# Generative Machine Learning

for

# Lattice Gauge Theories

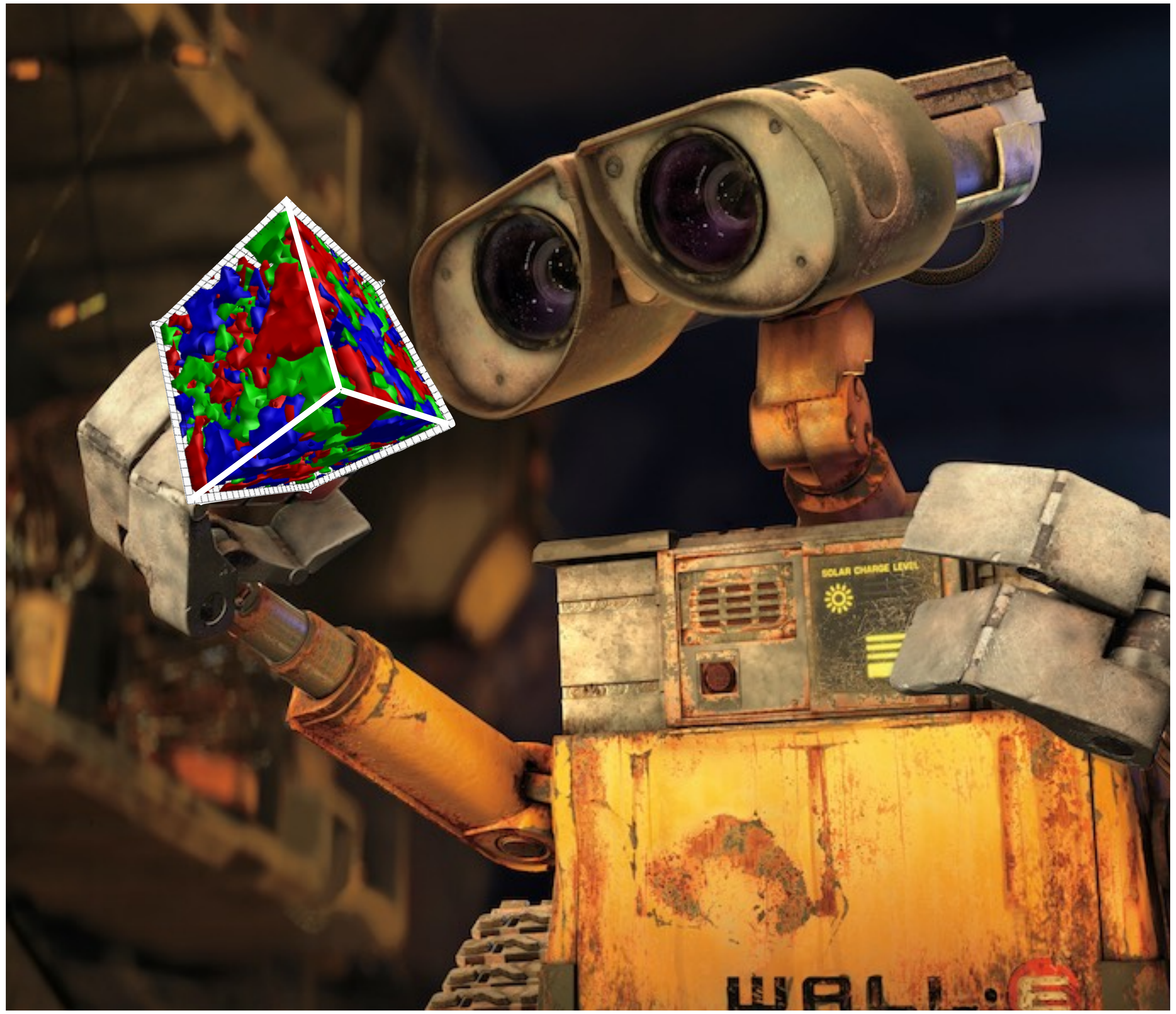


Image credit: Leinweber, Pixar

**Gurtej Kanwar**
Chancellor's Fellow in AI & Datascience
University of Edinburgh

**June 12, 2025**
**Nikhef Theory Seminar**

# Quantum field theories on lattices

*Discretized spacetime (spacing $a$) →*
***non-perturbative*, *gauge-invariant* UV regulator ~ $a^{-1}$.**

- Needed for theories at strong couplings
  - Strong nuclear force (QCD) at low energies
  - Strongly interacting BSM theories

- Numerical simulation to estimate observables
  - **Lattice QCD**: decades of algorithms and software development, execution at **extreme scale**

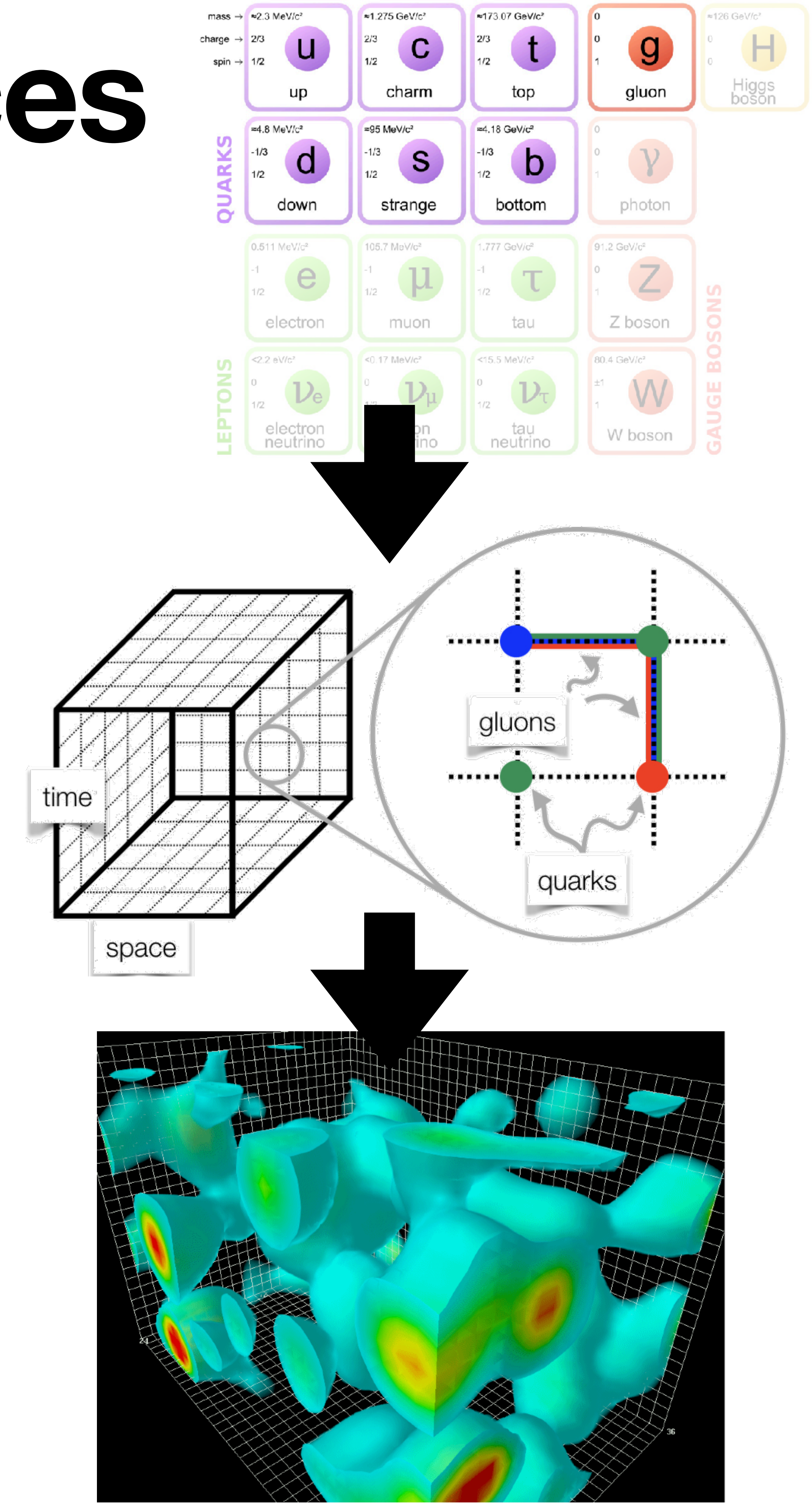



Image credit: Berkowitz, Leinweber

# Quantum field theories on lattices

*Discretized spacetime (spacing $a$) →*
***non-perturbative*, *gauge-invariant* UV regulator ~ $a^{-1}$.**

- Needed for theories at strong couplings
  - Strong nuclear force (QCD) at low energies
  - Strongly interacting BSM theories

- Numerical simulation to estimate observables
  - **Lattice QCD**: decades of algorithms and software development, execution at **extreme scale**

