Generative models for scalar field theories: How to deal with poor scaling?

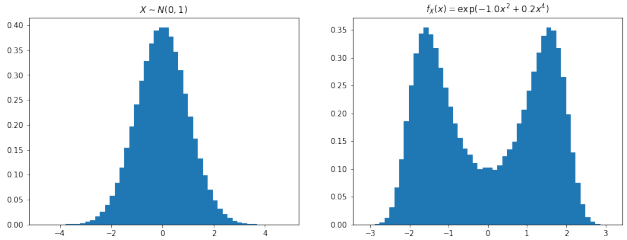
Javad Komijani
(with Marina K. Marinkovic)
ETH zürich

The 39th International Symposium on Lattice Field Theory
August, 2022

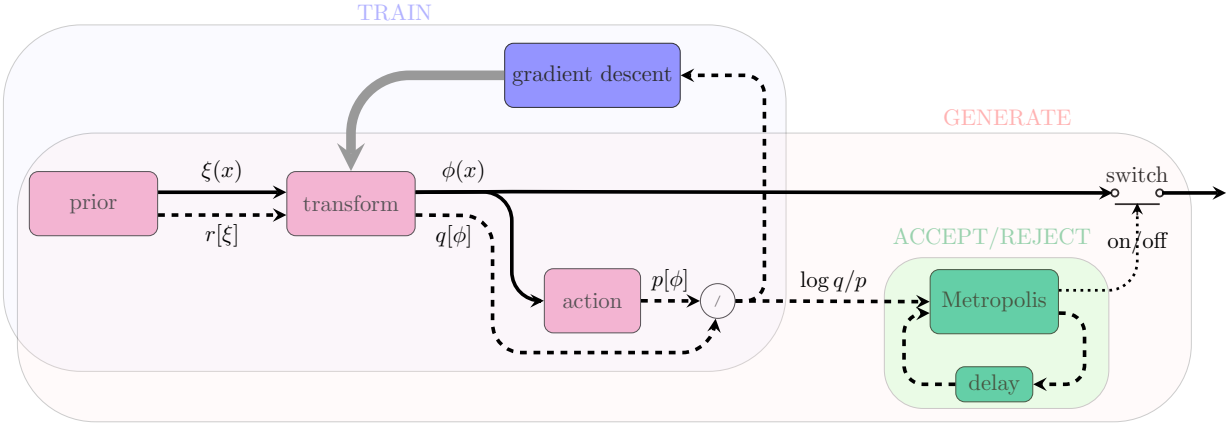
Inverse Transform Sampling & Normalizing Flows

- Inverse transform sampling (ITS) as a method to generate a random variable with a flexible distribution:

$$y \triangleq F_Y^{-1} \circ F_X(x)$$



- Normalizing flows (NFs) as a generalization of ITS to higher dimensions with a series of *learnable*, invertible transformations

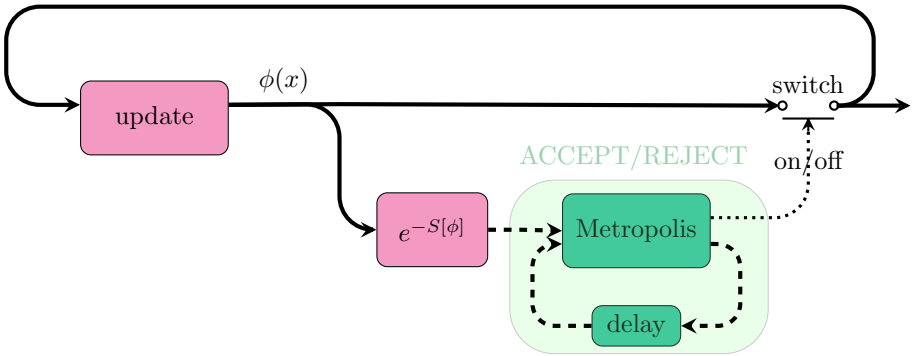


- (reverse) Kullback-Leibler divergence measures how similar two distributions are:

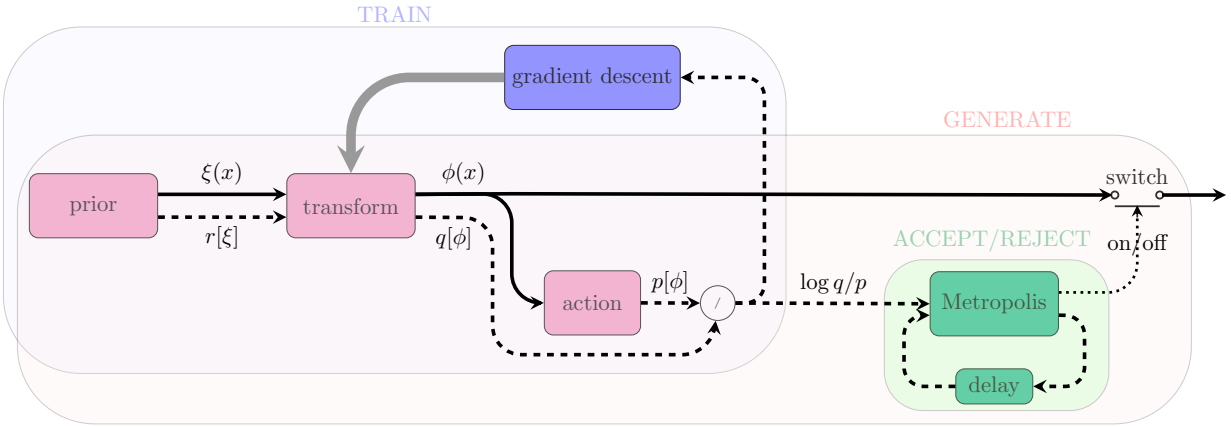
$$D_{\mathsf{KL}}(q||p) \equiv \int d\phi \; q[\phi]\Big(\log q[\phi] - \log p[\phi]\Big) \;\geq\; 0$$

