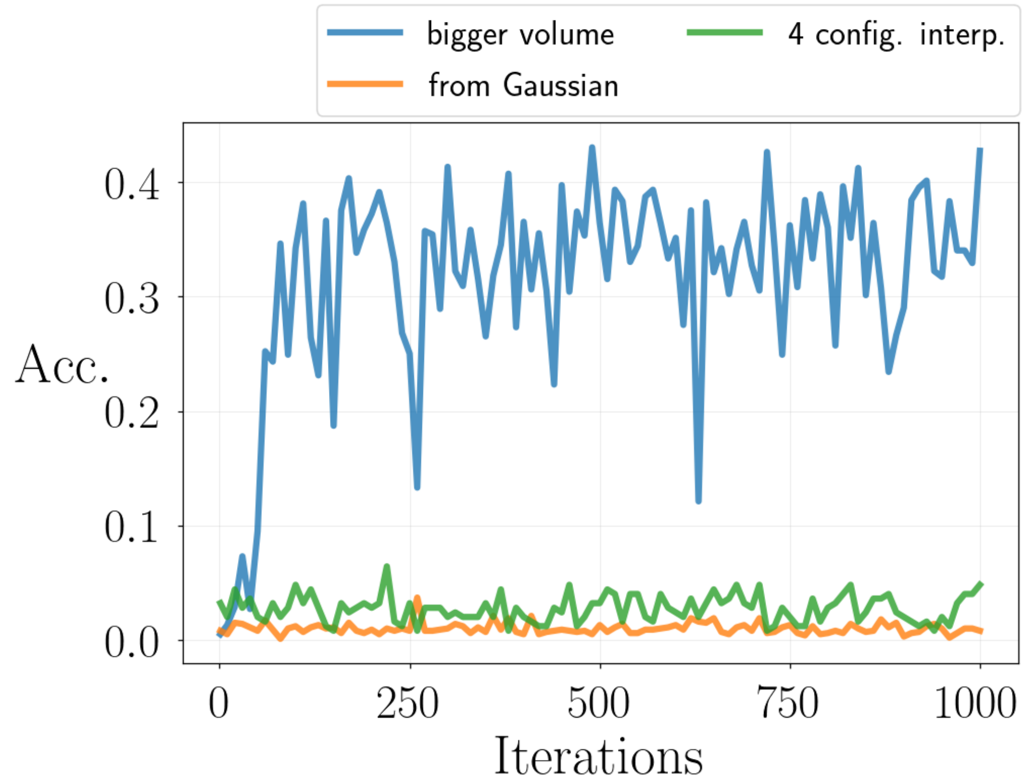
	Algorithm	τ_M
	HMC	77.9(1.5)
FHMC	● ● $\sim \mathcal{N}(0,1)$	56.9(1.8)
	● $\sim p_{\beta'}(\phi)$ ● $\sim \mathcal{N}(0,1)$	55.2(1.8)

