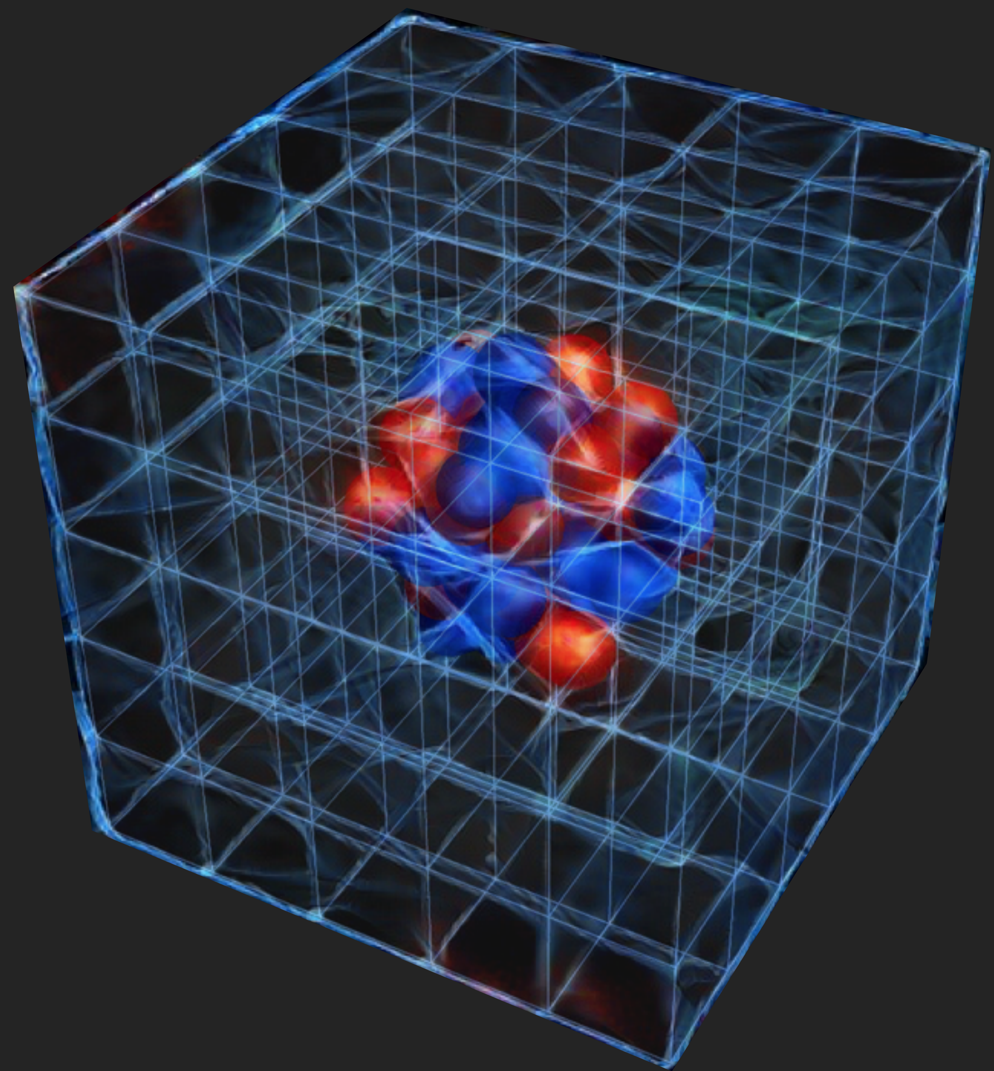
Machine learning for ensemble generation



Phiala Shanahan

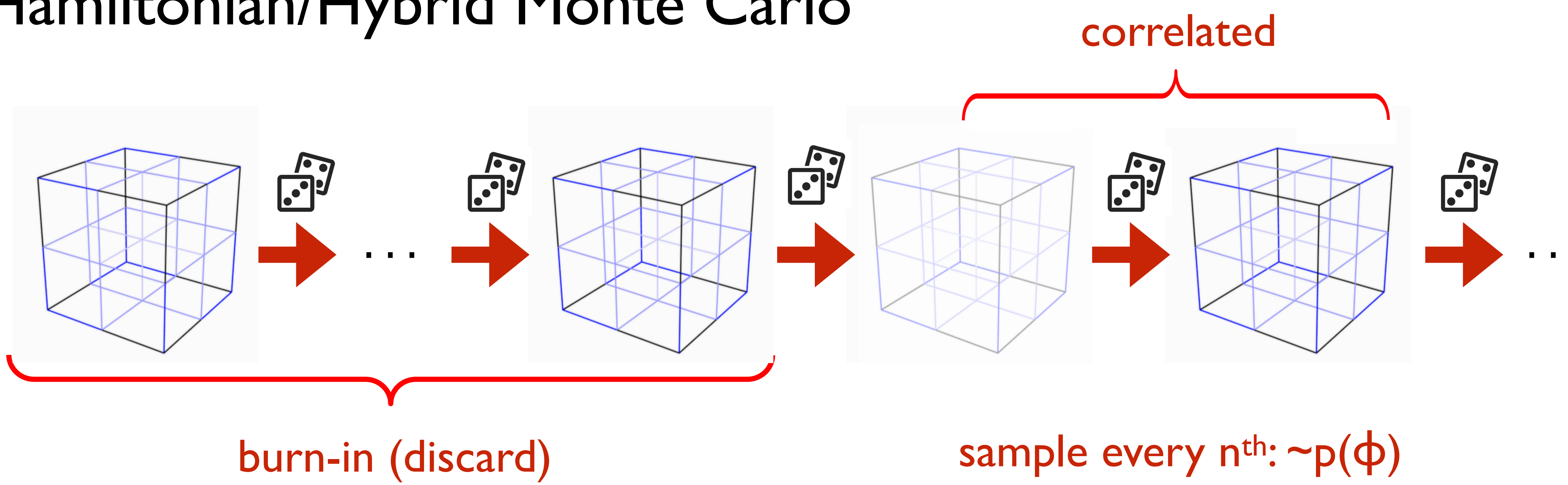
Massachusetts Institute of Technology

Generate QCD gauge fields

Generate field configurations $\phi(x)$ with probability

$$P[\phi(x)] \sim e^{-S[\phi(x)]}$$

Hamiltonian/Hybrid Monte Carlo

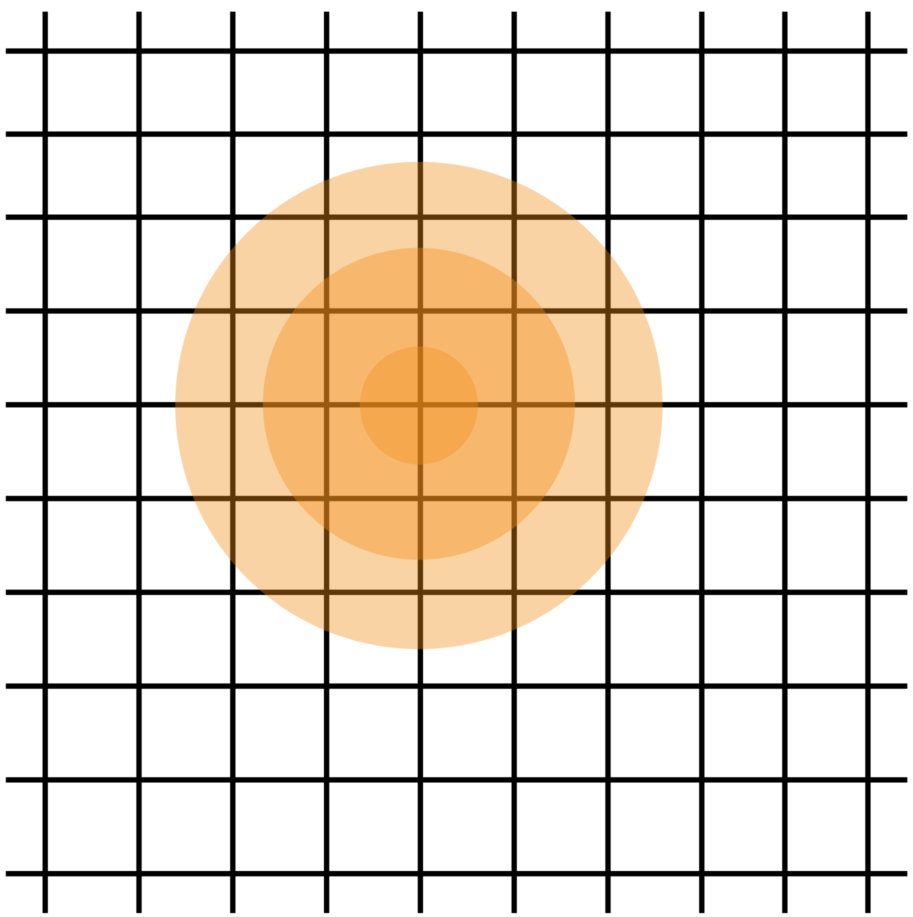


Burn-in time and correlation length dictated by Markov chain **'autocorrelation time'**: shorter autocorrelation time implies less computational cost

Generate QCD gauge fields

QCD gauge field configurations sampled via

Hamiltonian dynamics + Markov Chain Monte Carlo



Updates diffusive

Lattice spacing $\rightarrow$ 0

Number of updates to change fixed physical length scale $\rightarrow$ ∞

“Critical slowing-down” of generation of uncorrelated samples

Generate QCD gauge fields

QCD gauge field configurations sampled via

Hamiltonian dynamics + Markov Chain Monte Carlo

"Critical slowing-down" of generation of uncorrelated samples

Autocorrelation measure

$$\tau_{\mathcal{O}}^{\text{int}} = \frac{1}{2} + \lim_{\tau_{\max}\to\infty} \sum_{\tau=1}^{\tau_{\max}} \frac{\rho_{\mathcal{O}}(\tau)}{\rho_{\mathcal{O}}(0)}$$