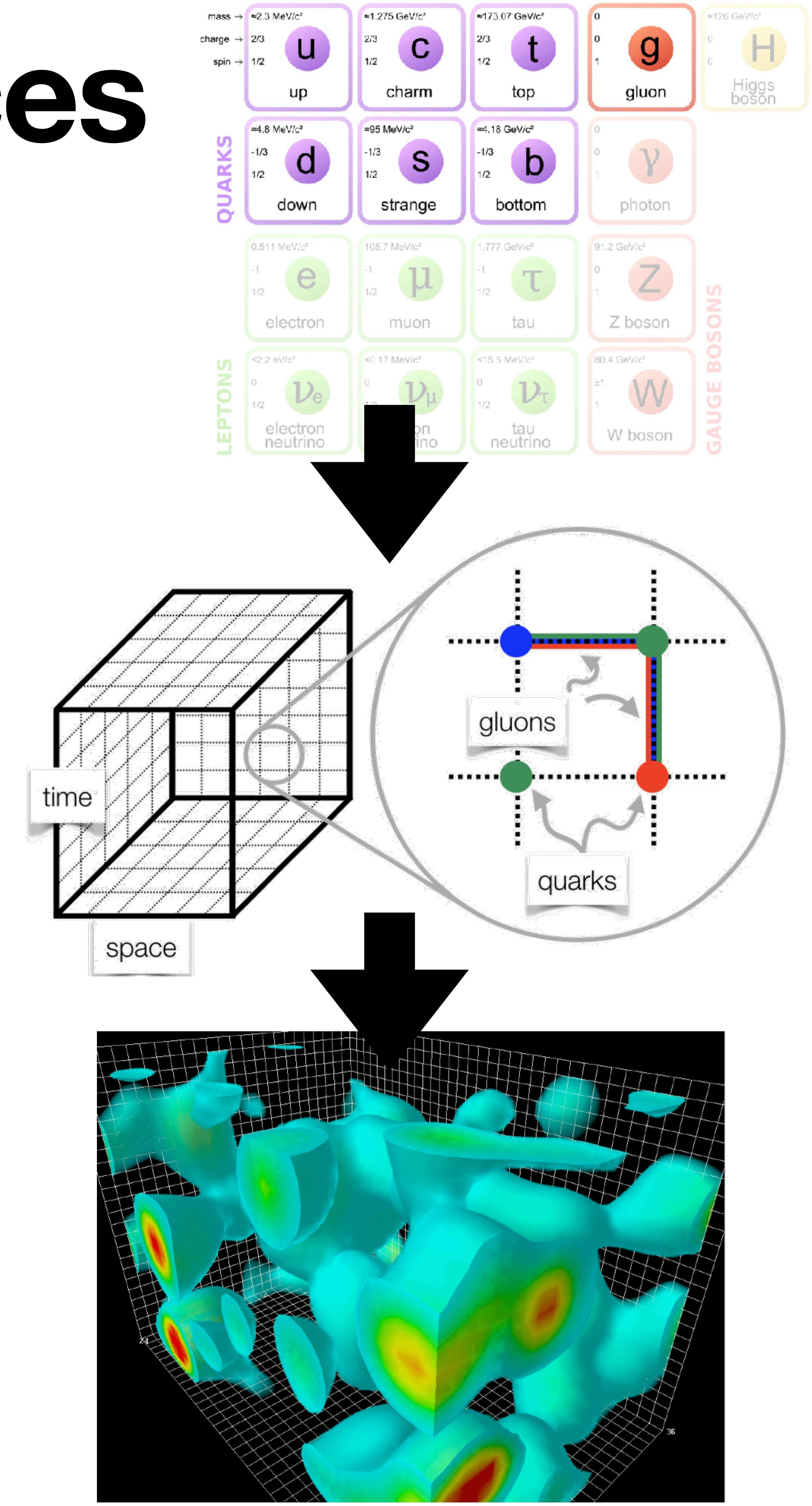
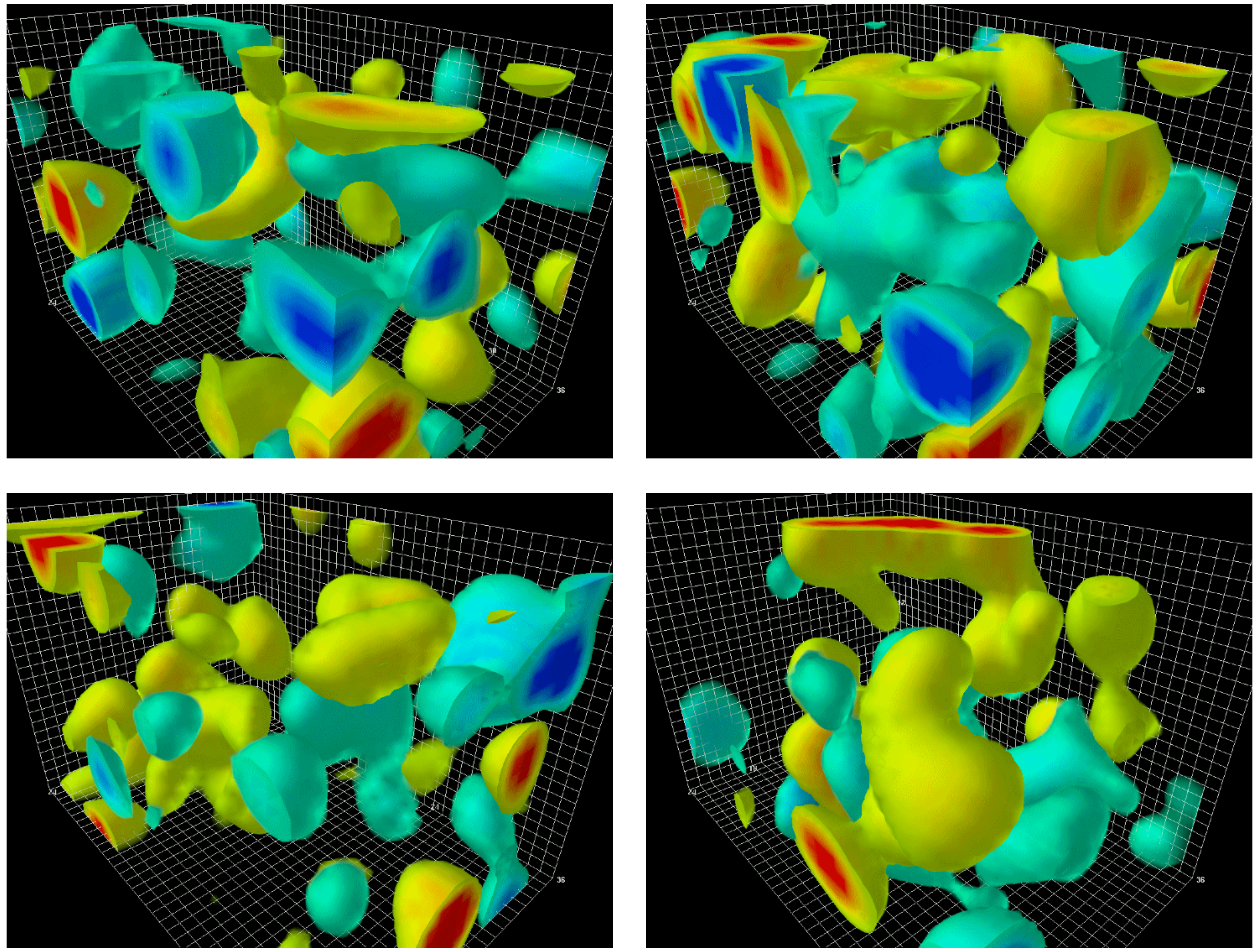



Image credit: Berkowitz, Leinweber

# Lattice simulations

**High-dimensional path integral** over degrees of freedom assigned to points and edges of a lattice

- Boltzmann weight $e^{-S(\phi)}$ encodes distribution over "typical" configurations



Partition function

$$Z \equiv \left[\prod_x \int_{-\infty}^{\infty} d\phi(x)\right] e^{-S(\phi)}$$

Thermal expt. value of operator $\mathcal{O}$

$$\langle \mathcal{O} \rangle = \left[\prod_x \int_{-\infty}^{\infty} d\phi(x)\right] \mathcal{O}(\phi)\, e^{-S(\phi)}/Z$$

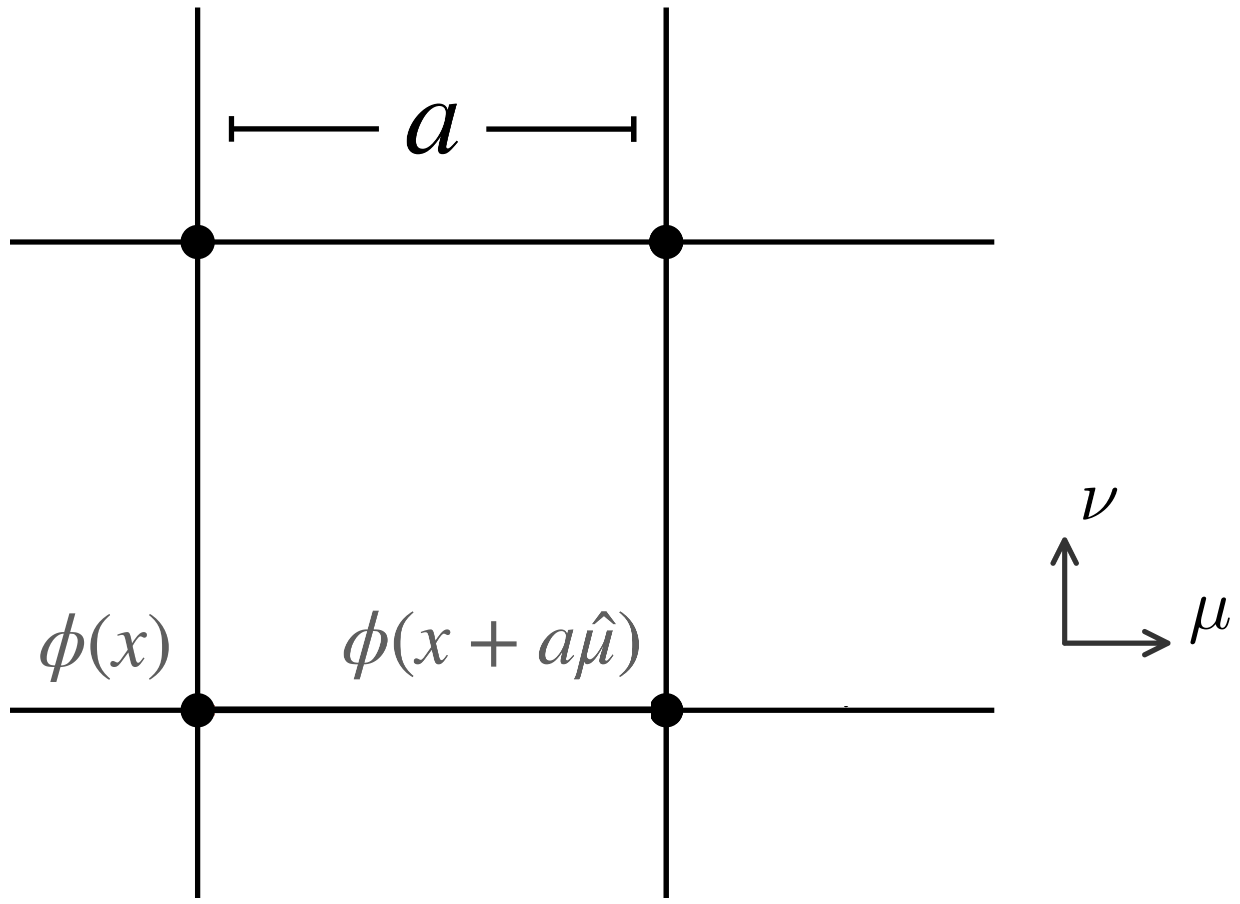
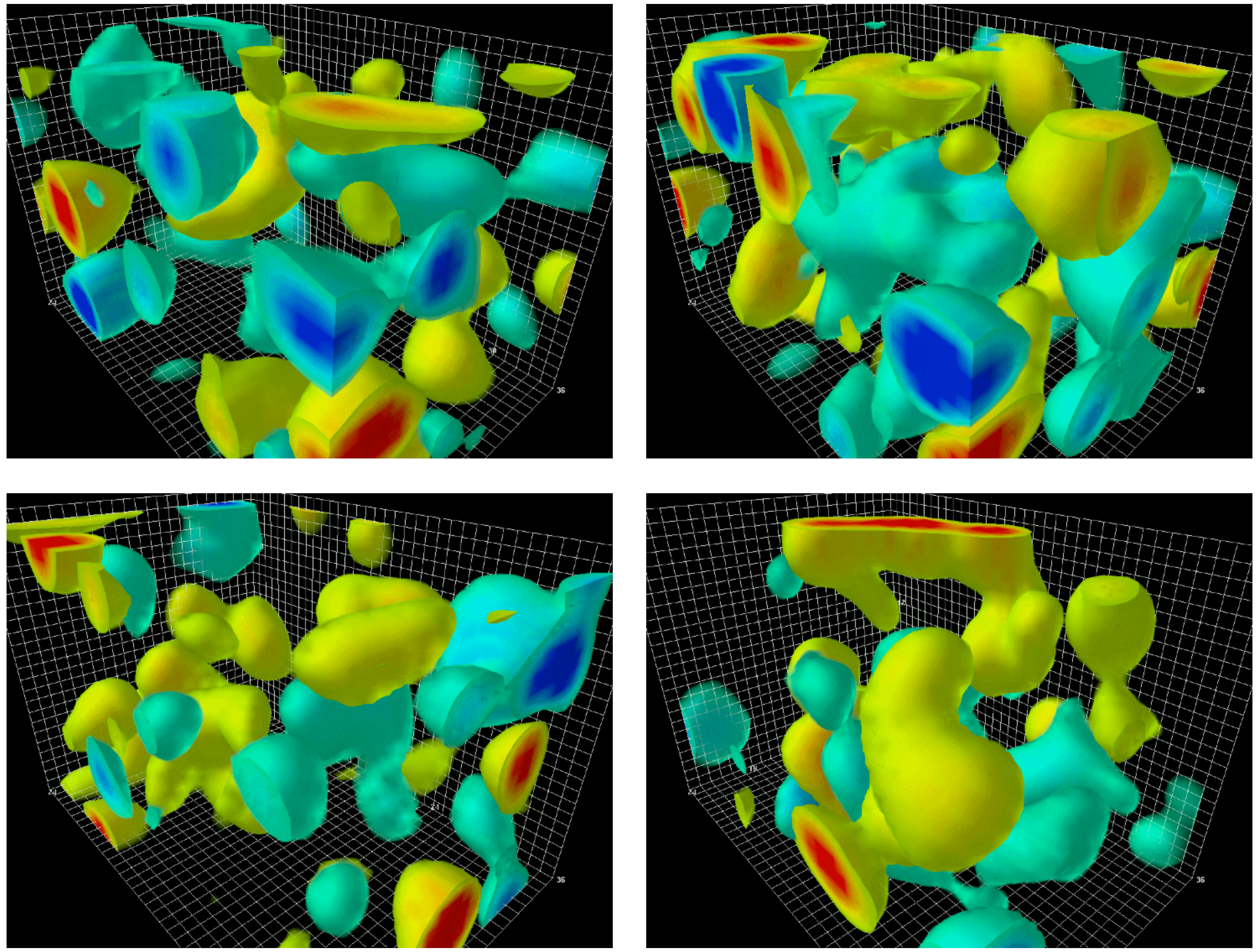