Not better than training directly from normal numbers

Training from a coarser theory: interpolation

☆ Combine 4 coarse configurations to reinforce information of correlation length

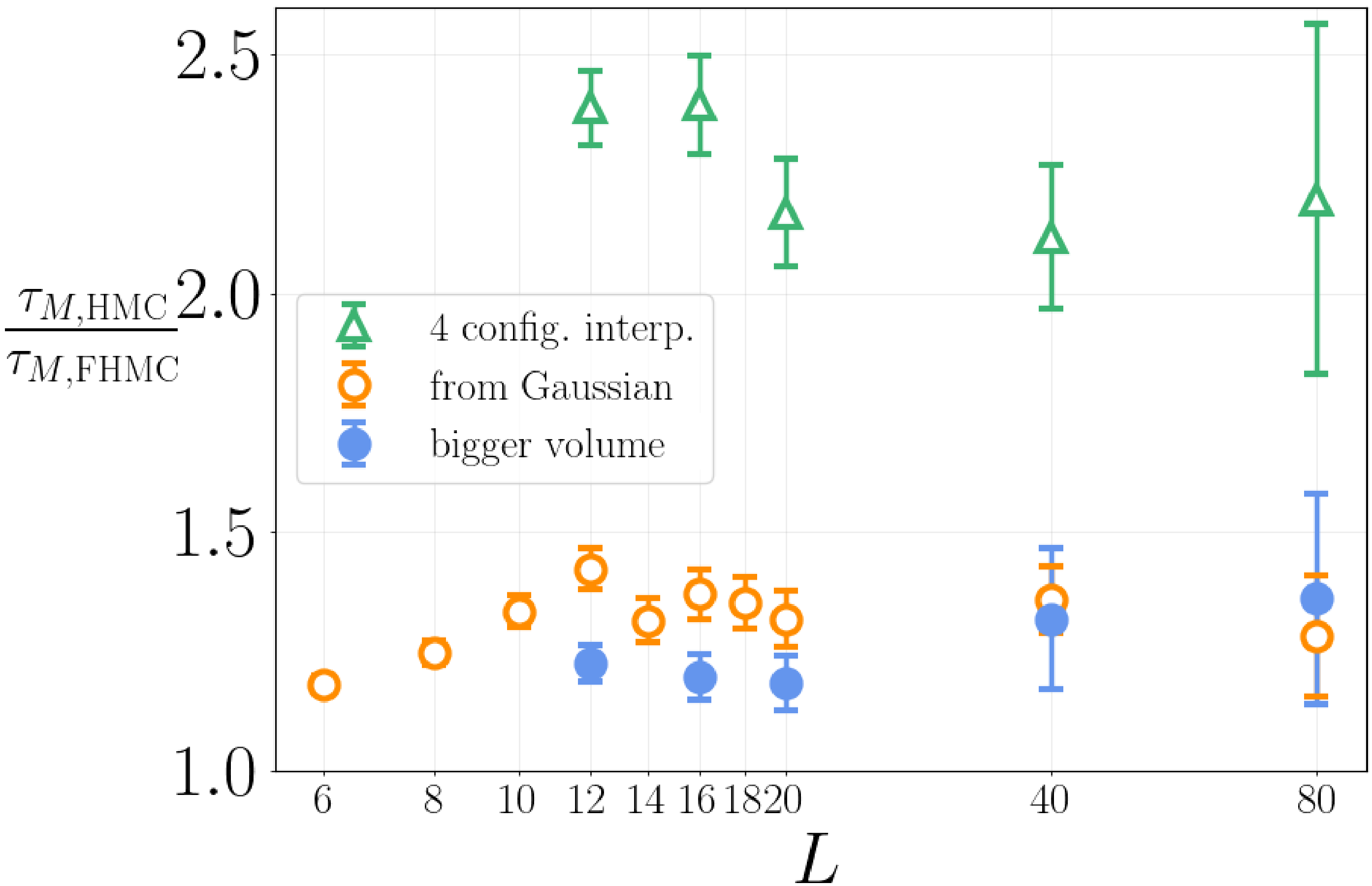




Algorithm	τ_M
HMC	77.9(1.5)
FHMC (bigger volume)	63.6(2.2)
FHMC (from Gaussian)	56.9(1.8)
FHMC (4-config. interpolation)	32.5(1.2)

⇨ Autocorrelations improved with respect to Gaussian

Training from a coarser theory: scaling study



- Autocorrelation times are decreased compared to HMC
- For a fixed architecture the scaling does not improve

⇨ Maybe iterative training to coarser theories can help

Summary & Outlook

- This works with simple network architectures
- The algorithm improves the autocorrelation times of HMC, but the scaling is the same with fixed architecture
- The networks can be trained at a small lattice size and reused at a larger volume (with no further training)
- Training from coarser lattices at bigger physical volume has better MH acceptance, but interpolation leads to better autocorrelation times at fixed architecture
  - ↳ Iterative application of training from coarsest lattice
- Although autocorrelation times are further improved, scaling towards the continuum is not
  - ↳ Can this algorithm help with topology freezing?

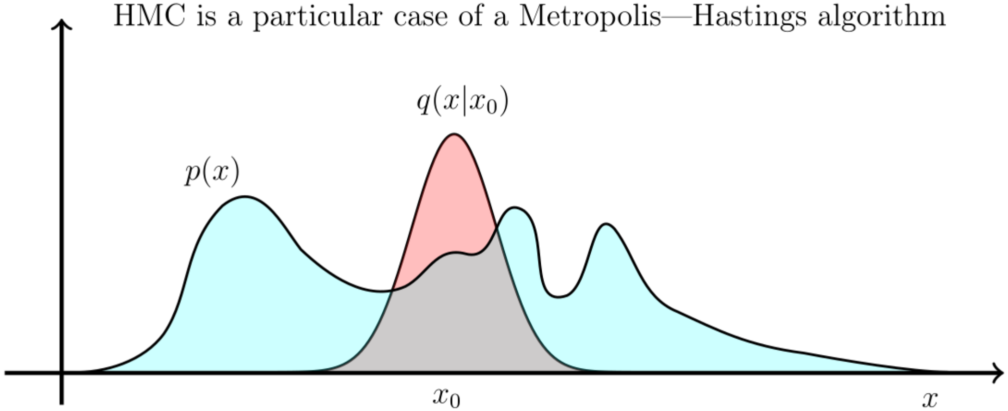
Training from a coarser theory: iterated interpolation