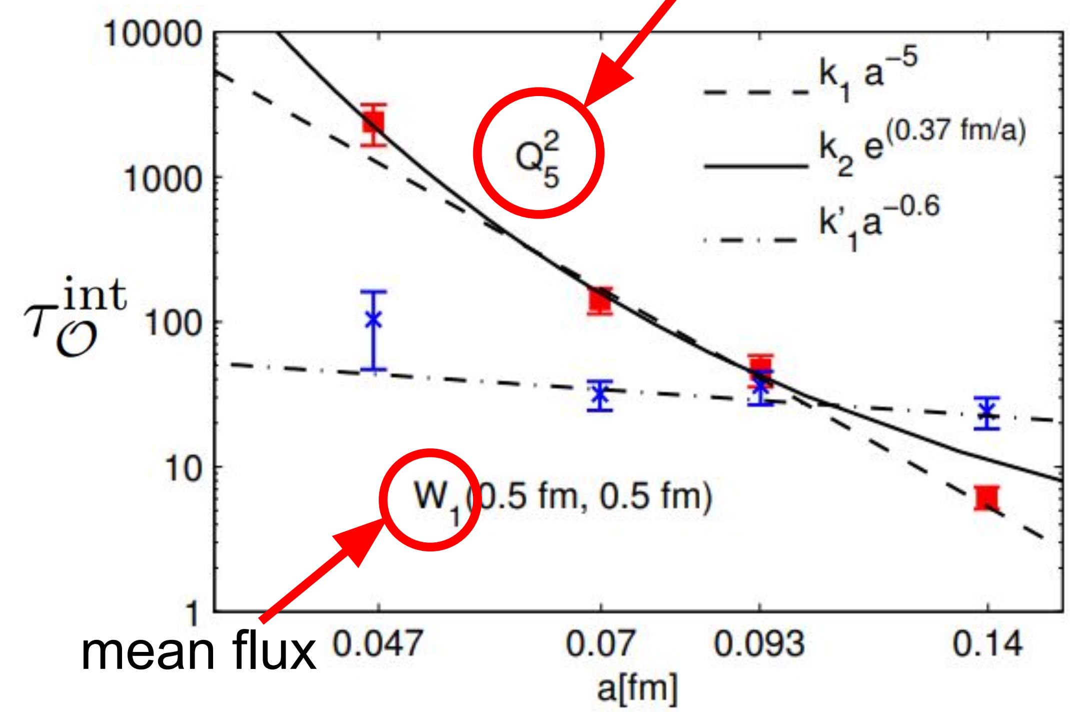
$$\tau_{\mathcal{O}}^{\text{int}} = \alpha_{\mathcal{O}} L^{z_{\mathcal{O}}}$$

Critical exponent

Correlation of observable $\mathcal{O}$ on configurations separated by τ Markov Chain steps

topological charge

[Schaefer et al. / ALPHA collaboration 1009.5228]



mean flux

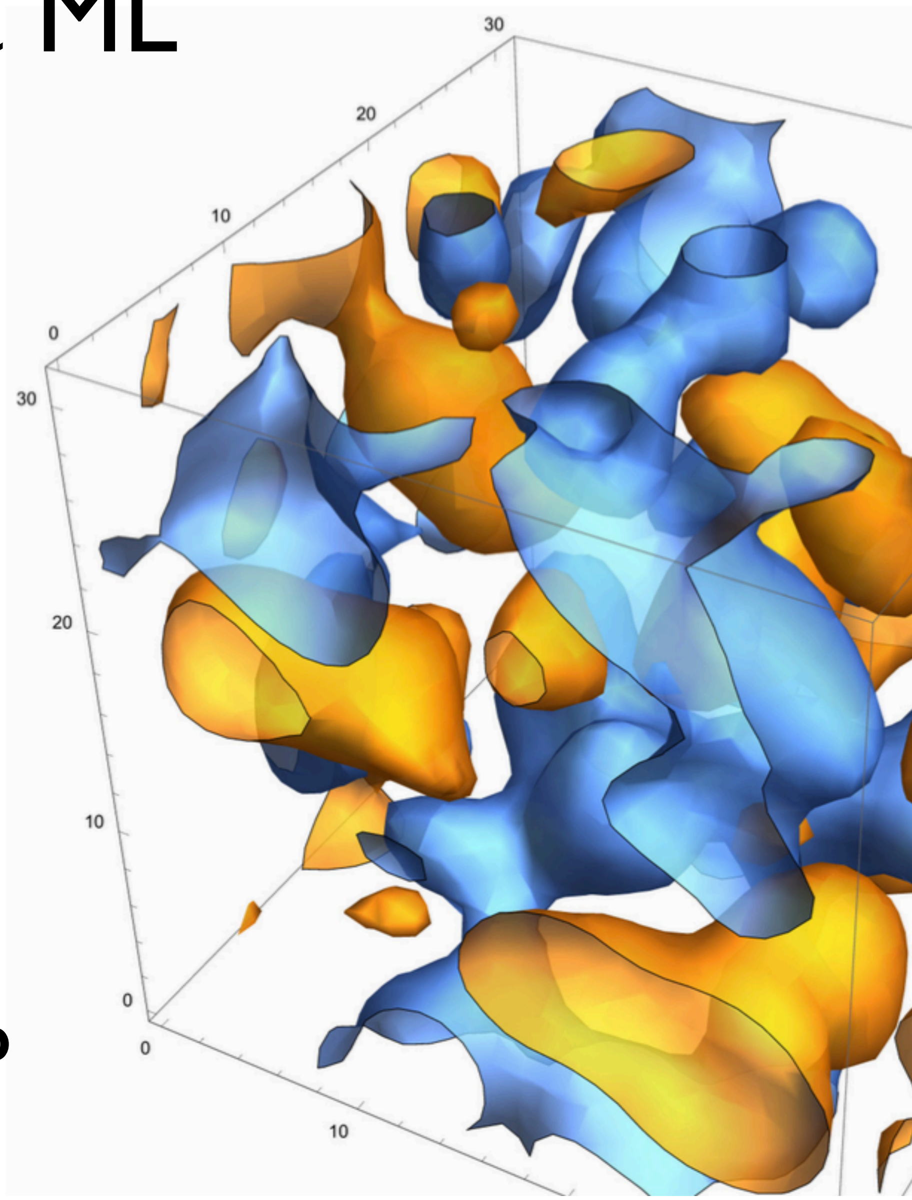
critical limit

Machine learning QCD

Accelerate gauge-field generation via ML

1. Multi-scale algorithms:
   parallels with image recognition
   Shanahan et al., PRD 97, 094506 (2018)

2. Generative models to replace Hybrid Monte-Carlo
   parallels with image generation
   Albergo et al., PRD 100, 034515 (2019)



Gurtej Kanwar (MIT)

Michael Albergo (NYU)

Consider only approaches which rigorously preserve quantum field theory in applicable limits

Machine learning QCD

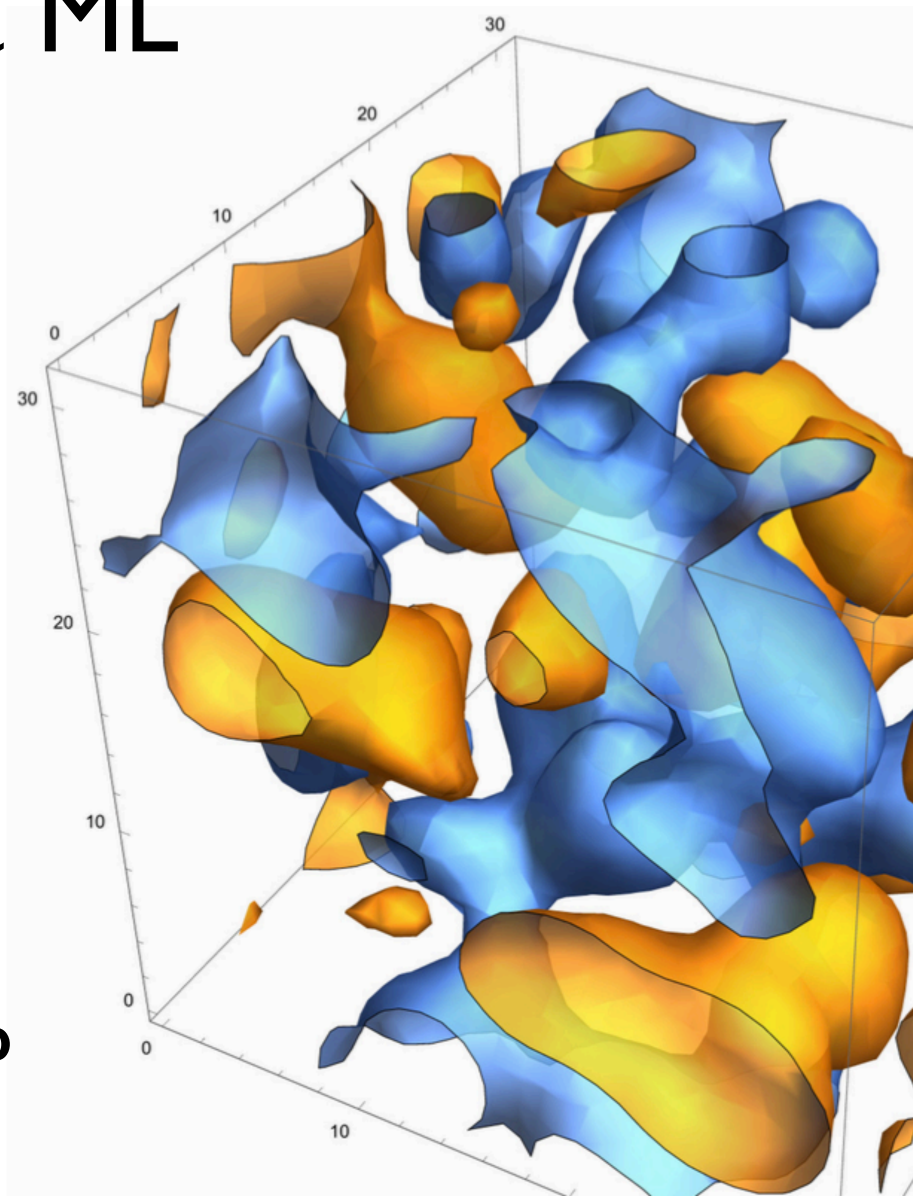
Accelerate gauge-field generation via ML