Image credit: Leinweber, Pixar

# Lattice simulations

**High-dimensional path integral** over degrees of freedom assigned to points and edges of a lattice

- Boltzmann weight $e^{-S(\phi)}$ encodes distribution over "typical" configurations



Partition function

$$Z \equiv \left[\prod_{x,\mu} \int dU_\mu(x)\right] e^{-S(U)}$$

Thermal expt. value of operator $\mathcal{O}$

$$\langle \mathcal{O} \rangle = \left[\prod_{x,\mu} \int dU_\mu(x)\right] \mathcal{O}(U)\, e^{-S(U)}/Z$$

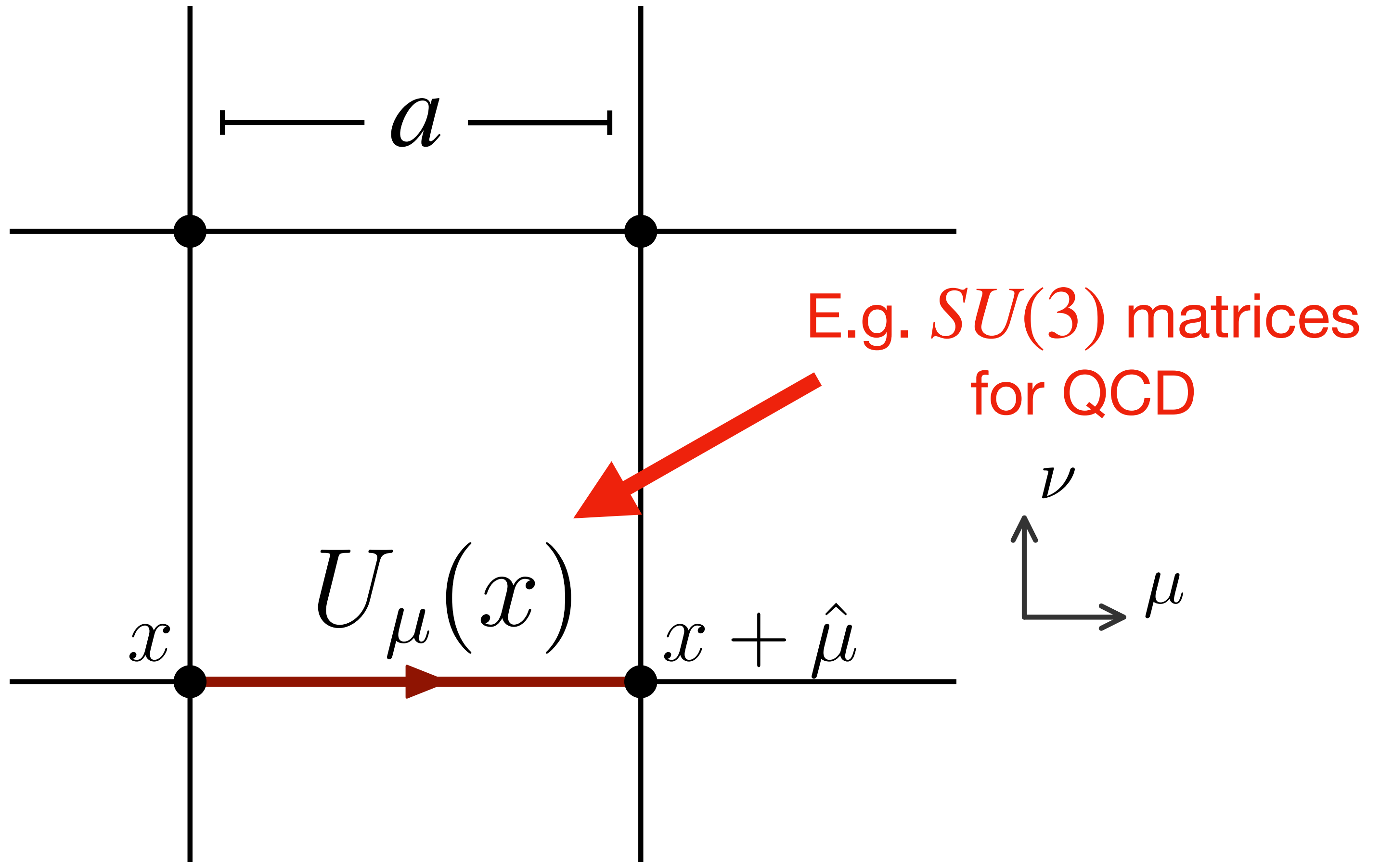
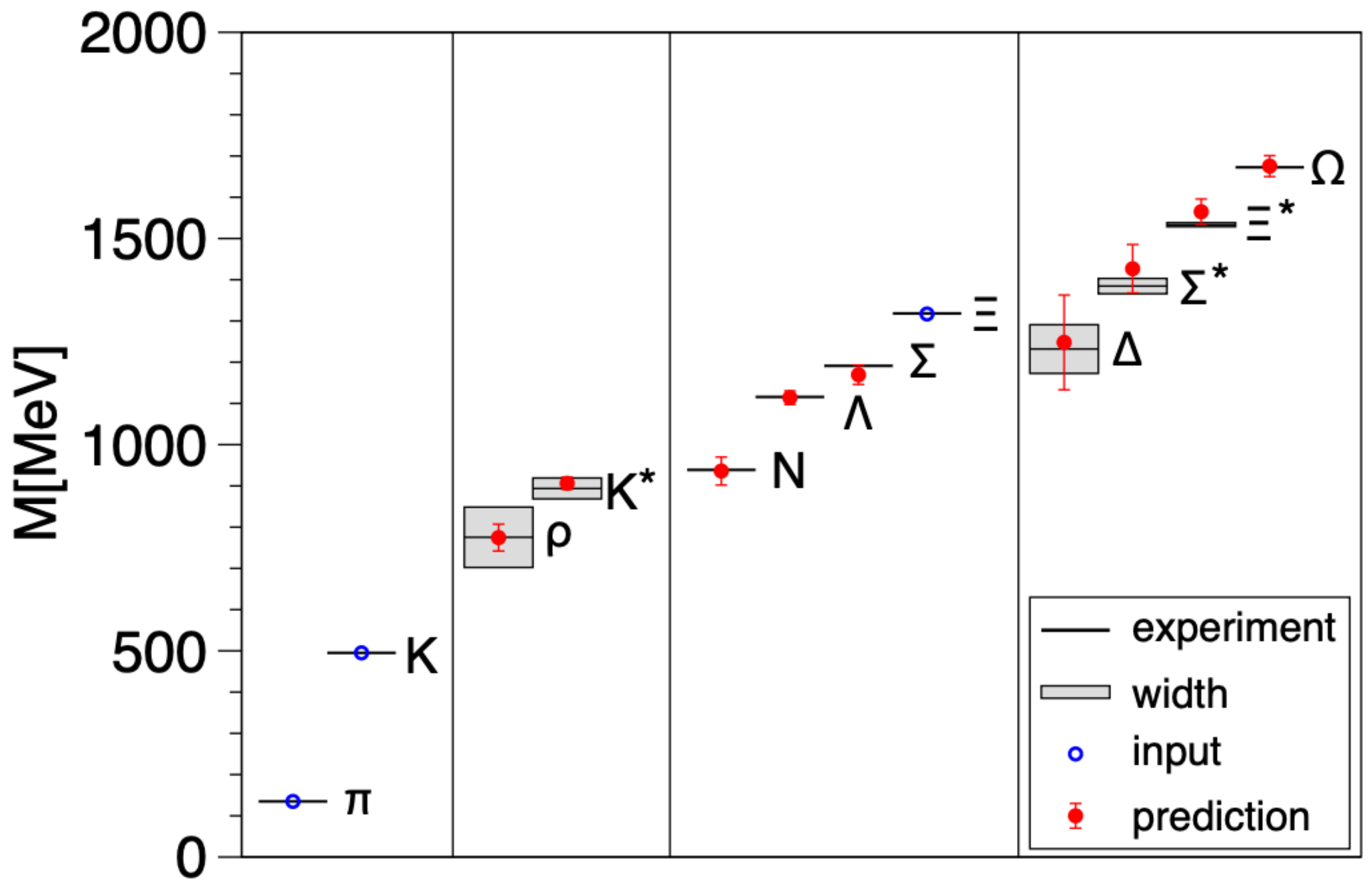