$L = 40$

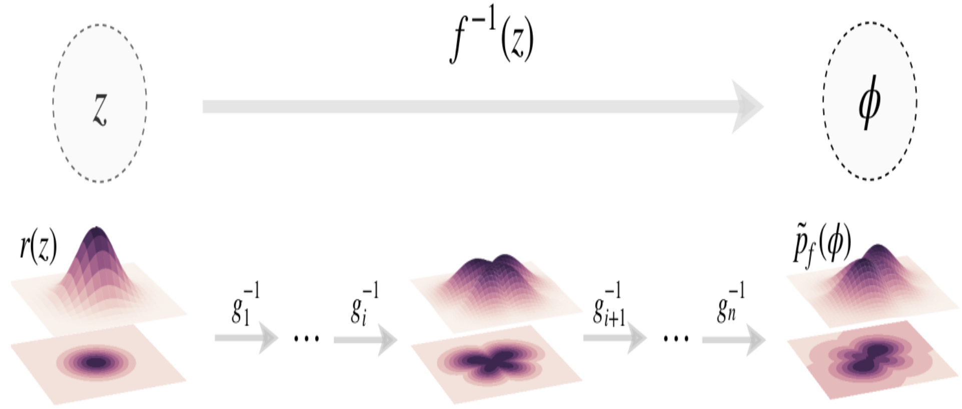
Algorithm	τ_M
HMC	570(21)
FHMC (from Gaussian)	420(16)
FHMC (4-config. interpolation)	269(17)
FHMC (4-config. interpolation) x2	164.5(5.0)

$L = 80$

Algorithm	τ_M
HMC	2518(130)
FHMC (from Gaussian)	1965(165)
FHMC (4-config. interpolation)	1146(182)
FHMC (4-config. interpolation) x2	788(128)



HMC



Normalizing flows

HMC	Normalizing flows
Target distribution	**Target distribution**
$p(\phi) = e^{-S[\phi]}$	$p(\phi) = e^{-S[\phi]}$
Proposal distribution	**Proposal distribution**
$q(\phi' \mid \phi) \to \text{Hamilton eqs.}$	$q(\phi' \mid \phi) \to \tilde{p}_f(\phi') = r(f(\phi')) \left\lvert \det \frac{\partial f(\phi')}{\partial \phi'} \right\rvert$
Accept-reject step	**Accept-reject step**
$p_{\text{acc}}(\phi' \mid \phi) = \min\left\{1, \frac{p(\phi')}{p(\phi)} \frac{q(\phi \mid \phi')}{q(\phi' \mid \phi)}\right\}$	$p_{\text{acc}}(\phi' \mid \phi) = \min\left\{1, \frac{p(\phi')}{p(\phi)} \frac{\tilde{p}_f(\phi)}{\tilde{p}_f(\phi')}\right\}$