1. Multi-scale algorithms:
   parallels with image recognition
   Shanahan et al., PRD 97, 094506 (2018)

2. Generative models to replace Hybrid Monte-Carlo
   parallels with image generation
   Albergo et al., PRD 100, 034515 (2019)

Gurtej Kanwar (MIT)

Michael Albergo (NYU)

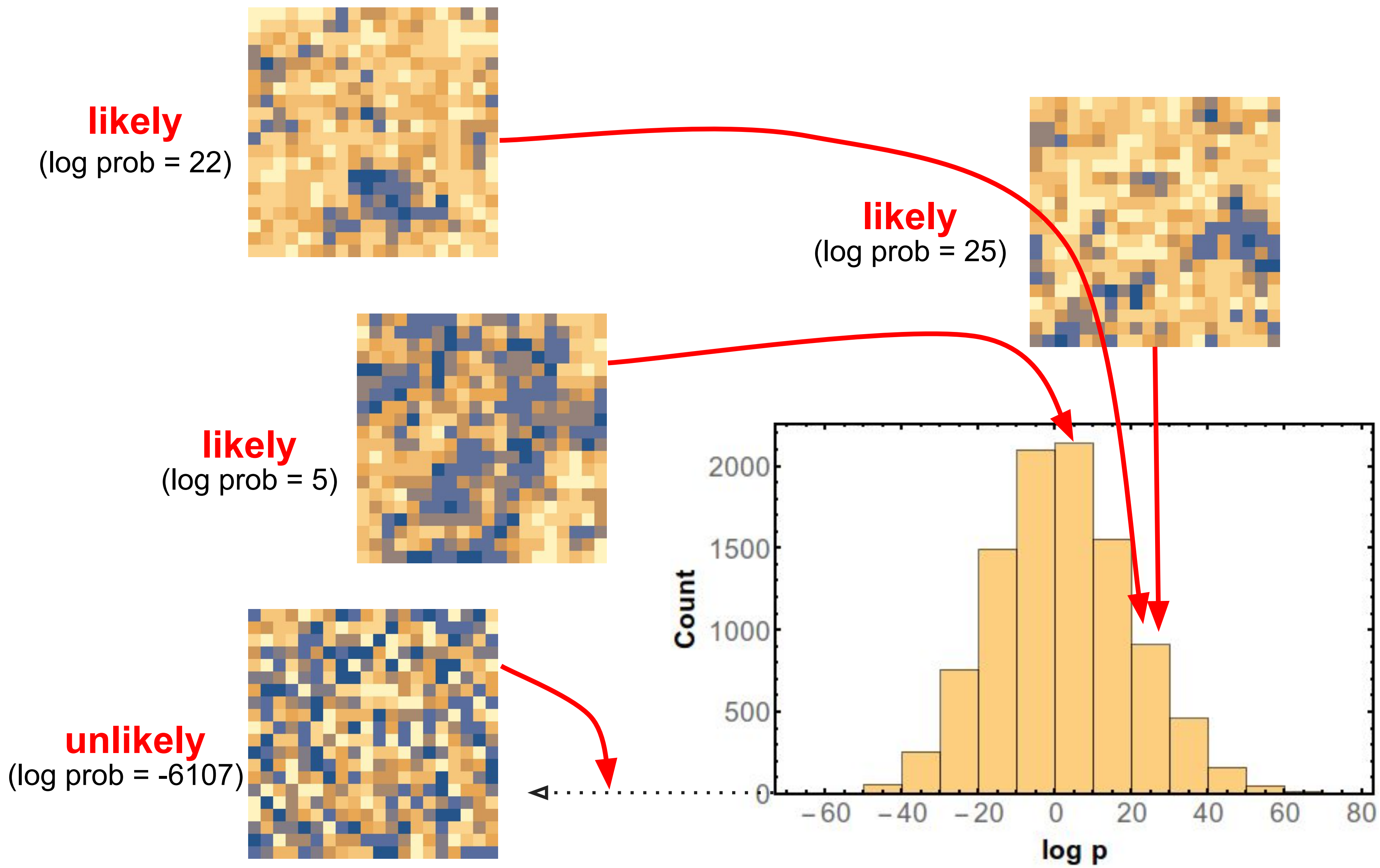


Consider only approaches which rigorously preserve quantum field theory in applicable limits

Sampling gauge field configs

Generate field configurations $\phi(x)$ with probability

$$P[\phi(x)] \sim e^{-S[\phi(x)]}$$



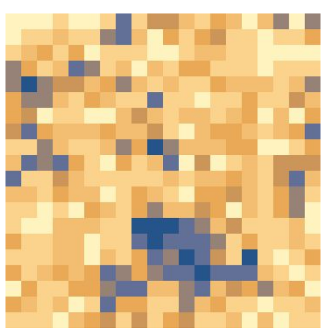
Sampling gauge field configs

Generate field configurations $\phi(x)$ with probability

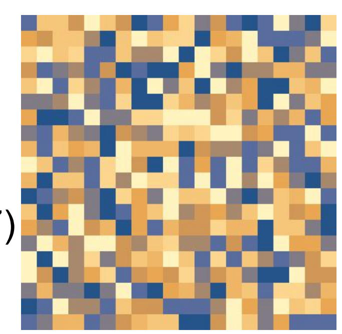
$$P[\phi(x)] \sim e^{-S[\phi(x)]}$$

Parallels with image generation problem

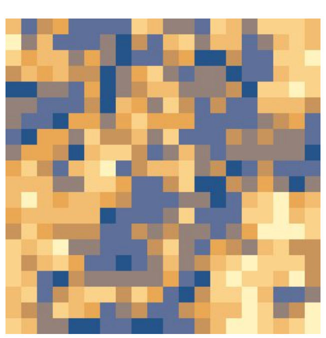
likely
(log prob = 22)



unlikely
(log prob = -6107)



likely
(log prob = 5)

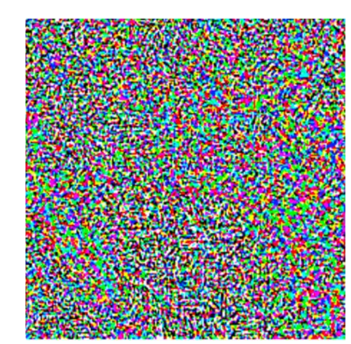


likely

[Karras, Lane, Aila / NVIDIA 1812.04948]

likely

unlikely



Sampling gauge field configs

- **Probability density can be computed for a given sample** (up to normalization) $p(..) = e^{-S(...)}/Z$

- **Distribution of gauge fields has precise symmetries**
  - Lattice symmetries (translation, rotation, reflection)
  - Internal symmetries (gauge symmetries mixing field components)

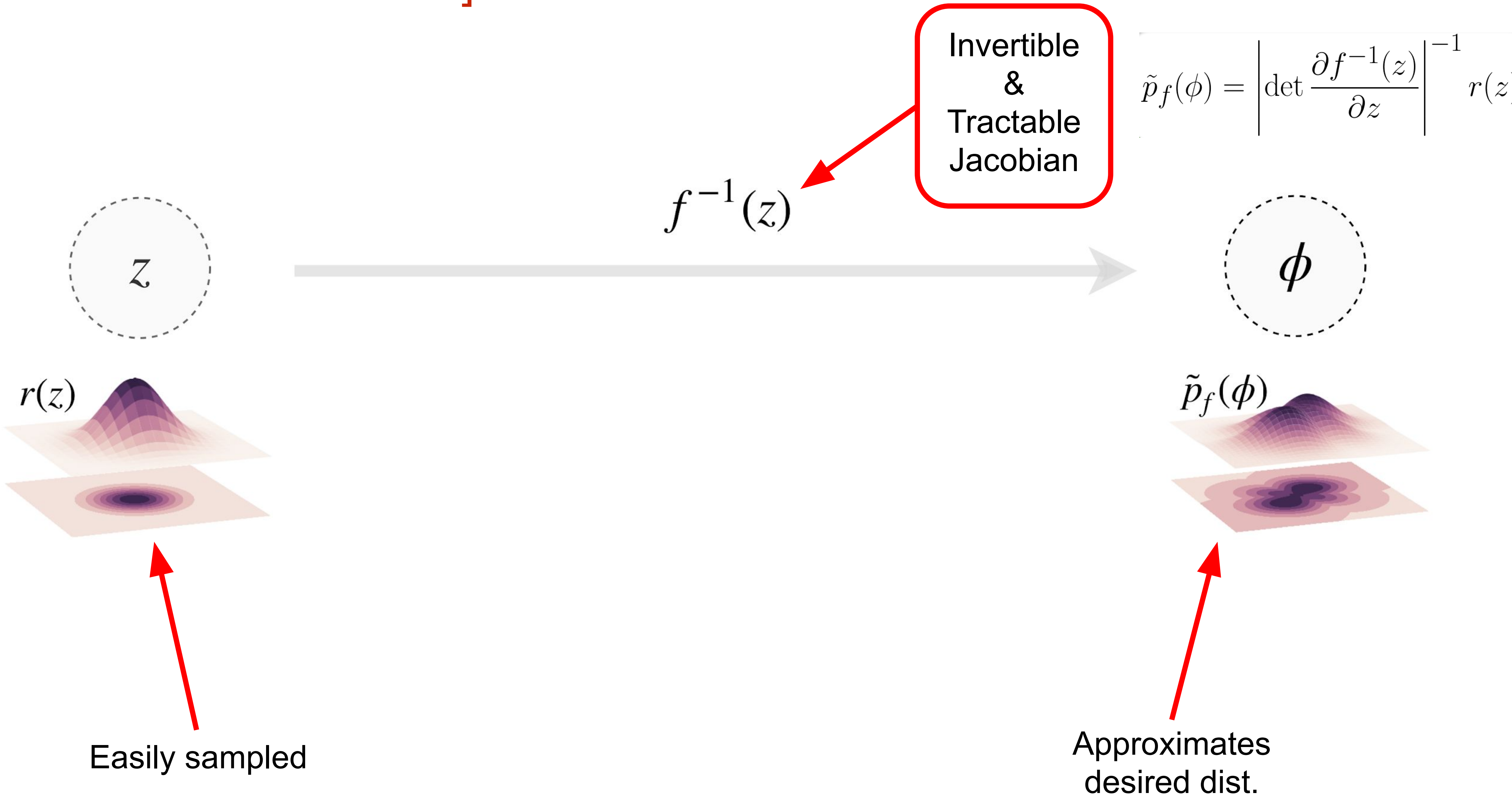
- **Data hierarchies are challenging**
  - 10^9 to 10^{12} variables per configuration
  - O(1000), samples available (fewer than # degrees of freedom per config)