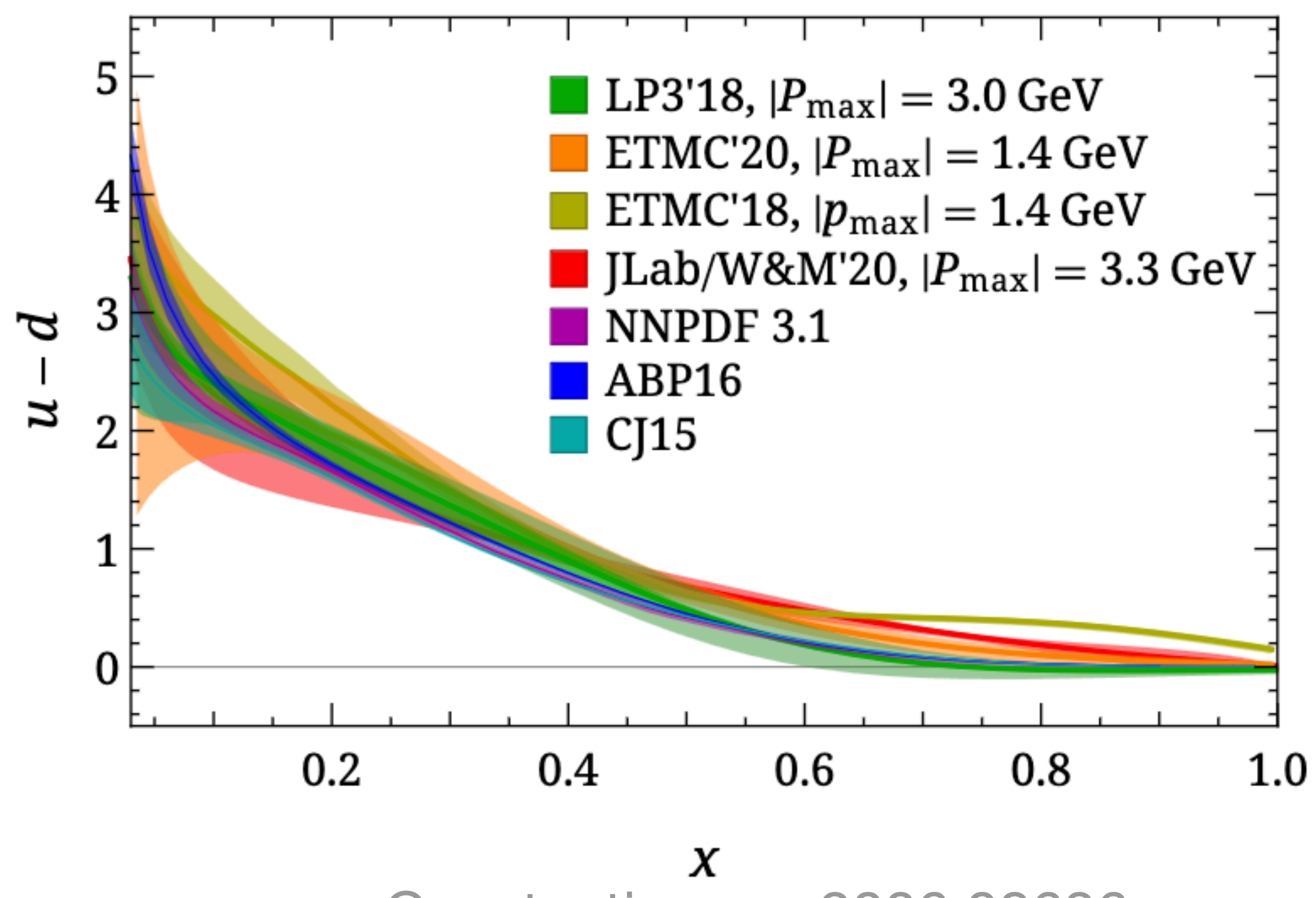

Image credit: Leinweber, Pixar

# Lattice QCD

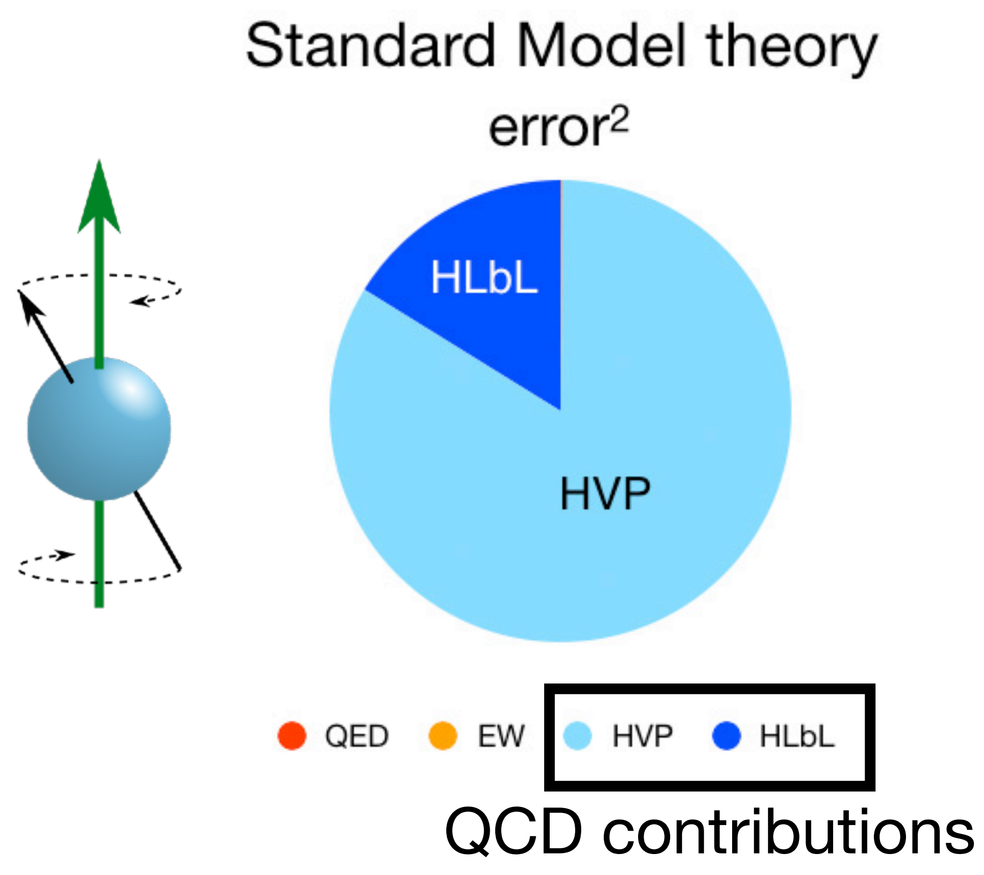
- Hadronic spectrum / structure
  - Heavy resonances
  - PDFs and their generalizations
  - Form factors
- QCD phase diagram
  - Critical point
  - Equation of state
- New physics searches
  - Muon g-2
  - Heavy meson decays
- …




Fodor & Hoelbling RMP84 (2012) 449




Constantinou+ 2006.08636




Muon g-2 Press release (2023)

# Why machine learning?

# The big challenge

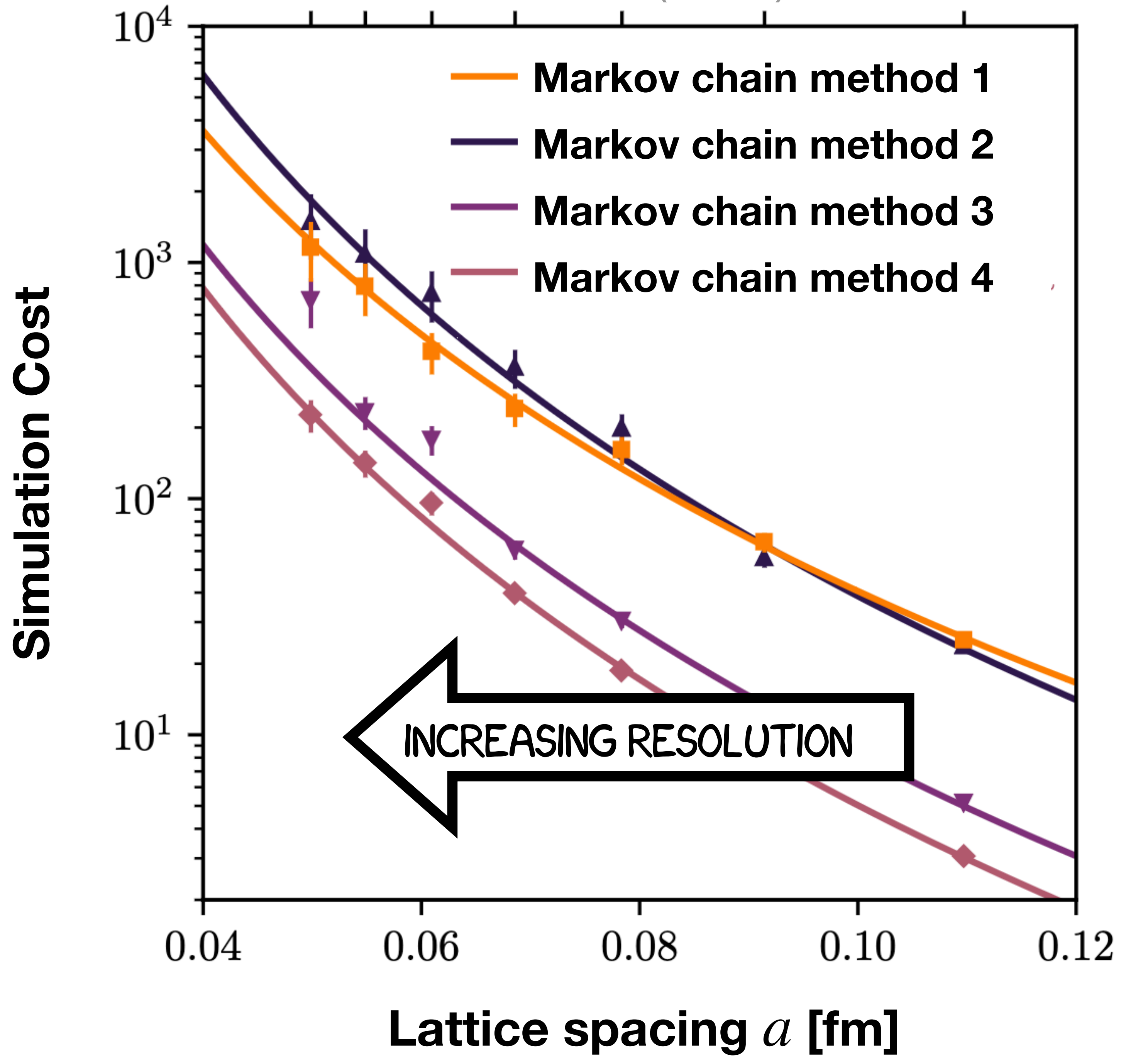
State-of-the-art LGT calculations require **enormous computational effort…**