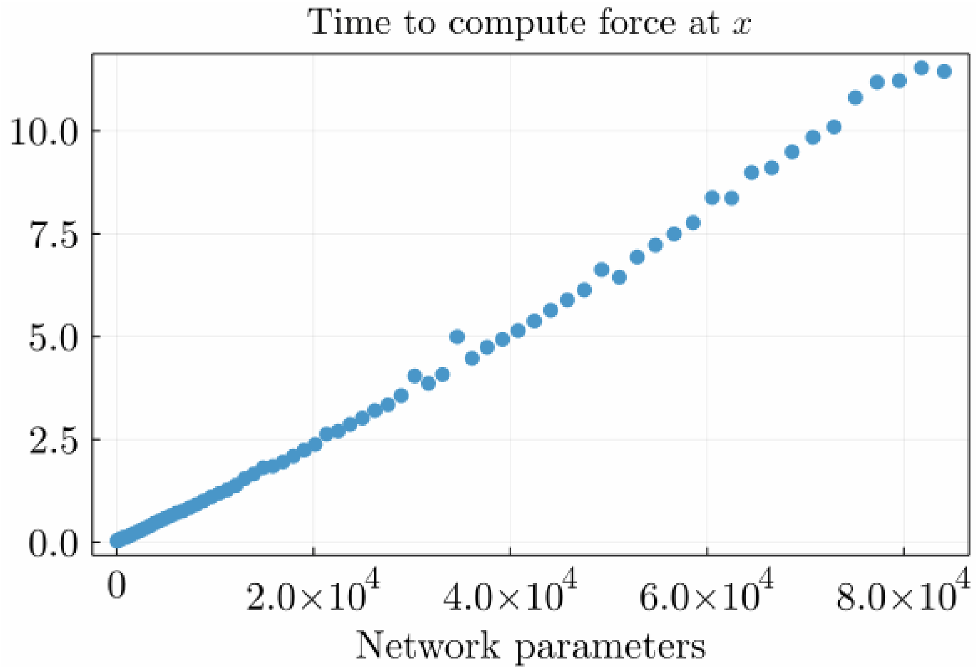
⇨ Independent proposals

Automatic differentiation

$$Z = \int D\phi \, e^{-S(\phi)} \xrightarrow{\tilde{\phi} = f(\phi)} \int D\tilde{\phi} \, e^{-S(f^{-1}(\tilde{\phi})) + \log\det J[f]} \equiv \int D\tilde{\phi} \, e^{-\tilde{S}(\tilde{\phi})}$$

- We need to compute the force of the new variables: $\tilde{F}_x = \frac{\partial \tilde{S}[\tilde{\phi}]}{\partial \tilde{\phi}_x}$

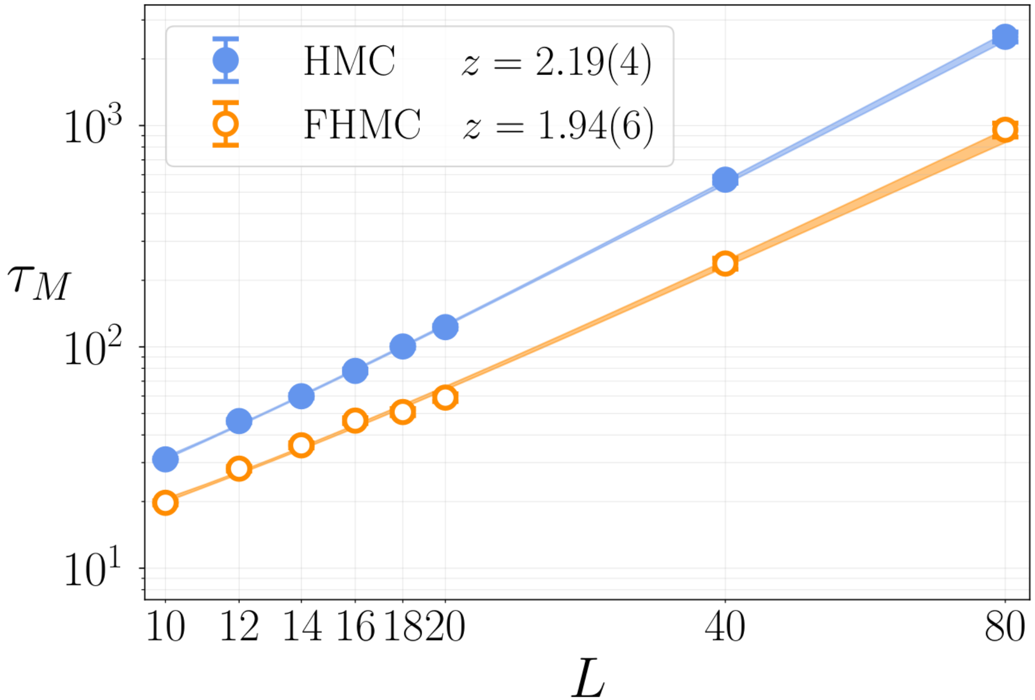
  ↳ automatic differentiation



$N_{\text{params.}} \propto k^2$

- Scaling the kernel size also increases the number of operations to compute the HMC force

Magnetization: $M = \frac{1}{V}\sum_x \phi_x$



- Fit autocorrelation to $\tau \propto \xi^z$

$$z_{\text{HMC}} = 2.19(4)$$

$$z_{\text{FHMC}} = 1.94(6)$$

- Scaling the kernel size leads to slight improvement in the autocorrelation scaling