  $\Rightarrow$ **Hard to use training paradigms that rely on existing samples from distribution**

Generative flow models

Flow-based models learn a change-of-variables that transforms a known distribution to the desired distribution

[Rezende & Mohamed 1505.05770]

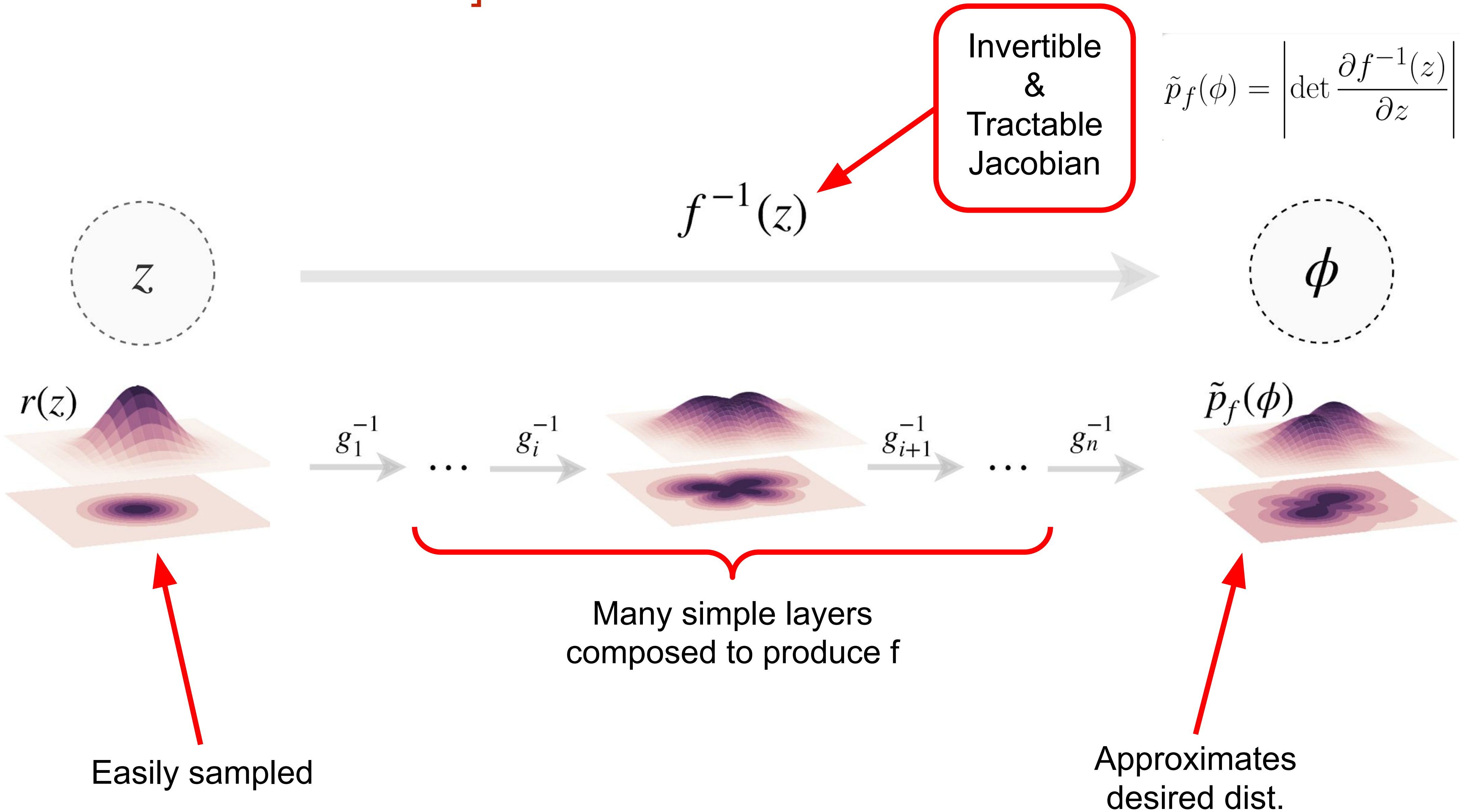


Generative flow models

Flow-based models learn a change-of-variables that transforms a known distribution to the desired distribution

[Rezende & Mohamed 1505.05770]

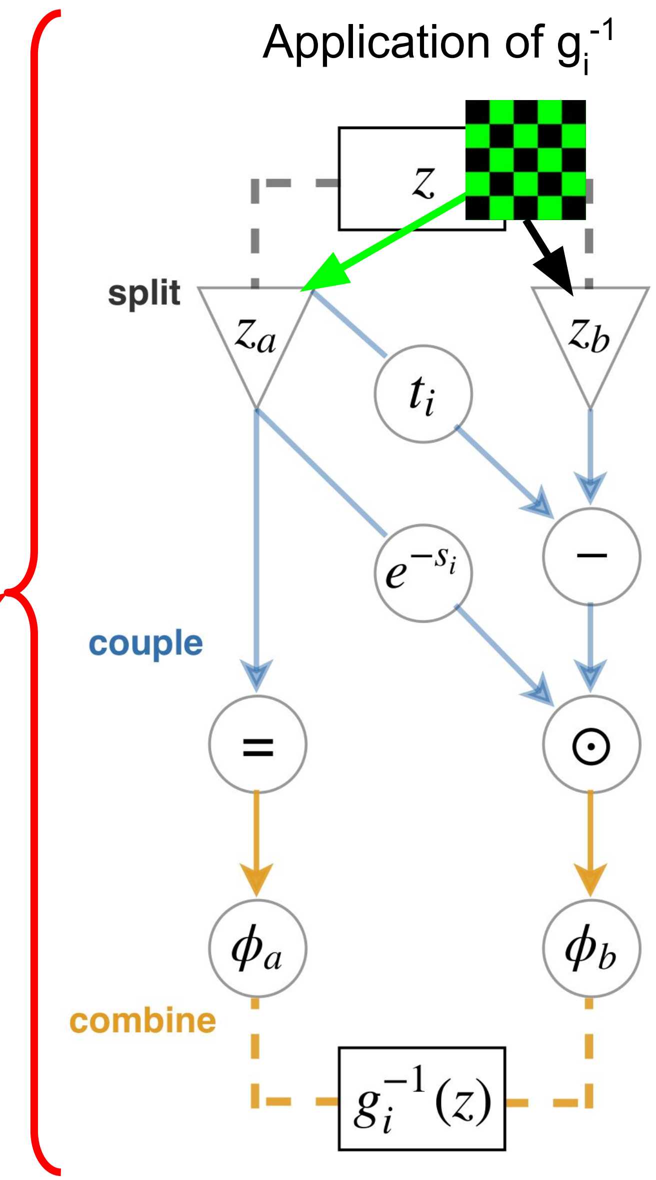
$$\tilde{p}_f(\phi) = \left|\det \frac{\partial f^{-1}(z)}{\partial z}\right|^{-1} r(z)$$