Lattice Field Theory & MC Simulations & NFs

- Monte Carlo simulations:
  - Traditional methods typically suffer from critical slowing down, topological freezing, $\cdots$



- Normalizing flows for (2-dim) scalar theories:
  explored first @ MIT & collaborators: [arXiv:1904.12072, 2002.02428, 2003.06413]



1. *Networks for Normalizing Flows*
2. *Poor Scaling at Large Volumes*

Designing Networks for Normalizing Flows

- Checkerboard strategy for normalizing flow is widely used: divide the data to active & passive, update the active, and....
- Widely used layers of neural networks

Linear (Dense) Net

- Great for small-size lattice
- Number of parameter $\propto N^2$

Conv Net

- Respect translational symmetry & fewer parameters
- Many layers needed to correlate a big lattice

Designing Networks for Normalizing Flows

- Checkerboard strategy for normalizing flow is widely used: divide the data to active & passive, update the active, and....
- Widely used layers of neural networks

Linear (Dense) Net

- Great for small-size lattice
- Number of parameter $\propto N^2$

Conv Net

- Respect translational symmetry & fewer parameters
- Many layers needed to correlate a big lattice

- Other possibilities?

What about constructing layers inspired by symmetries of the action & effective theories to propagate correlation in more efficient ways?

Effective Action & Power Spectral Density

- A scalar field theory in n spacetime dimensions:

$$S[\phi] = \int d^n x \left(\frac{1}{2} \partial_\mu \phi \partial_\mu \phi + \frac{1}{2} m^2 \phi^2 + \sum_{j=3}^{J} g_j \phi^j \right)$$

- The quantum effective action:

$$\Gamma[\phi] = \frac{1}{2} \int d^n k \, \tilde{\phi}(-k) \Big(k^2 + m^2 - \Pi(k^2) \Big) \tilde{\phi}(k) + \cdots$$

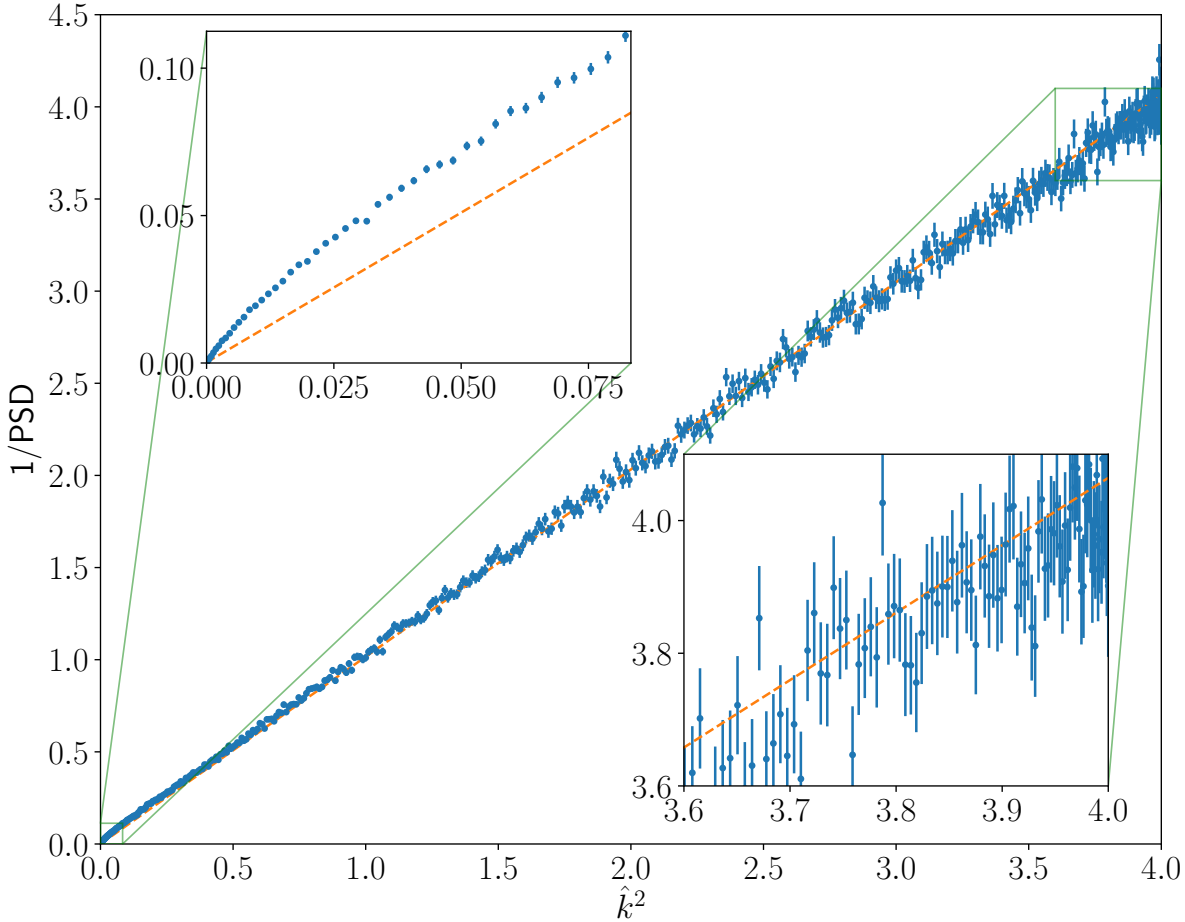
(Tree-level Feynman diagrams give the complete scattering amplitude)