- $\gtrsim 10^9$ degrees of freedom
- "Critical slowing down" as $a \to 0$
- Costly matrix inversion for propagators $\langle \psi \bar{\psi} \rangle$ (especially as $m_q \to 0$)

… so physics results have **limited precision.**

- Statistical uncertainties
- Systematic uncertainties ($a \to 0,\ m_\pi^{\text{latt}} \to m_\pi,\ V \to \infty$)

Eichhorn, et al. (2023) 2307.04742



# Why machine learning?

Lattice calculations have useful features

- Problem involving **lots** of well-structured data
- Analytic information available (e.g. action)
- Freedom of choice in many aspects

Can now apply ML methods to lattice

- Generative models with exactness now exist
- Industry hardened, scalable ML frameworks

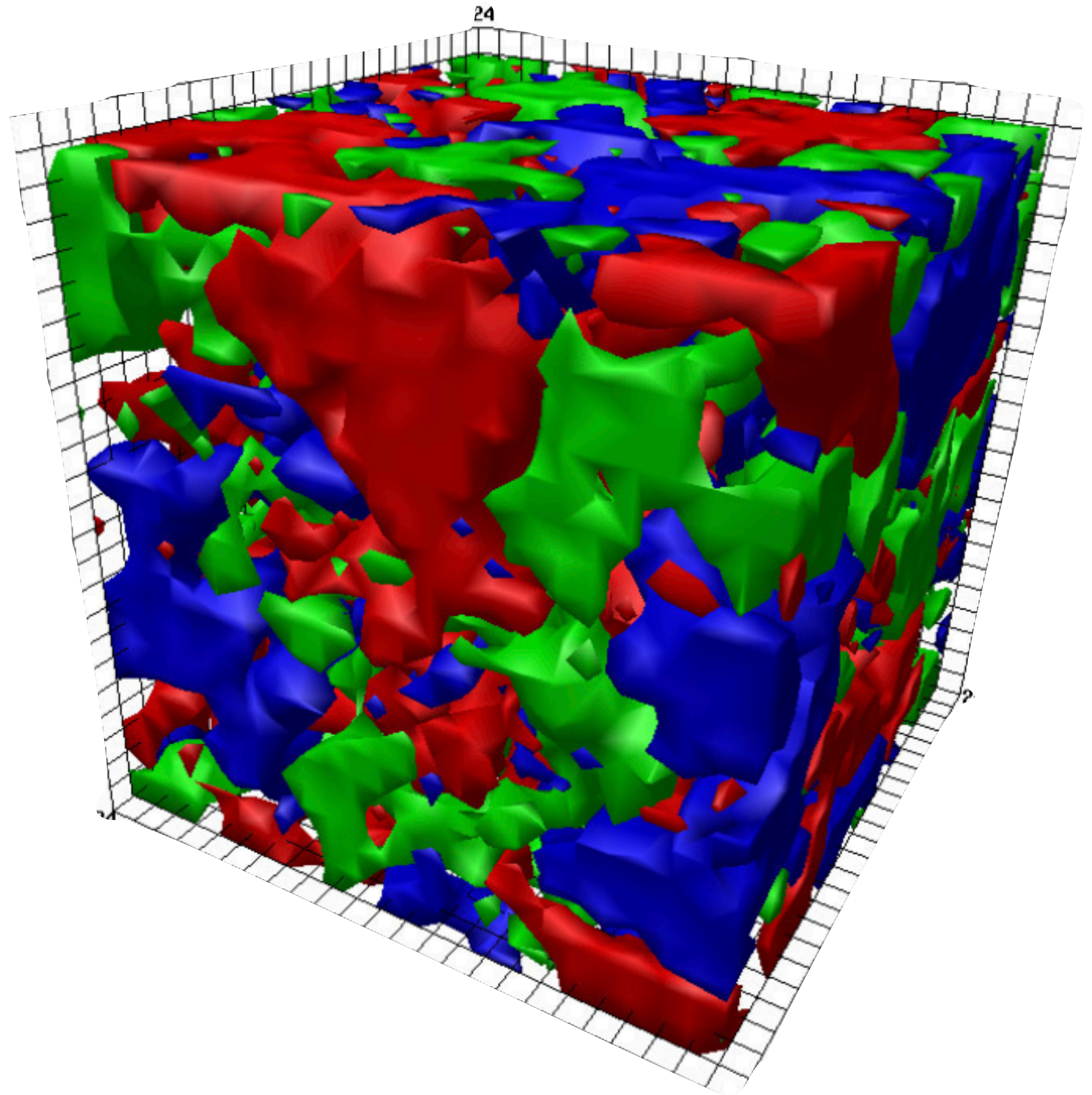


Image credit: Leinweber

# Personal perspective

Focus on methods that avoid introducing systematic bias

> Model quality only determines efficiency

Take a broad perspective on machine learning

> Not just a black box > Become ML researchers

# Some applications of ML

**Two major components** to a lattice calculation.
Ongoing efforts to apply ML to both of these.

1. Ensemble generation
2. Observable measurements & analysis

# Some applications of ML

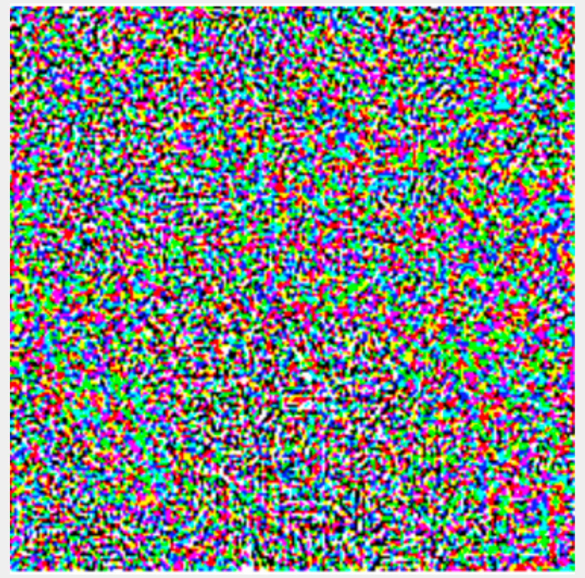
**Two major components** to a lattice calculation.
Ongoing efforts to apply ML to both of these.

1. Ensemble generation
2. Observable measurements & analysis

… is analogous to image generation

few typical "configurations"

many atypical "configurations"



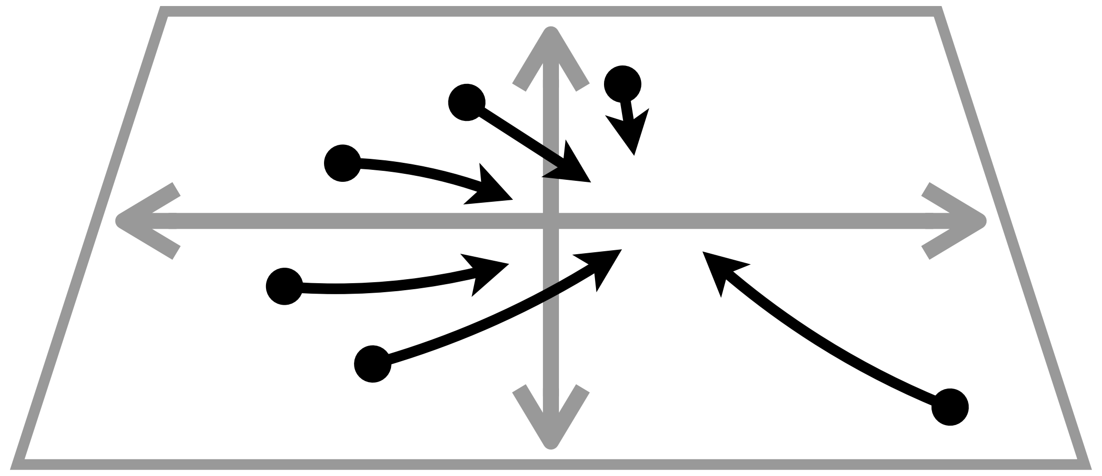
Karras, Lane, Aila / NVIDIA 1812.04948

generated images!

# Some applications of ML

**Two major components** to a lattice calculation.
Ongoing efforts to apply ML to both of these.

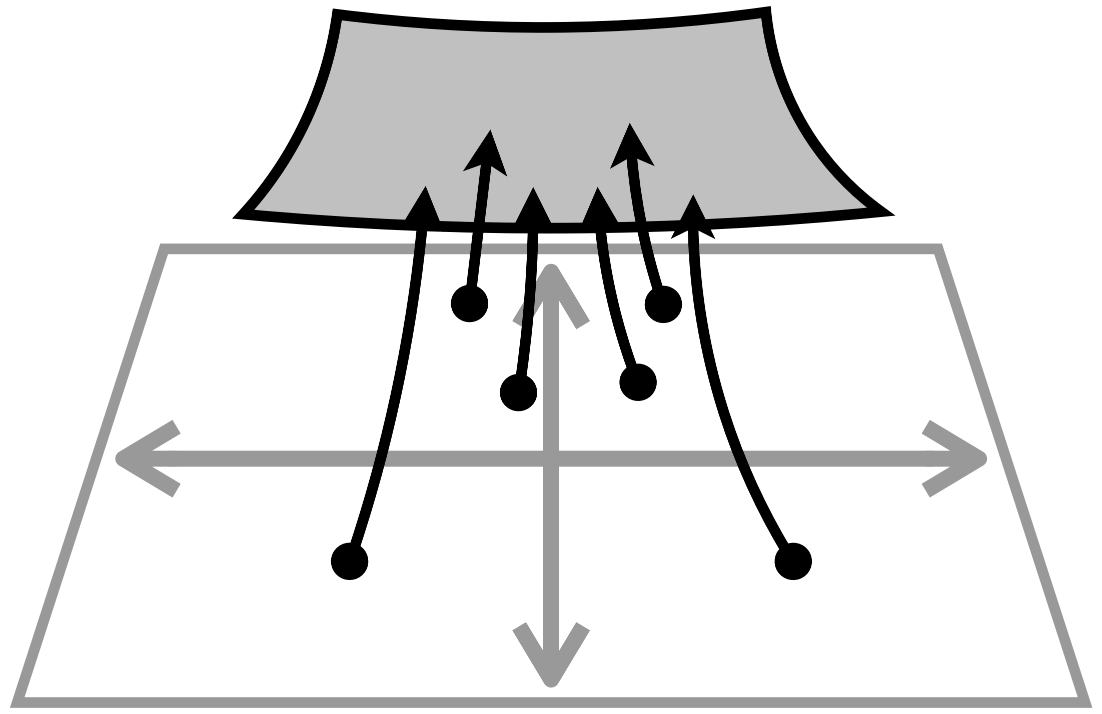
1. Ensemble generation



## Normalizing flow models

- PRD100 (2019) 034515, 2101.08176, 2107.00734
- PRL125 (2020) 121601, ICML (2020) 2002.02428, PRD103 (2021) 074504, 2305.02402
- PRD104 (2021) 114507, PRD106 (2022) 014514, PRD106 (2022) 074506, PoSLATTICE (2022) 036
- 2211.07541, 2401.10874, 2404.10819, 2404.11674, 2502.00263