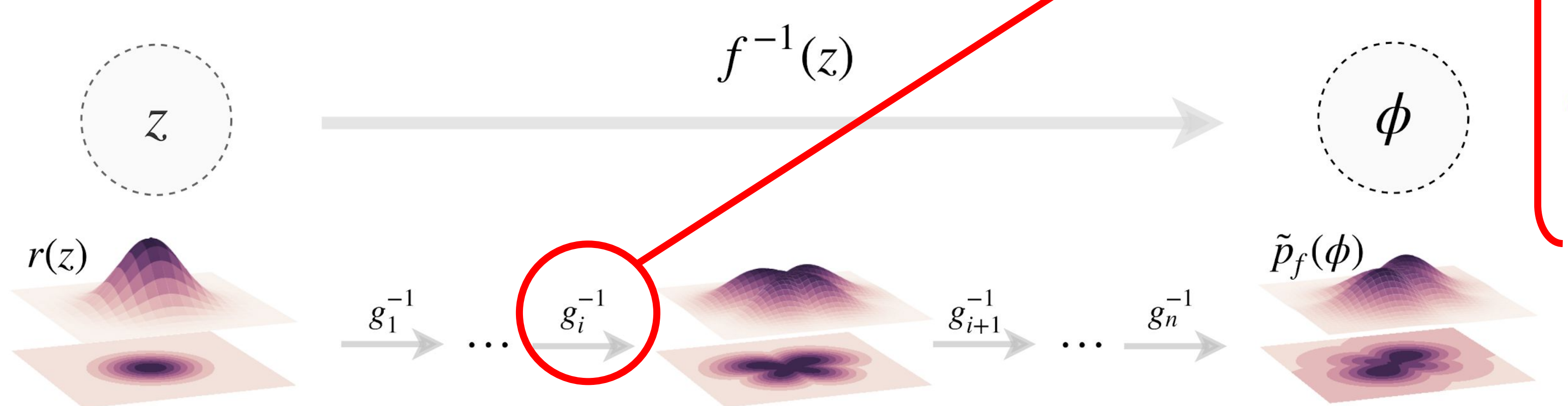


Generative flow models

Choose real non-volume preserving flows:
[Dinh et al. 1605.08803]

- Affine transformation of half of the variables:
  - scaling by exp(s)
  - translation by t
  - s and t arbitrary neural networks depending on untransformed variables only
- Simple inverse and Jacobian



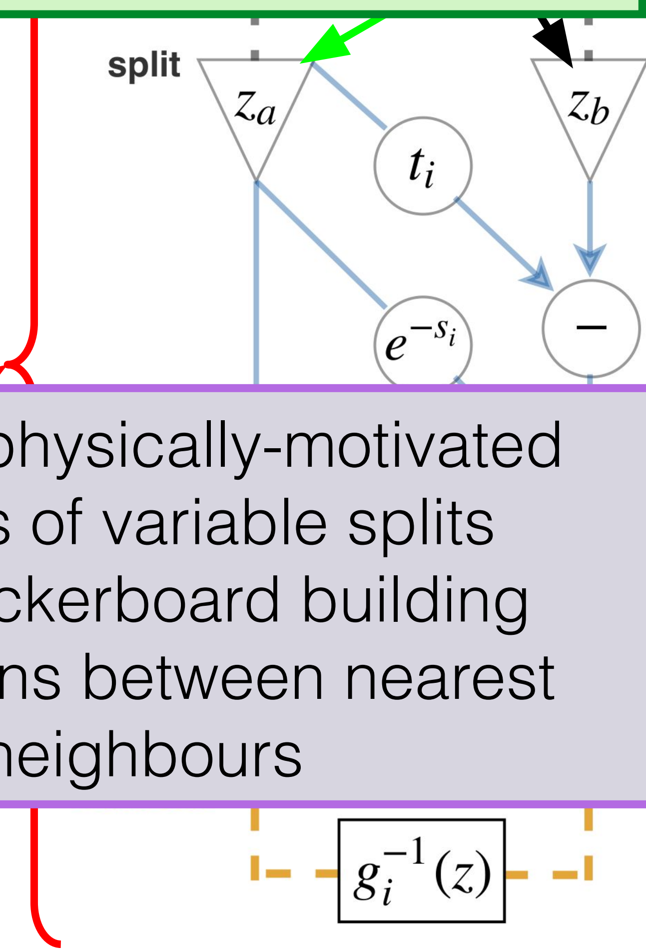


Generative flow models

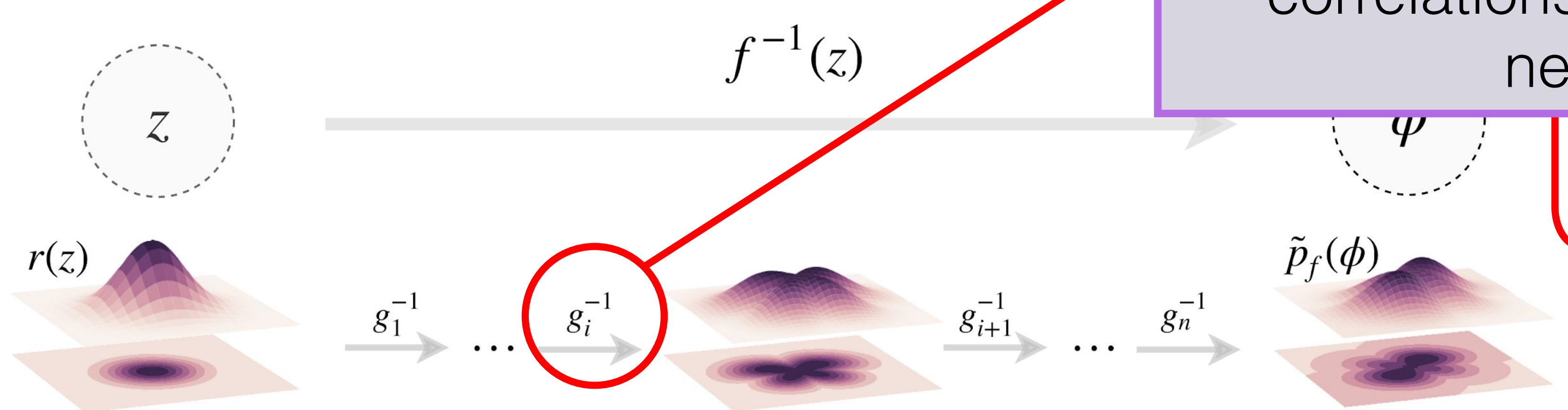
Choose real **non-volume preserving** flows:
[Dinh et al. 1605.08803]

- Affine transformation of **half of the variables**:
  - scaling by exp(s)
  - translation by t
  - s and t arbitrary neural networks depending on untransformed variables only
- Simple inverse and Jacobian

Density can be squished/stretched by change-of-variables



Can use physically-motivated choices of variable splits e.g. checkerboard building correlations between nearest neighbours



Training the model

Target distribution is known up to normalisation

$$p(\phi) = e^{-S(\phi)}/Z$$

Train to minimise shifted KL divergence: [Zhang, E, Wang 1809.10188]

shift removes unknown normalisation Z

$$\begin{aligned} L(\tilde{p}_f) &:= D_{KL}(\tilde{p}_f || p) - \log Z \\ &= \underbrace{\int \prod_j d\phi_j \, \tilde{p}_f(\phi)} \left(\log \tilde{p}_f(\phi) + S(\phi) \right) \end{aligned}$$

allows **self-training**: sampling with respect to model distribution $\tilde{p}_f(\phi)$ to estimate loss

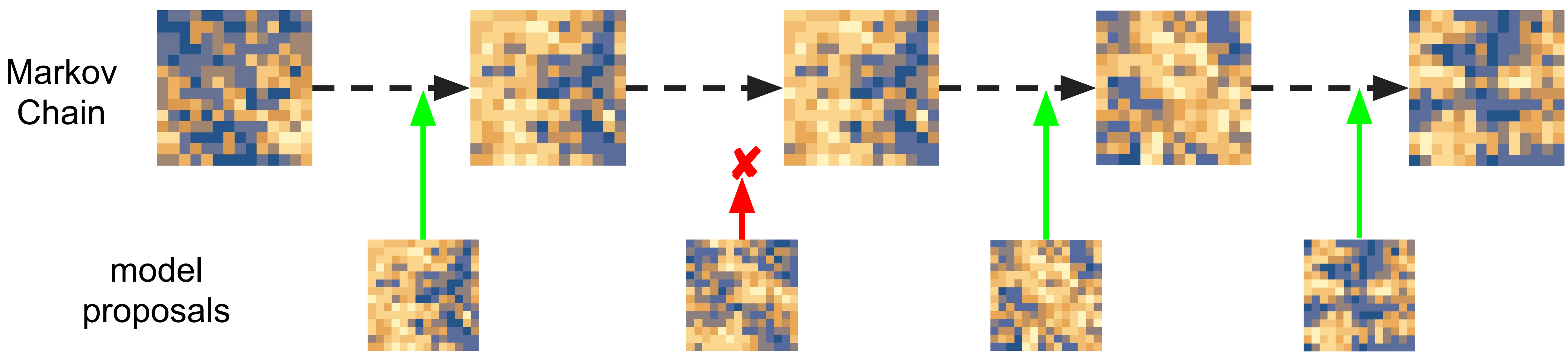
Exactness via Markov chain

Guarantee exactness of generated distribution by forming a Markov chain: accept/reject with Metropolis-Hastings step

Acceptance probability

$$A(\phi^{(i-1)}, \phi') = \min\left(1, \frac{\tilde{p}(\phi^{(i-1)})}{p(\phi^{(i-1)})} \frac{p(\phi')}{\tilde{p}(\phi')}\right)$$

proposal independent of previous sample



Exactness via Markov chain

Guarantee exactness of generated distribution by forming a Markov chain: accept/reject with Metropolis-Hastings step