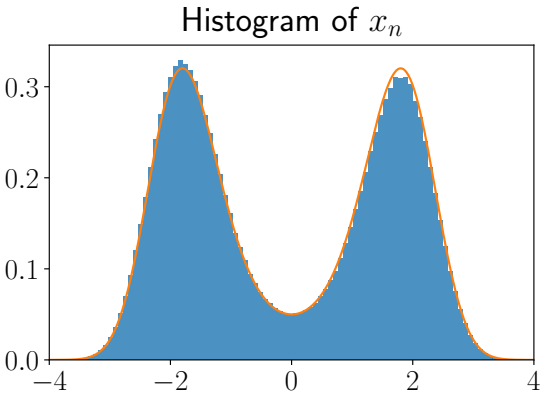
$\Big(k^2 + m^2 - \Pi(k^2) \Big)$ is the inverse of two-point correlator/Green's function

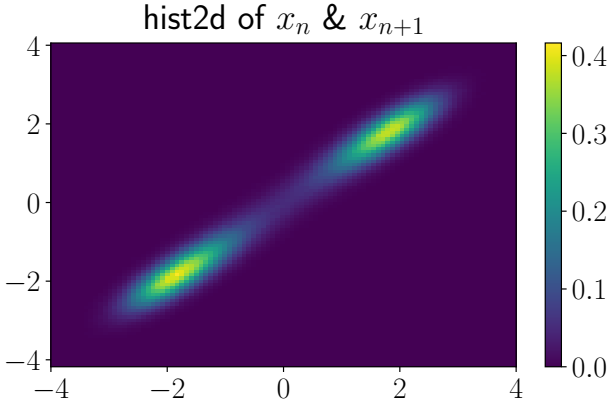
$\Big(k^2 + m^2 - \Pi(k^2) \Big)$ is the inverse of power spectral density

A close look to PSD:

- The inverse of PSD of a 1-dim double-well potential (from MC simulation)

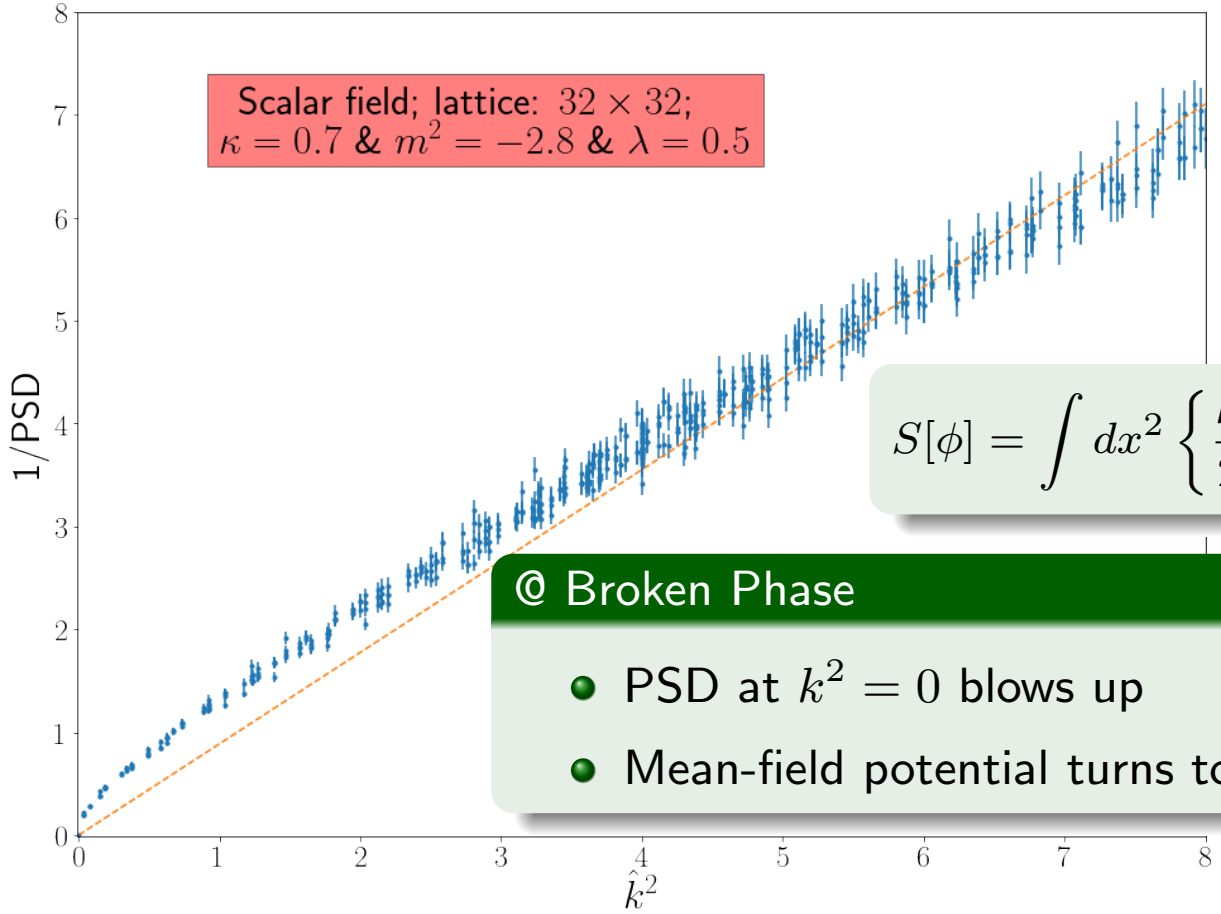






- 1/PSD can be manipulated using a positive, monotonically increasing function of $\hat{k}^2$; ML techniques can be employed to construct such a function
- Manipulating PSD is NOT a local operation; it affects the correlation in data at largest & shortest scales