2. Observable measurements & analysis



## Learned contour deformations

- PRD98 (2018) 074511, PoS LATTICE2018 176
- PRD102 (2020) 014514, PRD103 (2021) 094517
- 2309.00600, NeurIPS ML4PS (2023), 2410.03602

# Disclaimer

*I will present only a narrow view of one approach in this wide field.*

- View of the overarching goals of this program
- Some transferrable lessons

*I will not cover several related works:*

- Learned control variates for observables
- Learned preconditioners for Dirac matrix inversion
- Learned spectral function reconstruction
- …

**See Boyda, et al. 2202.05838 for a semi-recent review**

# Normalizing flow models

# Markov chains

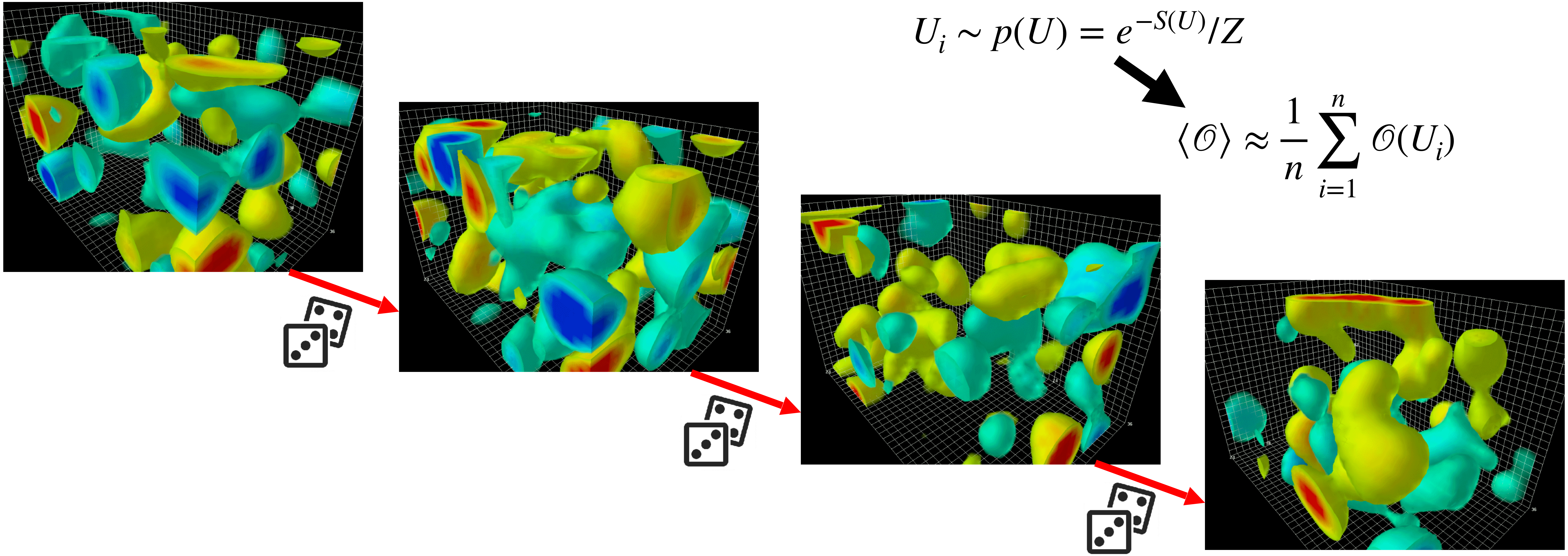
$$\langle \mathcal{O} \rangle = \left[ \prod_{x,\mu} \int dU_\mu(x) \right] \mathcal{O}(U)\, e^{-S(U)}/Z$$

Usually approximate the path integral using **Markov chain Monte Carlo**

Positive integrand allows interpreting path integral weights as a probability measure:

$$U_i \sim p(U) = e^{-S(U)}/Z$$

$$\langle \mathcal{O} \rangle \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{O}(U_i)$$



# Critical slowing down

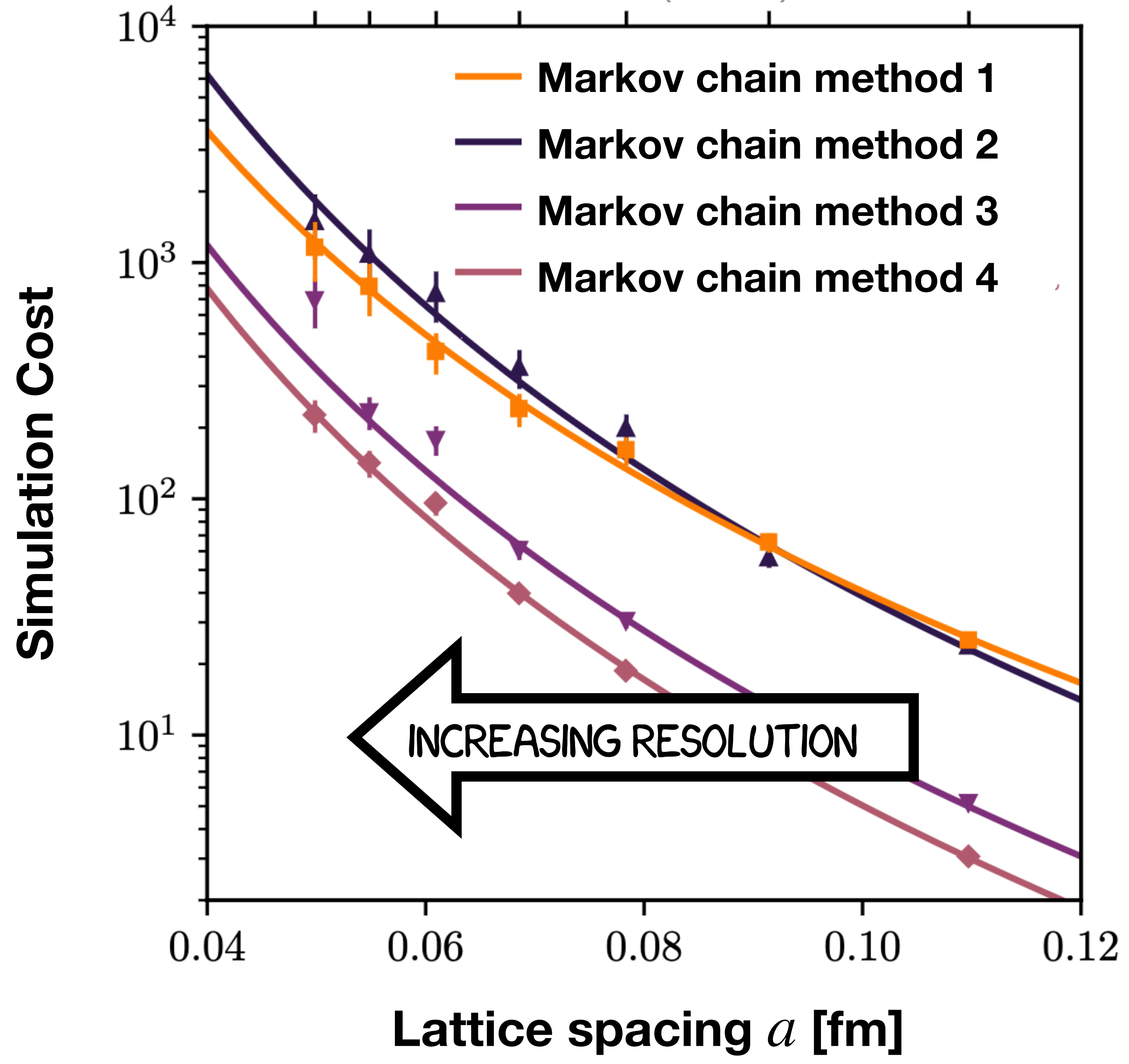
Local/diffusive Markov chains inefficient as $a \to 0$

- Correlation length grows, information transfer is local
- Rare to update entire field coherently

**Critical slowing down:** autocorrelations diverge due to local information transfer

**Topological freezing:** Markov chain gets "stuck" in topological sectors

Eichhorn, et al. (2023) 2307.04742



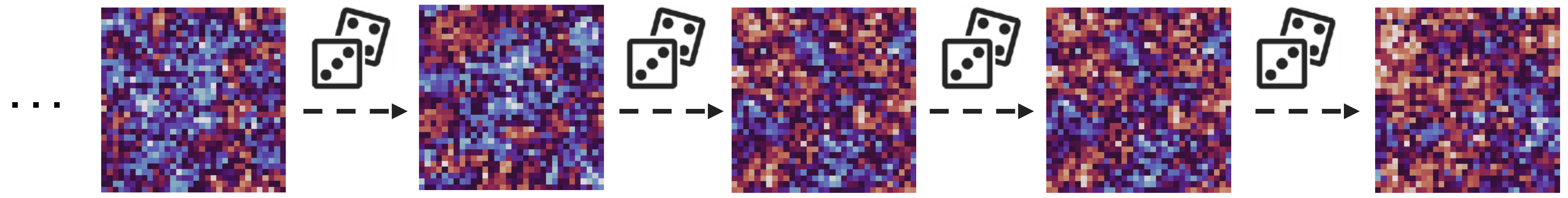



CSD also affects a number of other models:

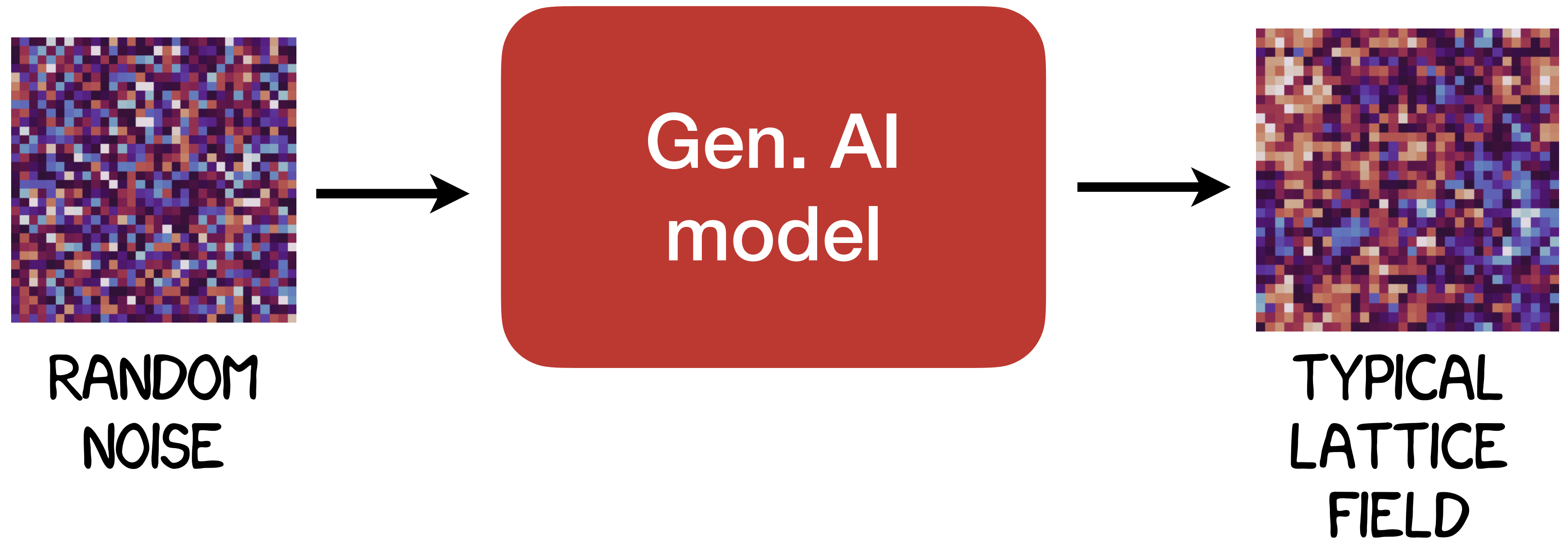
- $CP^{N-1}$ Flynn, et al. **1504.06292**
- O(N) Frick, et al. **PRL63 (1989) 2613**
- $\phi^4$ Vierhaus; Thesis, **doi:10.18452/14138**
- ...

# Can we use **Generative AI** to solve critical slowing down?

Replace or augment **Markov Chain Monte Carlo…**

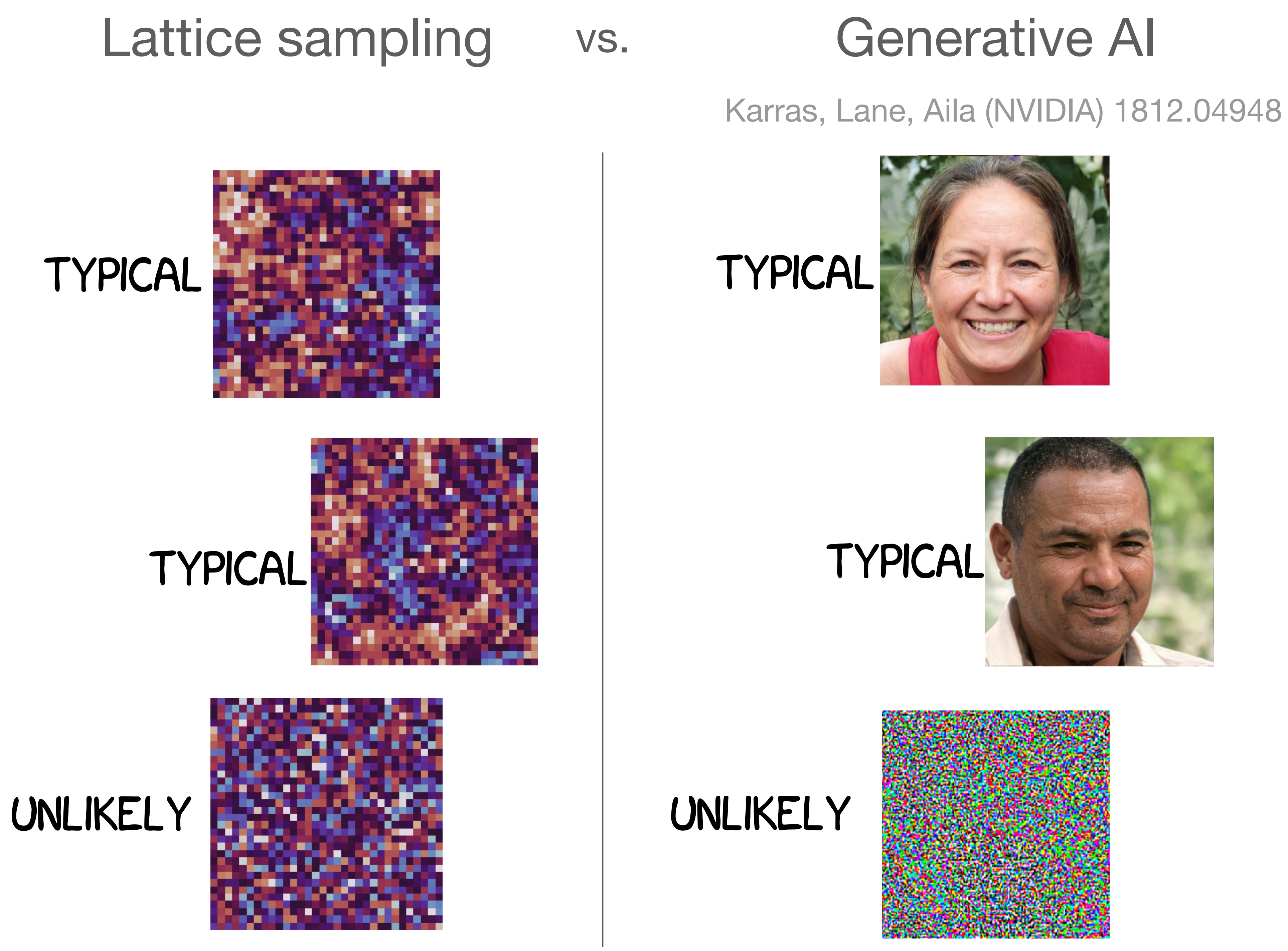



… with generated samples?



# Generative AI to solve CSD?

*Generative models can directly produce "typical" samples*




Unique features:

- ⚠️ Demand **unbiased results**
- ⚠️ Not much **existing training data**
  … $O(1000)$ samples at finest resolutions
  … $O(10{,}000{,}000{,}000)$ components/sample
- ✓ Know **target probability density**
- ✓ Know **physical symmetries**

Albergo, GK, Shanahan PRD100 (2019) 034515

Massachusetts Institute of Technology

The NSF Institute for Artificial Intelligence and Fundamental Interactions


Phiala Shanahan


Denis Boyda


Fernando Romero-López


Julian Urban


Ryan Abbott

HARVARD UNIVERSITY

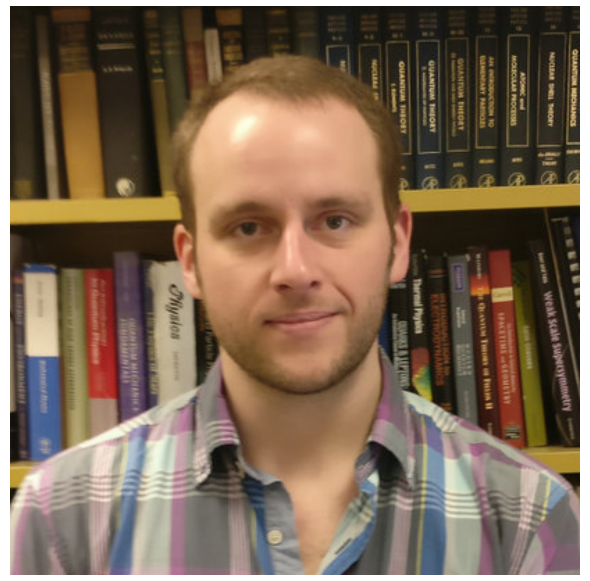
The NSF Institute for Artificial Intelligence and Fundamental Interactions


Michael Albergo

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON


Kyle Cranmer

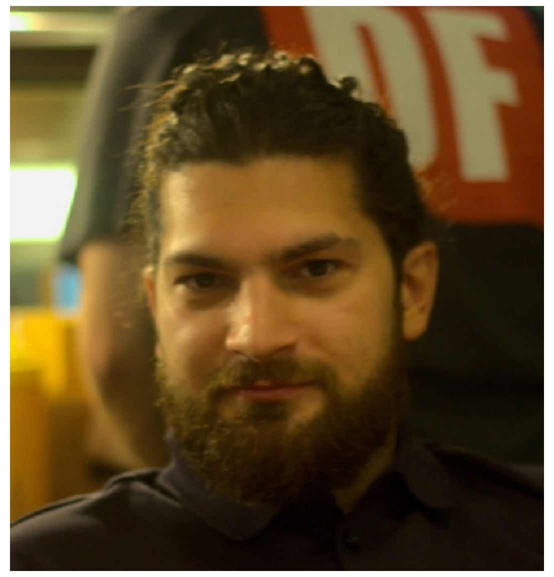
Fermilab


Dan Hackett

DeepMind


Sébastien Racanière


Danilo Rezende


Aleksander Botev

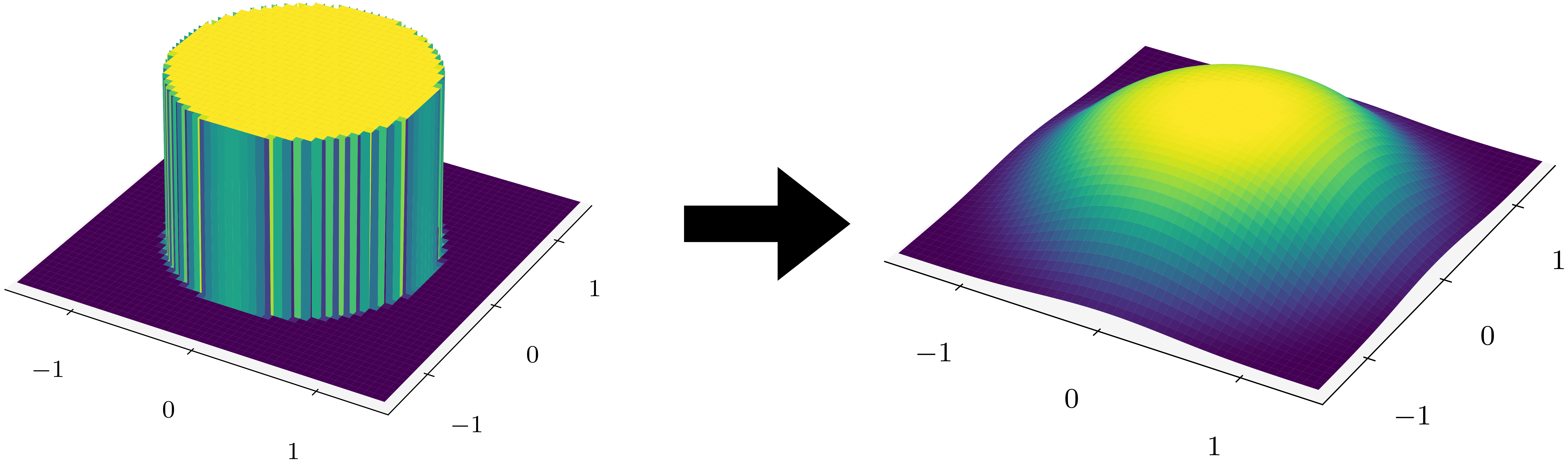

Alexander Matthews


Ali Razavi

# Direct sampling using flows

Box-Muller transform (Marsaglia polar form)