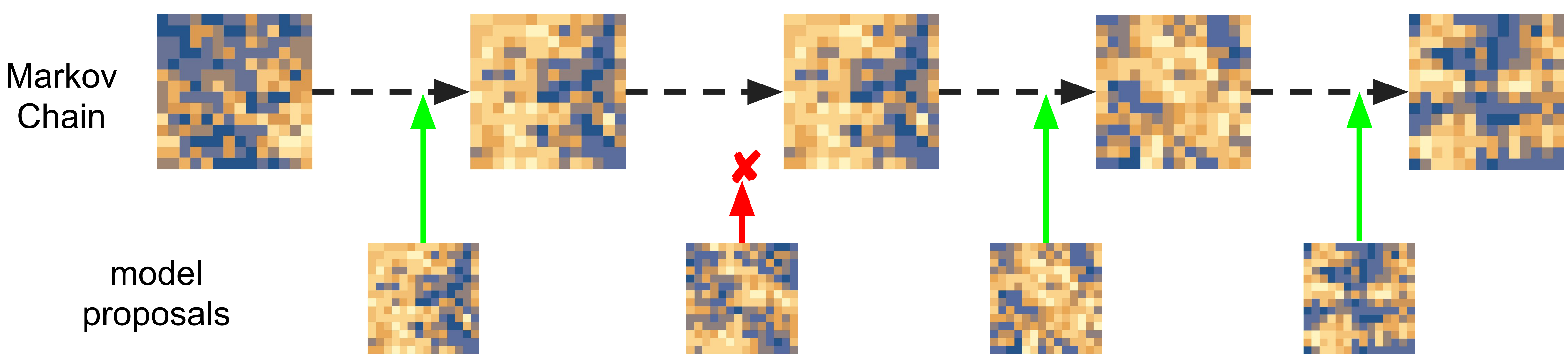
Acceptance probability

$$A(\phi^{(i-1)}, \phi') = \min\left(1, \frac{\boxed{\tilde{p}(\phi^{(i-1)})}}{\boxed{p(\phi^{(i-1)})}} \frac{\boxed{p(\phi')}}{\boxed{\tilde{p}(\phi')}}\right)$$

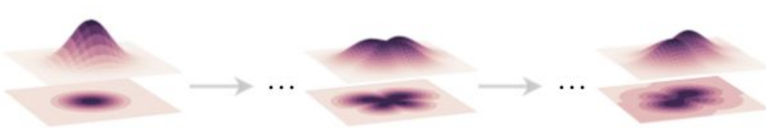
True dist

Model dist

proposal independent
of previous sample

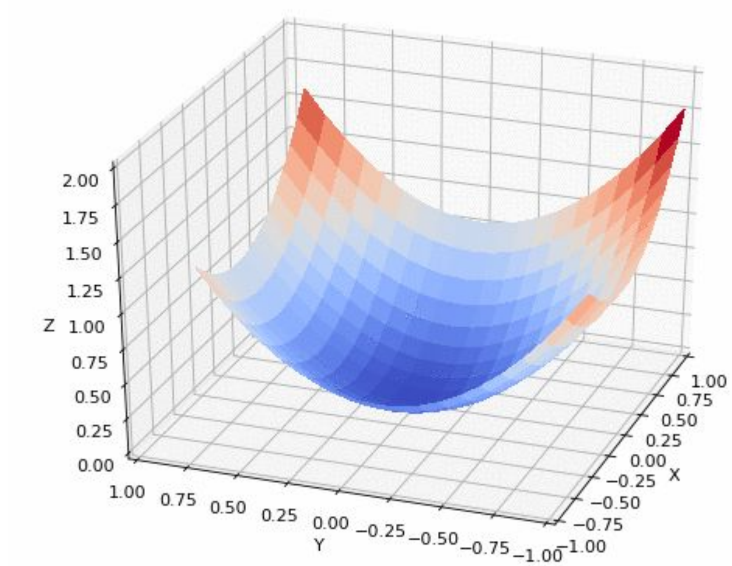


Fields via flow models



Parameterize flow using Real NVP coupling layers

Each layer contains arbitrary neural nets **s** and **t**



Training step

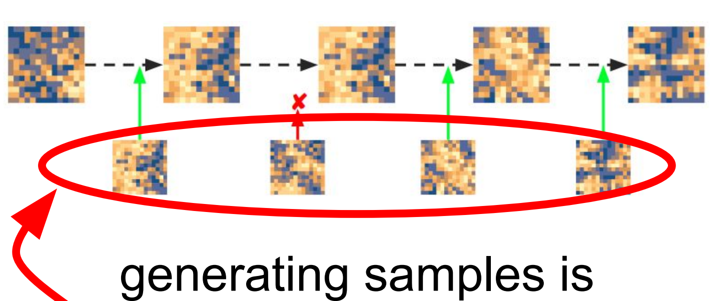
Draw samples from model

Compute loss function

Gradient descent

Desired accuracy?

Save trained model



generating samples is "embarrassingly parallel"

Markov chain using samples from model

Summary chart: Tej Kanwar

Application: scalar field theory

First application: scalar lattice field theory

- One real number $\phi(x) \in (-\infty, \infty)$ per lattice site x (2D lattice)
- Action: kinetic terms and quartic coupling

$$S(\phi) = \sum_x \left(\sum_y \frac{1}{2}\phi(x)\Box(x,y)\phi(y) + \frac{1}{2}m^2\phi(x)^2 + \lambda\phi(x)^4 \right)$$

5 lattice sizes: $L^2 = \{6^2, 8^2, 10^2, 12^2, 14^2\}$ with parameters tuned for analysis of critical slowing down

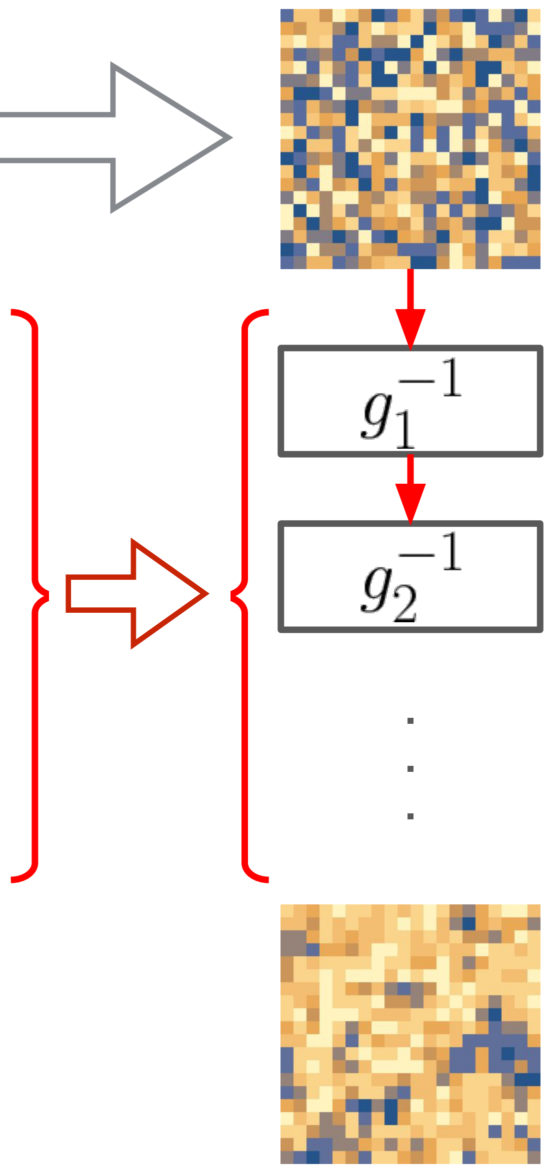
	E1	E2	E3	E4	E5
L	6	8	10	12	14
m^2	-4	-4	-4	-4	-4
λ	6.975	6.008	5.550	5.276	5.113
$m_p L$	3.96(3)	3.97(5)	4.00(4)	3.96(5)	4.03(6)

Application: scalar field theory

First application: scalar lattice field theory

- Prior distribution chosen to be uncorrelated Gaussian:

$$\phi(x) \sim \mathcal{N}(0,1)$$

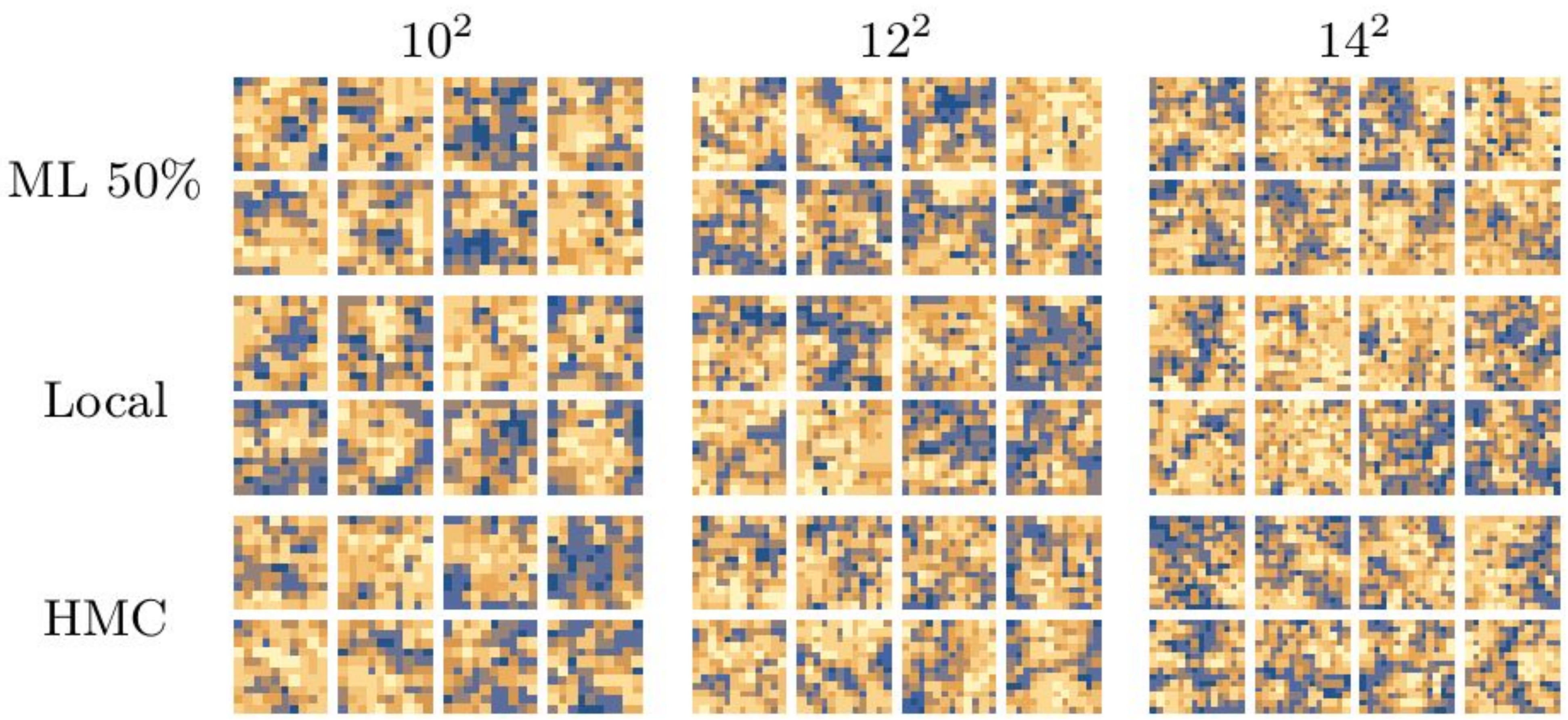


- Real non-volume-preserving (NVP) couplings
  - 8-12 Real NVP coupling layers
  - Alternating checkerboard pattern for variable split
  - NNs with 2-6 fully connected layers with 100-1024 hidden units
- Train using shifted KL loss with Adam optimizer
  - Stopping criterion: fixed acceptance rate in Metropolis-Hastings MCMC

Application: scalar field theory

First application: scalar lattice field theory

Compare with standard updating algorithms: 'local', 'HMC'

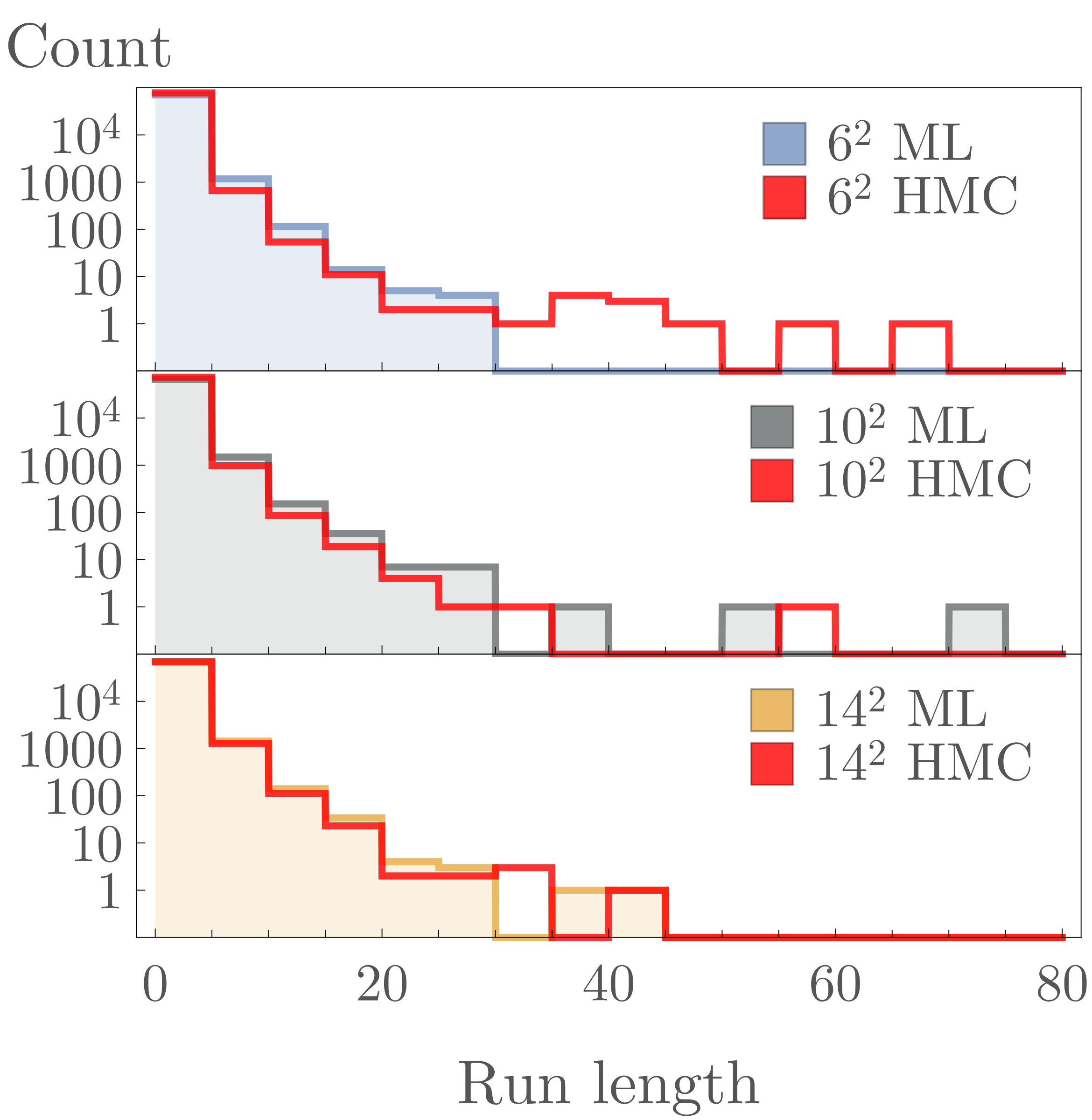


ML model produces **varied samples** and **correlations at the right scale**

Application: scalar field theory

First application: scalar lattice field theory

Compare with standard updating algorithms: 'local', 'HMC'

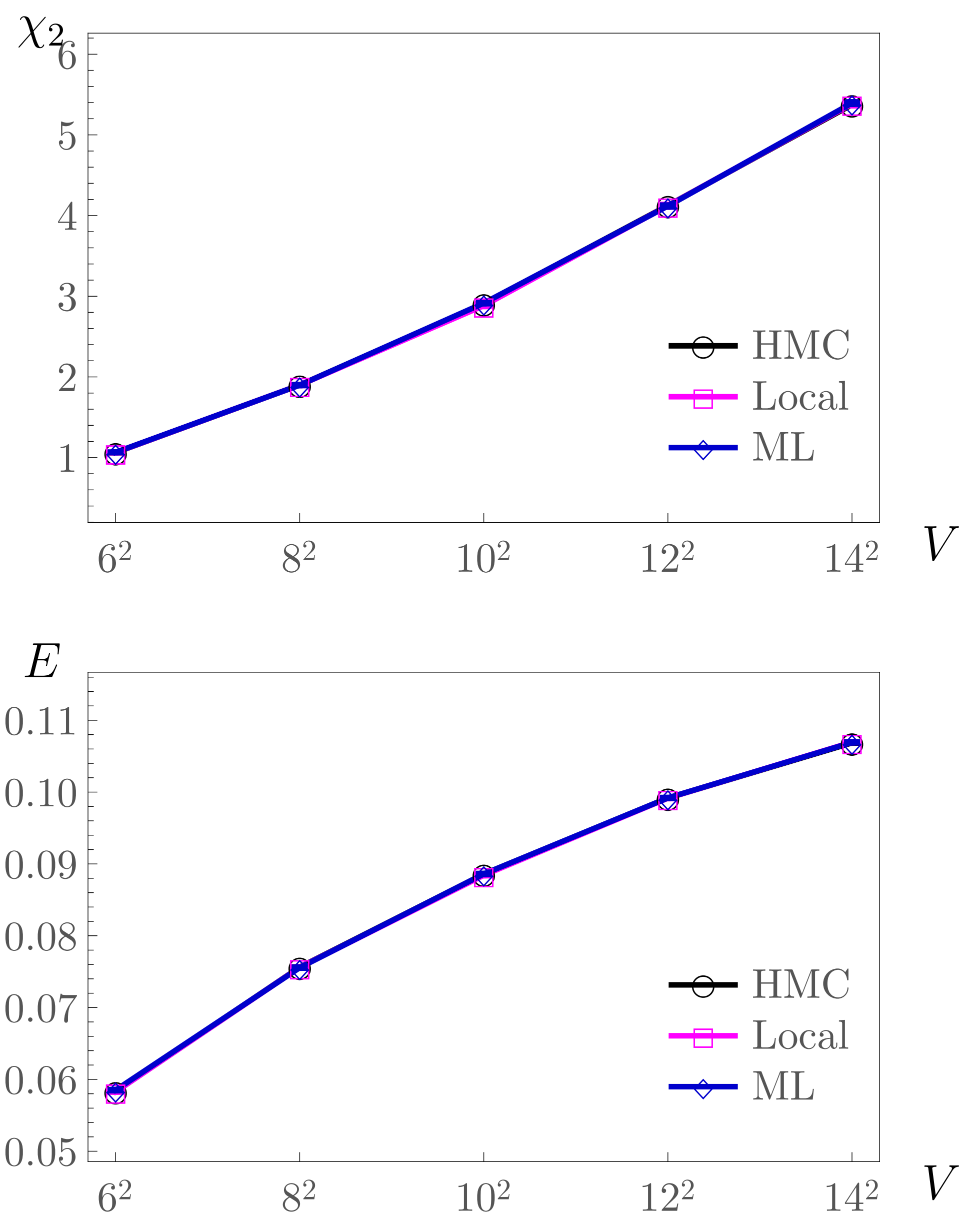


Rejectance runs in the Metropolis-Hastings accept/reject step are **comparable to those in Hamiltonian Monte-Carlo** tuned to same acceptance