The inverse of PSD for a 2-dim double-well potential (from MC simulation)



$$S[\phi] = \int dx^2 \left\{ \frac{\kappa}{2} (\partial_\mu \phi(x))^2 + \frac{m^2}{2} \phi(x)^2 + \lambda \phi(x)^4 \right\}$$

@ Broken Phase

- PSD at $k^2 = 0$ blows up
- Mean-field potential turns to a double-well potential

Inspired by mean-field theory

One can build a general function (a neural network) to map the mean field to a mean field of interest

For the sake of comparison with [arXiv:2105.12481, Debbio et.al.] we consider this 2-dim action

$$S[\phi] = \int dx^2 \left\{ \frac{\kappa}{2}(\partial_\mu \phi(x))^2 + \frac{m^2}{2}\phi(x)^2 + \lambda\phi(x)^4 \right\}$$

where $\kappa = \beta$, $m^2 = -4\beta$, and $\lambda = 0.5$, with $\beta \in [0.5, 0.8]$ in our simulations.

Goal:

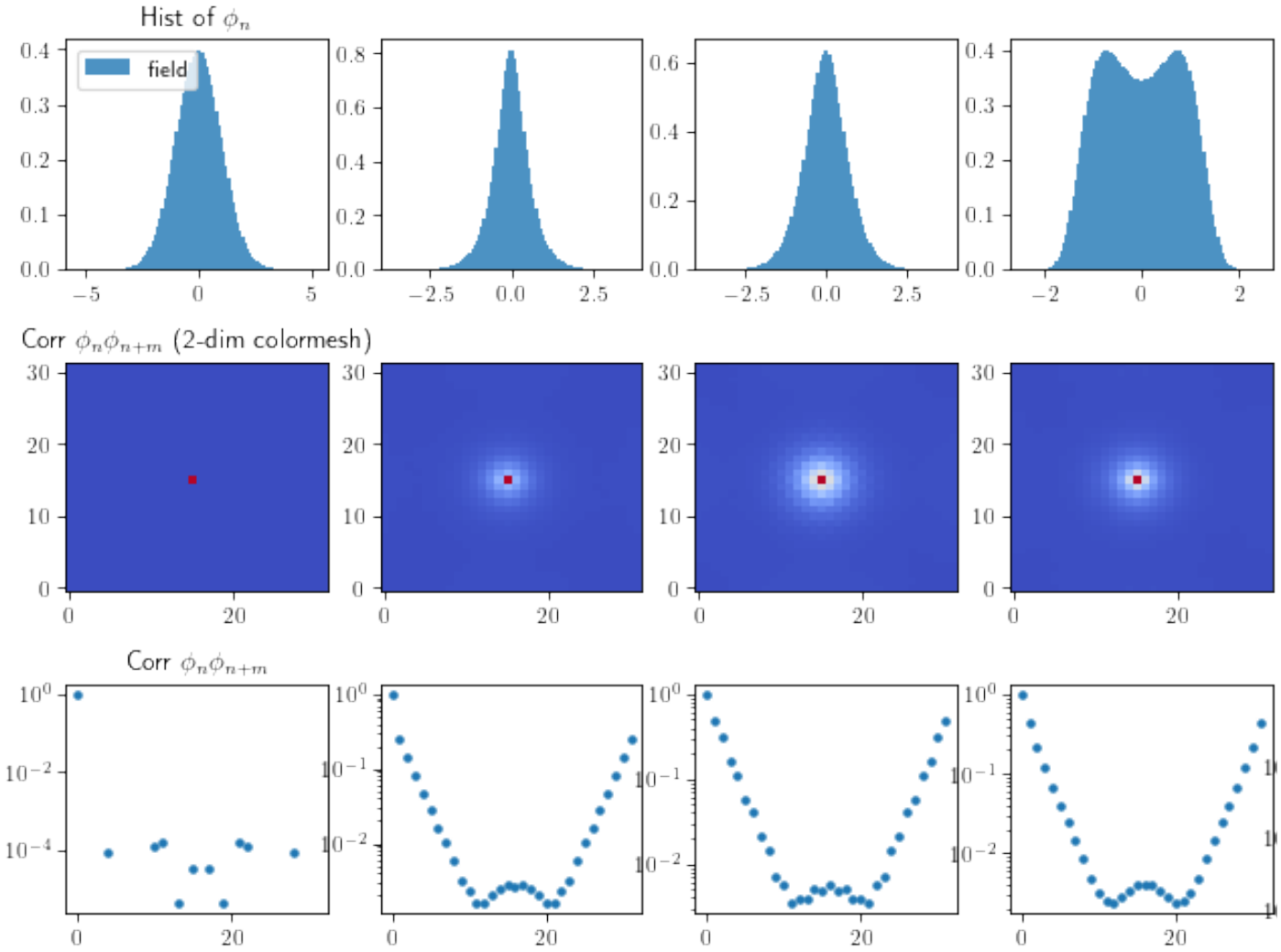
Following suggestions inspired by effective theories, we aim to construct neural networks that are

- economic w.r.t. parameters
- do not require many layers of ConvNet to propagate correlations

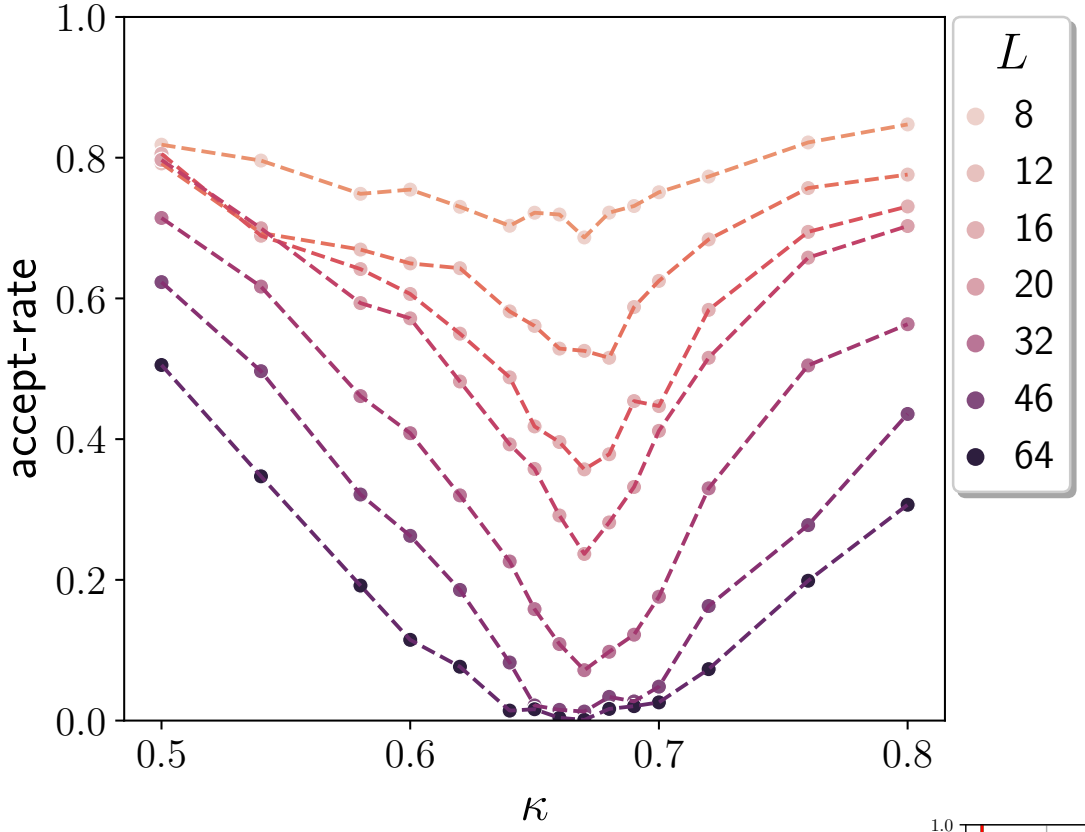
An architecture for 2-dim scalar theories

1. an initial layer to manipulate PSD of white normal noise & general activation
2. followed by two layers of affine coupling implemented with ConvNet & general activation

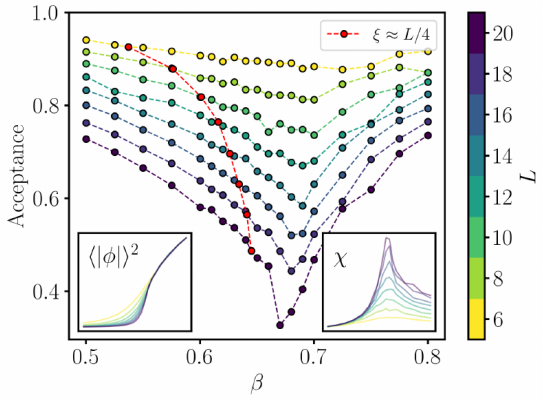
Simulation parameters: $\kappa = 0.6$ & $L = 32$



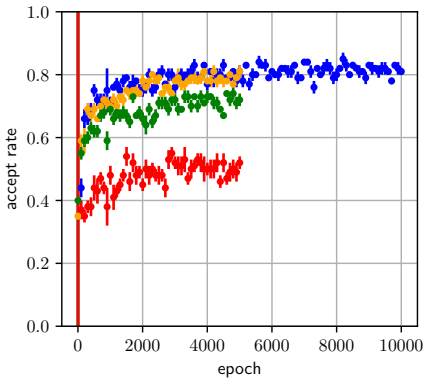
Acceptance rate & critical point & large volume

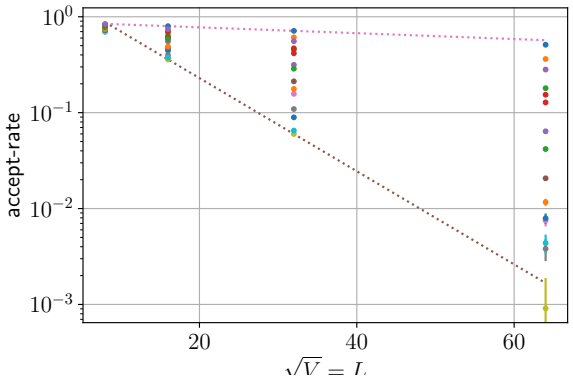


Compare with
[arXiv:2105.12481, Debbio et.al.]



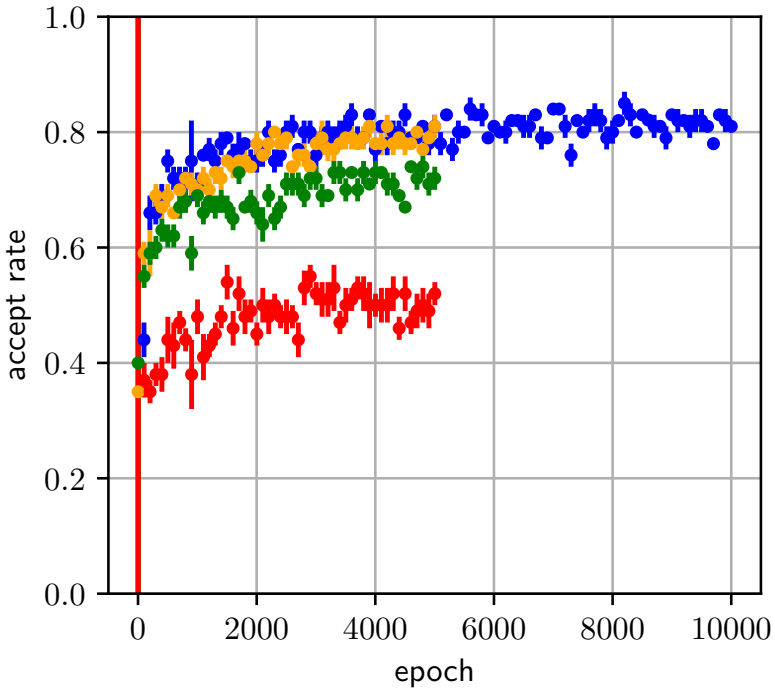
- $L \in [8, 64]$
- # parameters $\approx$ 3.4K for all cases
- trained with transfer learning
- poor scaling @ critical point