Application: scalar field theory

First application: scalar lattice field theory

Compare with standard updating algorithms: 'local', 'HMC'



Physical observables match computed on ensembles generated from ML model and from standard methods

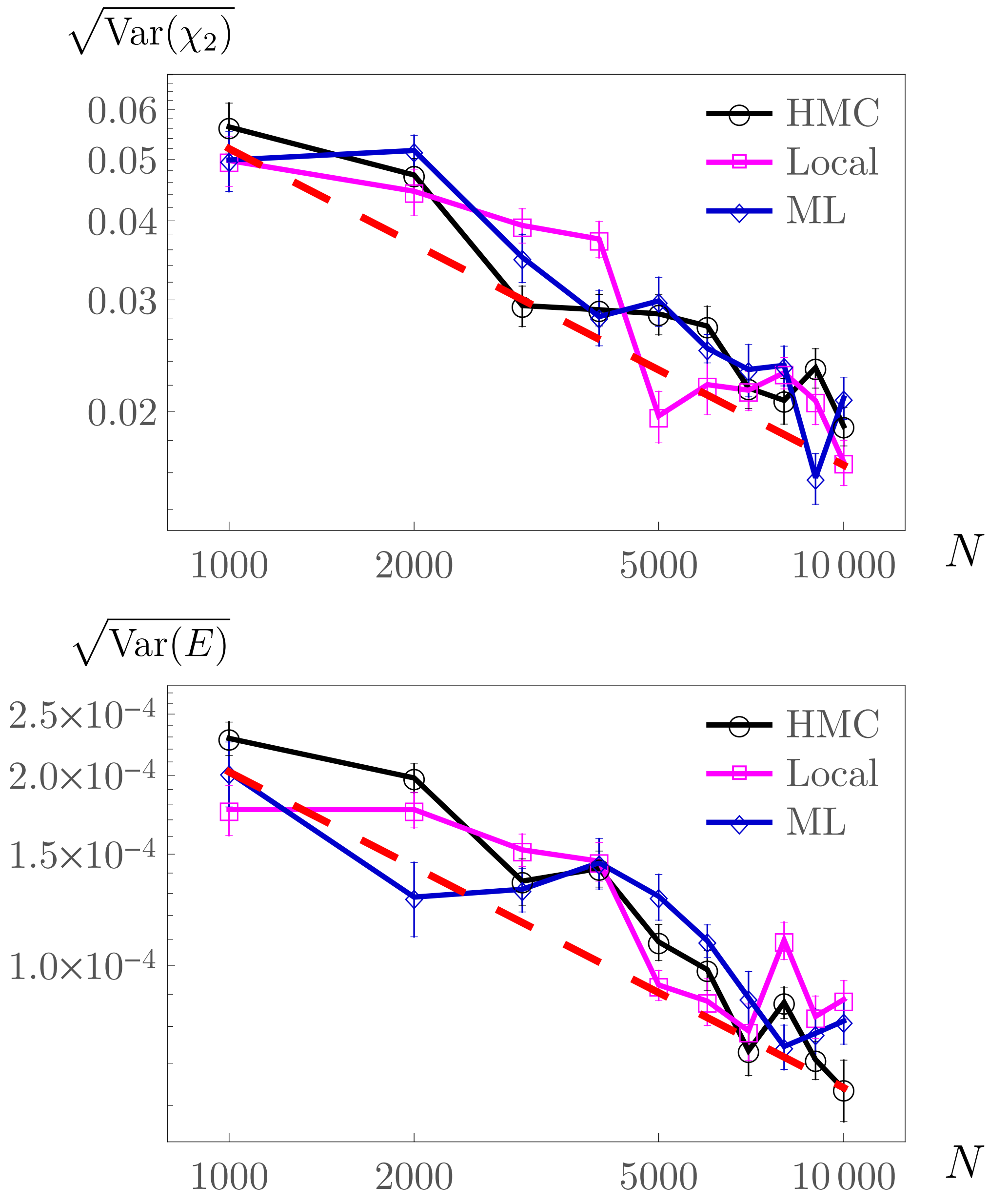
Two-point susceptibility $\chi_2 = \sum_x G_c(x)$

Ising limit energy $E = \frac{1}{d} \sum_{1 \leq \mu \leq d} G_c(\hat{\mu})$

Application: scalar field theory

First application: scalar lattice field theory

Compare with standard updating algorithms: 'local', 'HMC'



Uncertainties in physical observables follow statistical scaling as the number of samples is increased

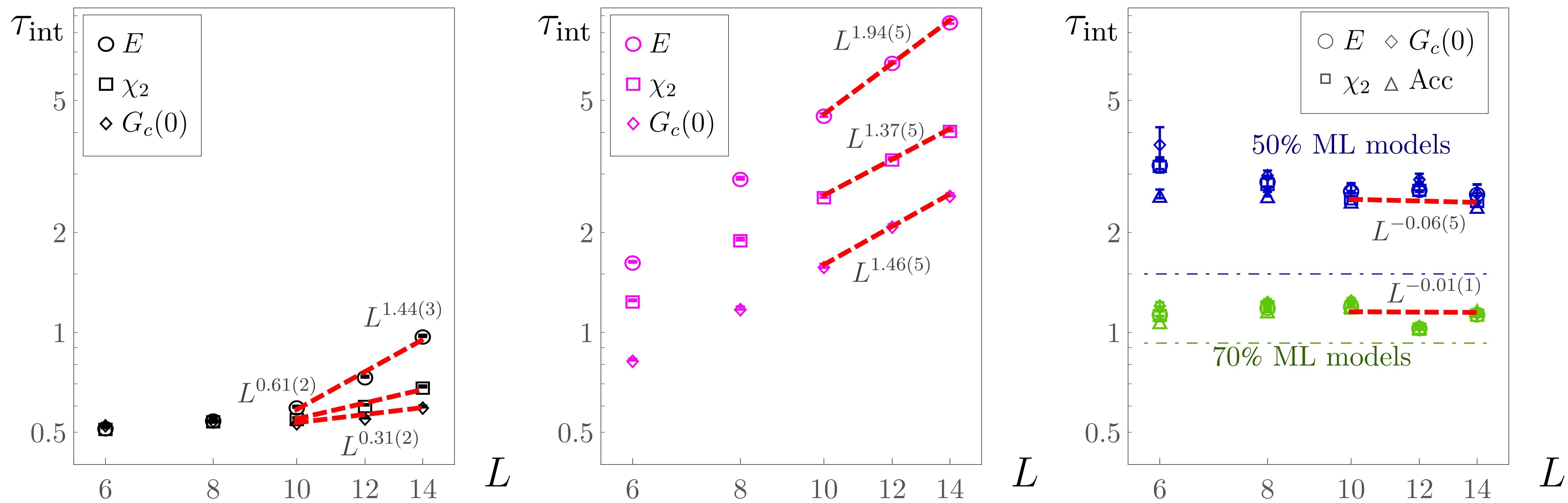
red dashed curve:

$$\propto 1/\sqrt{N}$$

Application: scalar field theory

First application: scalar lattice field theory

Success: Critical slowing down is eliminated
Cost: Up-front training of the model



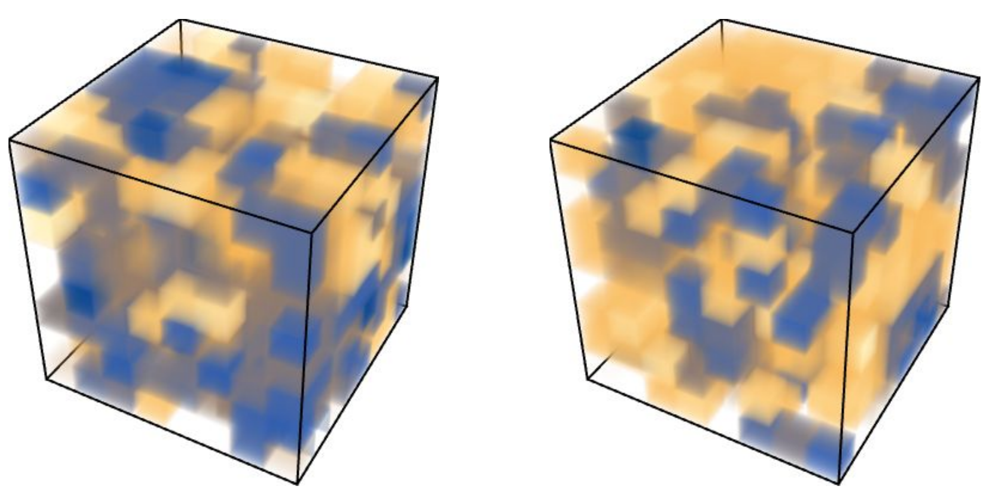
(a) HMC ensembles (b) Local Metropolis ensembles (c) Flow-based MCMC ensembles

Dynamical critical exponents consistent with zero

Next steps

Target application: LQCD

1. Scale number of dimensions
2. Scale number of degrees of freedom
3. Methods for gauge theories



Google

DeepMind

Outlook

IF a generative flow model can be trained for QCD

After the up-front cost of training the model, it is

- Cheap to generate an arbitrarily large ensemble
- No need to store configurations, only the trained model
- Volume scaling is ~free via hierarchical flow and transfer learning approach
- Cheap to re-train the model to move to nearby parameter values (quark masses, beta)

i.e., if possible, this approach would have significant advantages, even if initial training is expensive