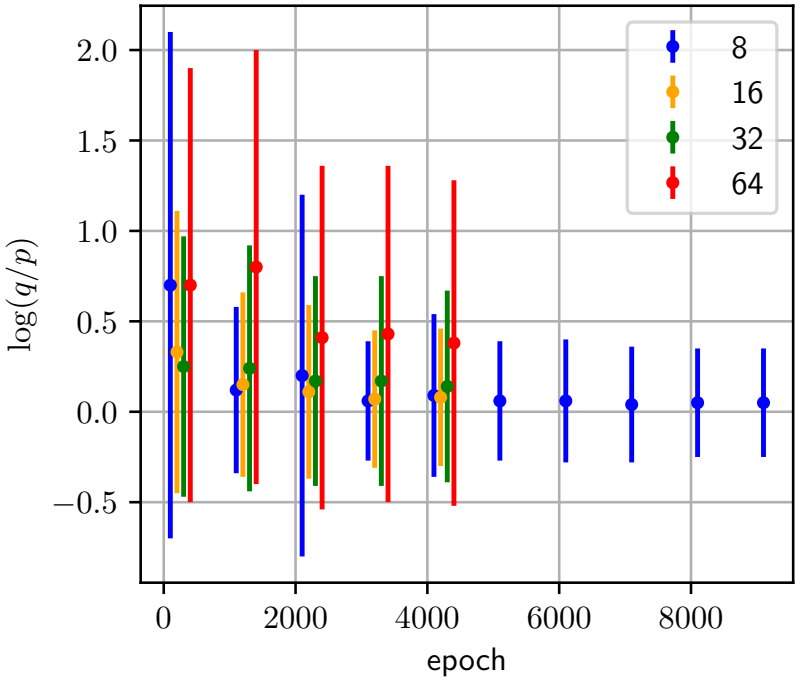




Uncertainty in $\log(q/p)$ & acceptance rate

- Optimization for $\kappa = 0.5$ for $L \in \{8, 16, 32, 64\}$:

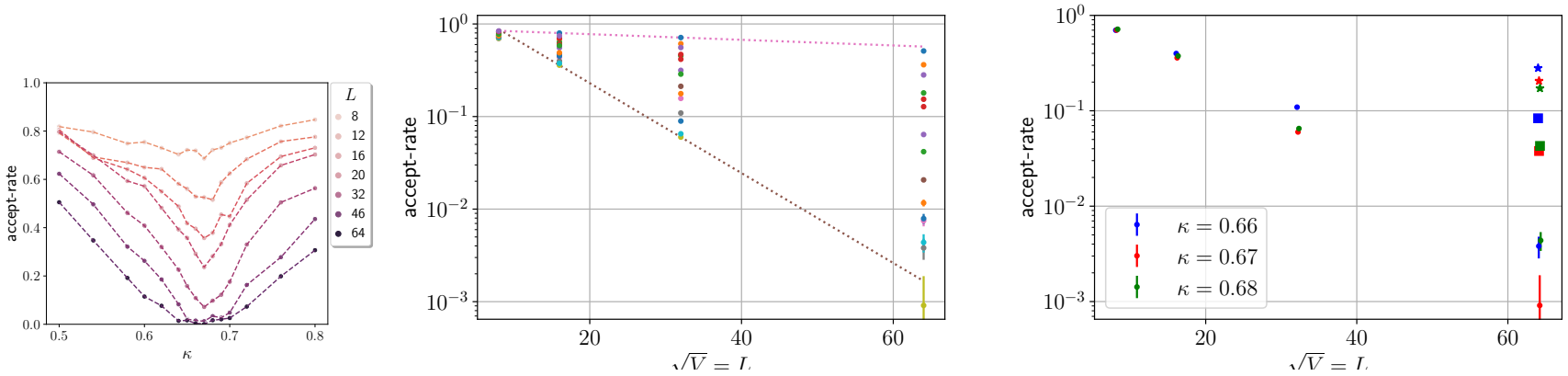




- The uncertainty in $\log(q/p)$ determines acceptance rate
- It looks like uncertainty in $\log(q/p)$ scales with $\sqrt{\text{volume}}$ at large volumes
- Justification: divide the lattice into n blocks with almost independent fluctuations

NFs with block updating

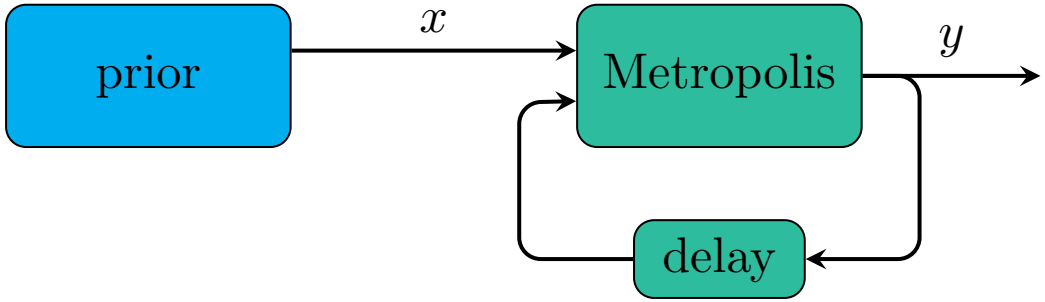
- The uncertainty in $\log(q/p)$ determines acceptance rate
- It looks like uncertainty in $\log(q/p)$ scales with $\sqrt{\text{volume}}$ at large volumes
- Justification: divide the lattice into n blocks with almost independent fluctuations

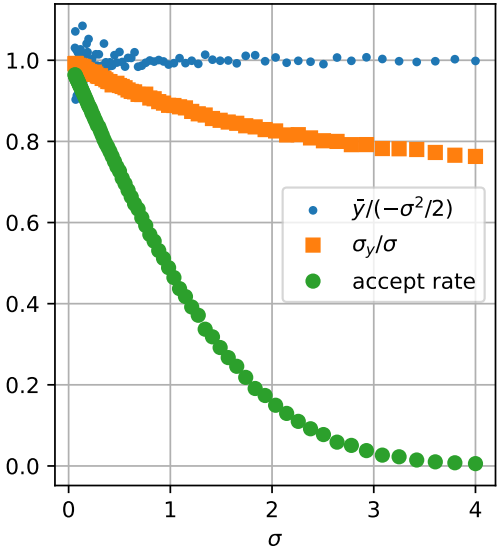


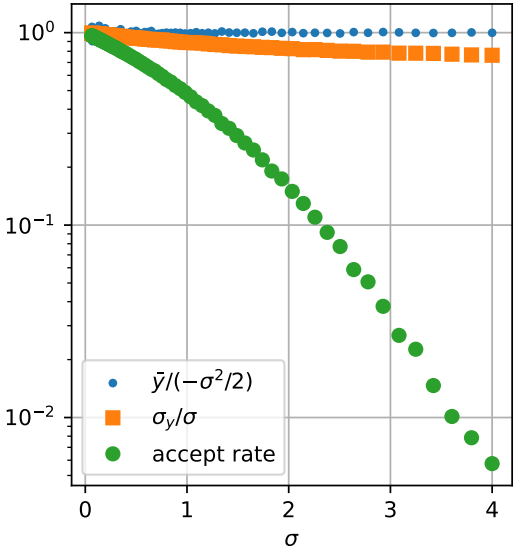
- $L = 64$ model is sampled block by block
  - $n_{\text{blocks}} = 2^2$ (square), acceptance rate $\sim L = 32$
  - $n_{\text{blocks}} = 4^2$ (star), acceptance rate $\sim L = 16$
- Asymptotic scaling & saturated training

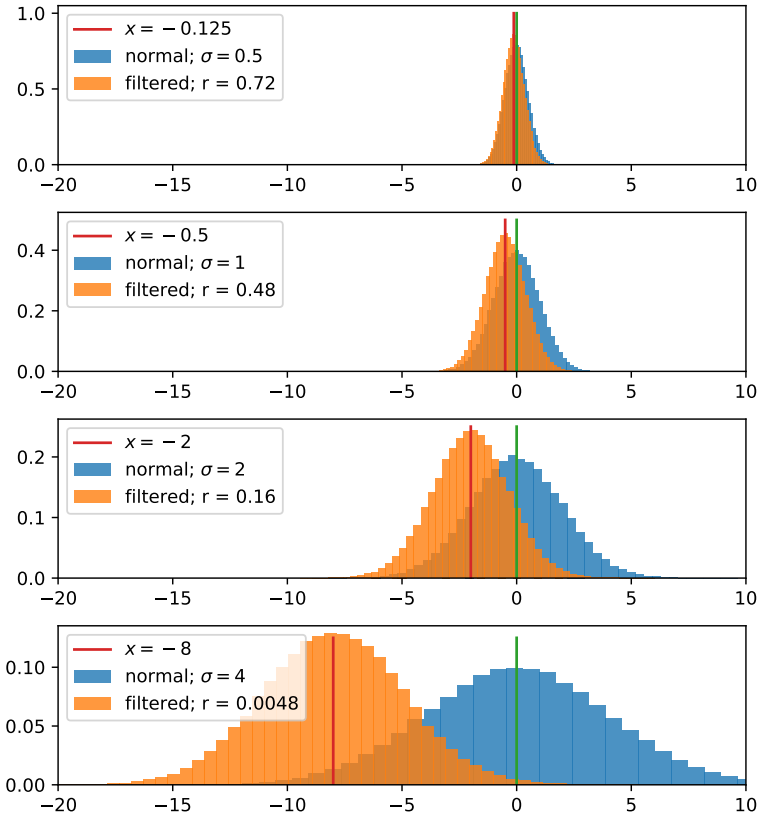
Uncertainty in $\log(q/p)$ & block size

- Toy model: $x \sim N(0, \sigma^2)$ and y is the output of the "Metropolis Filter"



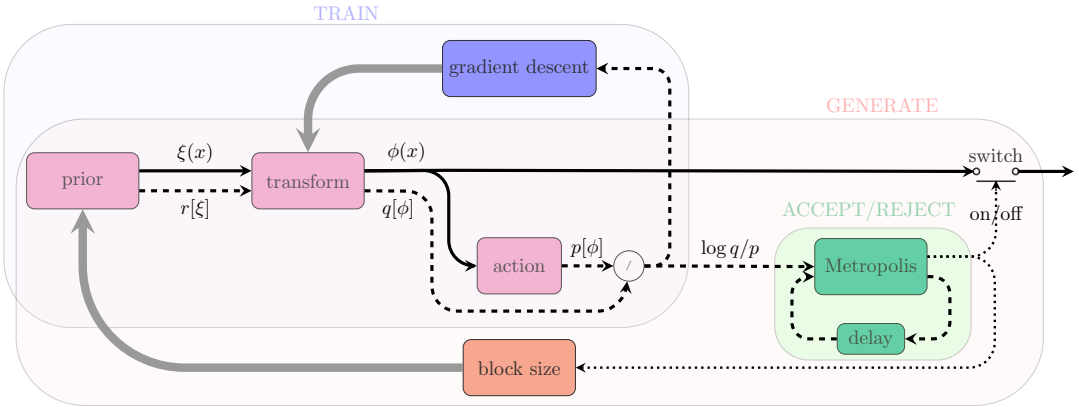




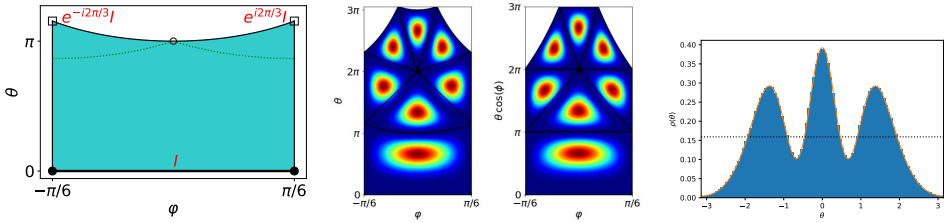


Conclusion & Outlook

- Effective theories to design layers changing the data at long&short scales
- Still, the acceptance rate drops fast as the lattice volume increases
- Suggestion: Divide&Conquer
  - Divide the current sample into blocks & update block by block
  - Optimum block size (about 1/4 or so)
  - In progress...

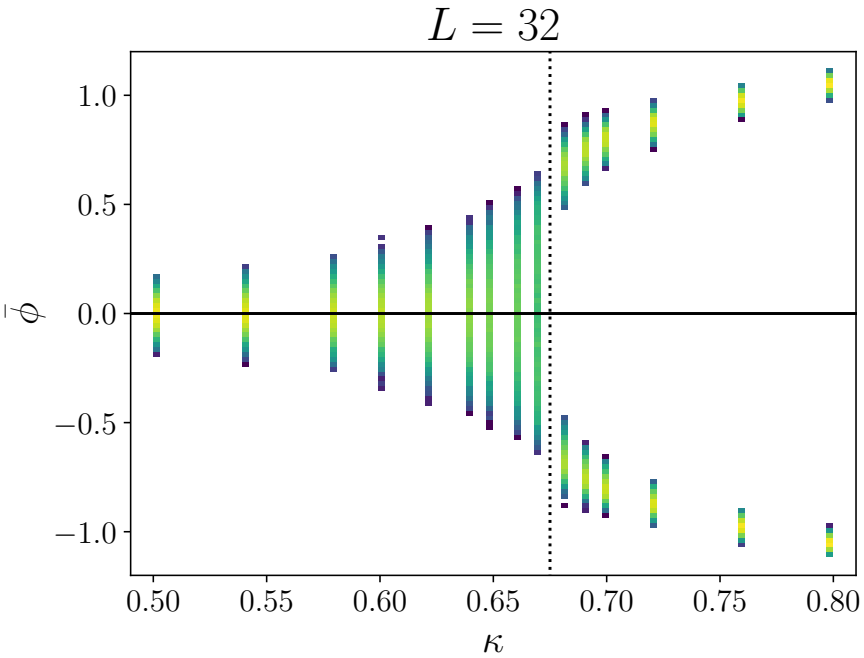


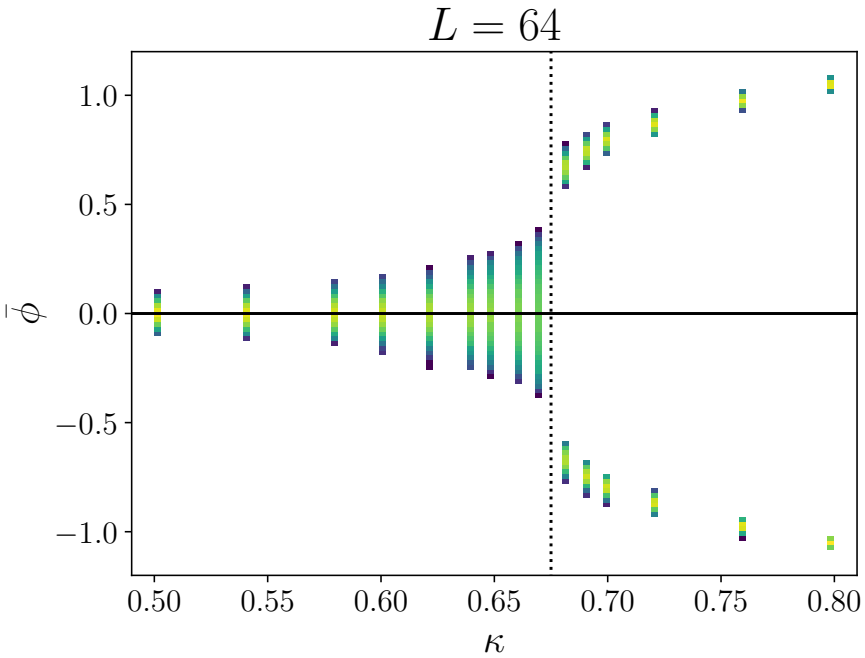
- Outlook: SU(n) gauge theories



back-up slides

Magnetization & critical point & (un)broken phase





Block updating & autocorrelation time