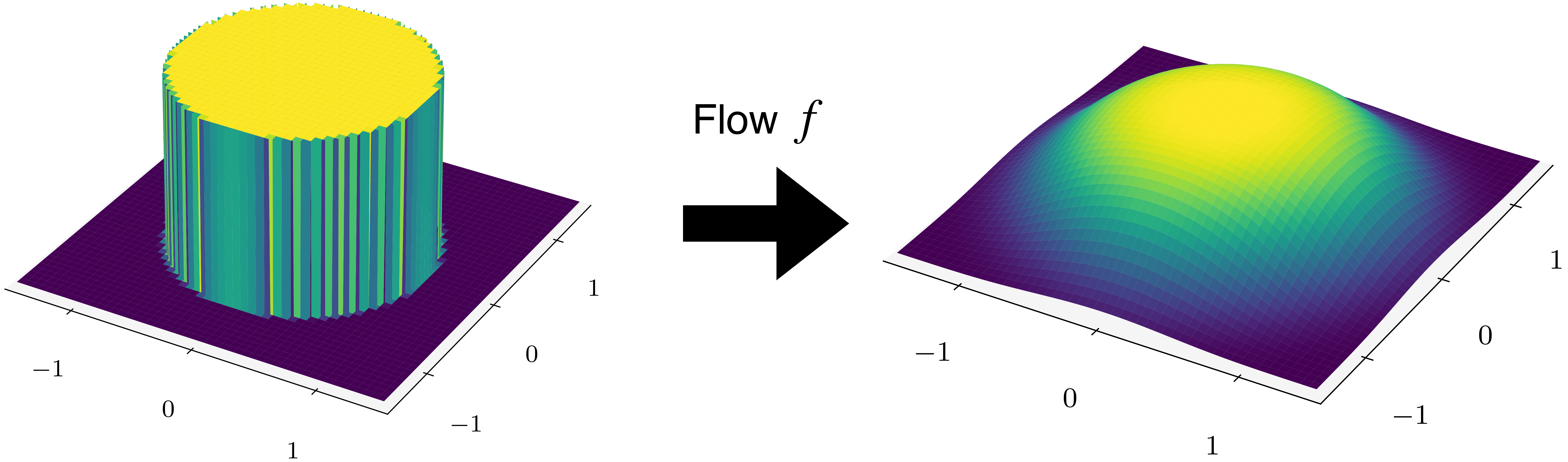
$$x' = \frac{x}{r}\sqrt{-2\ln r^2} \qquad y' = \frac{y}{r}\sqrt{-2\ln r^2}$$



# Direct sampling using flows

Box-Muller transform (Marsaglia polar form)

$$x' = \frac{x}{r}\sqrt{-2\ln r^2} \qquad y' = \frac{y}{r}\sqrt{-2\ln r^2}$$



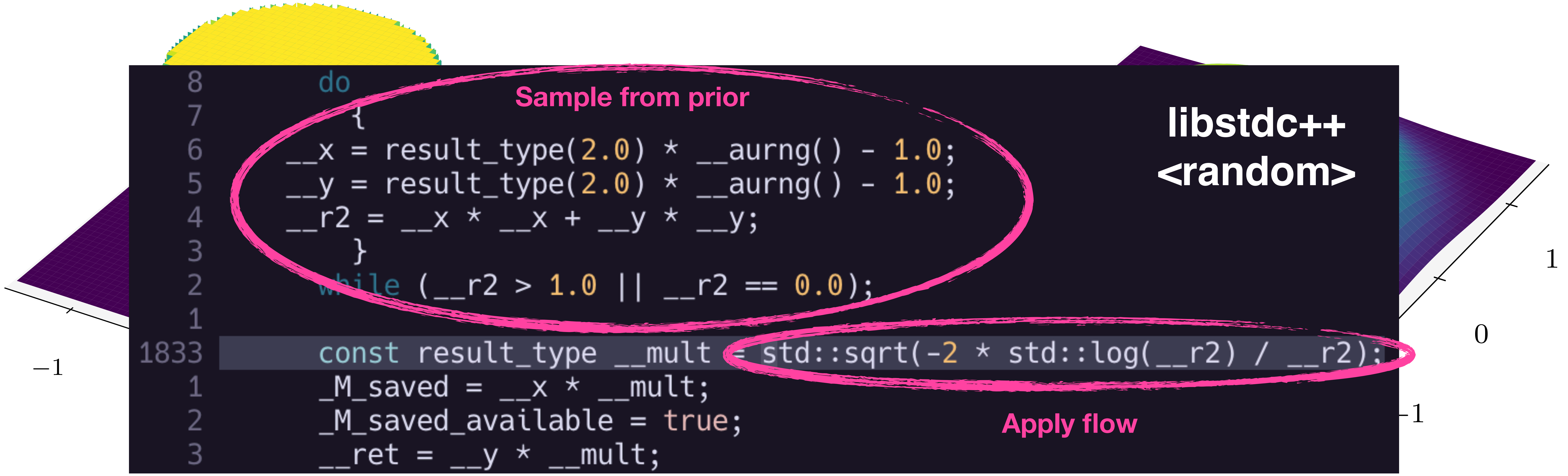

(Simple) Prior density:
$r(x, y)$

(More complex) Output density:
$q(x', y') = r(x, y)\,|\det J|^{-1}$

# Direct sampling using flows

Box-Muller transform (Marsaglia polar form)

$$x' = \frac{x}{r}\sqrt{-2\ln r^2} \qquad y' = \frac{y}{r}\sqrt{-2\ln r^2}$$




(Simple) Prior density:
$r(x, y)$

(More complex) Output density:
$q(x', y') = r(x, y)\,|\det J|^{-1}$

# Normalizing flow models

Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145

- Sample from "easy" prior density $r(V)$
- Apply parametrized diffeomorphism $f$ (the "flow")

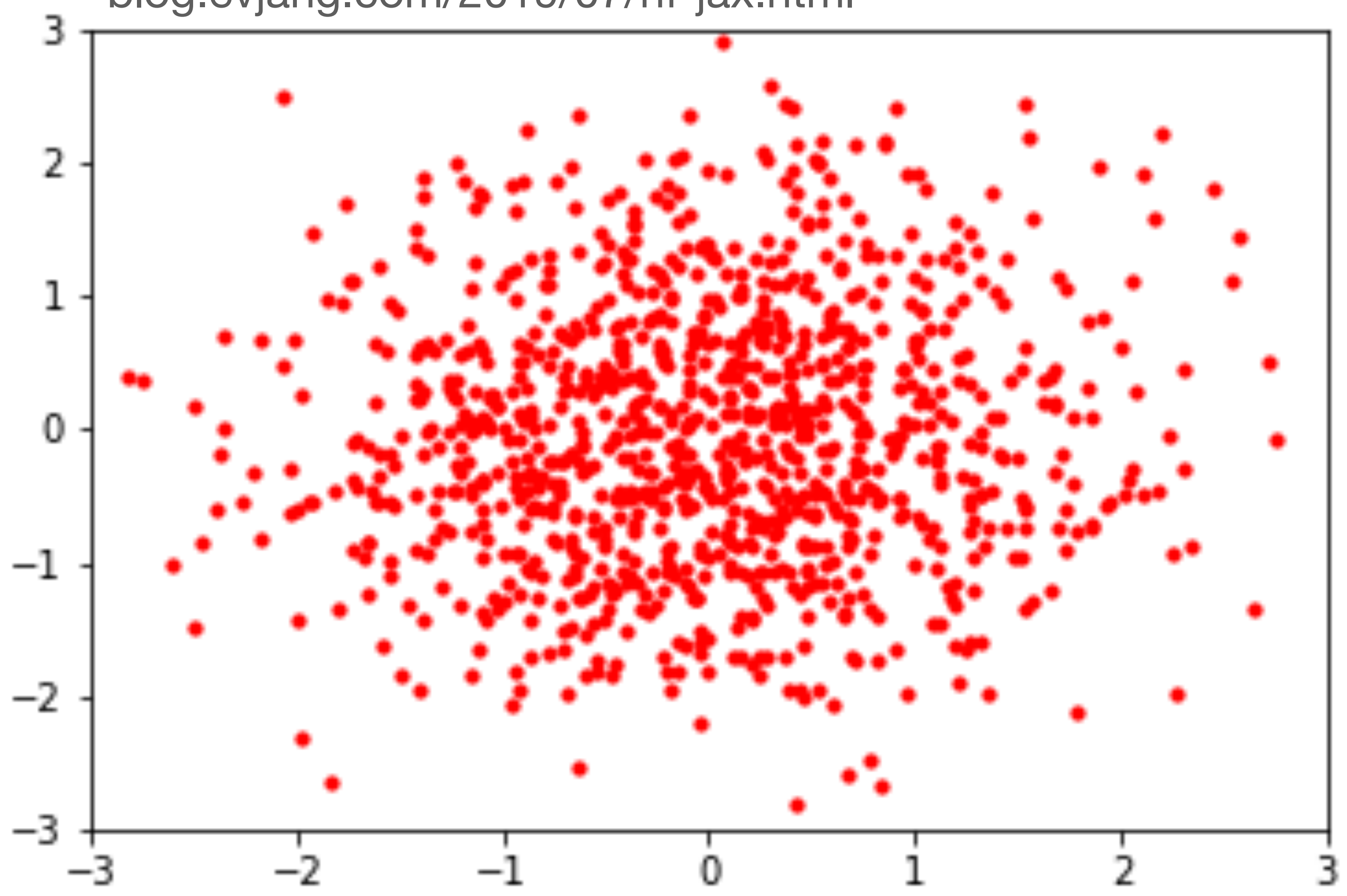
$$U = f(V)$$

- Output samples follow computable "model density"

$$q(U) = r(V) \det \left| \frac{\partial f(V)}{\partial V} \right|^{-1}$$

- Flow $f$ can be **trained** to match target density!

blog.evjang.com/2019/07/nf-jax.html




Simple prior distribution $r(V)$ → Complex model distribution $q(U)$

# Normalizing flow models

Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145

- Sample from “easy” prior density $r(V)$
- Apply parametrized diffeomorphism $f$ (the “flow”)

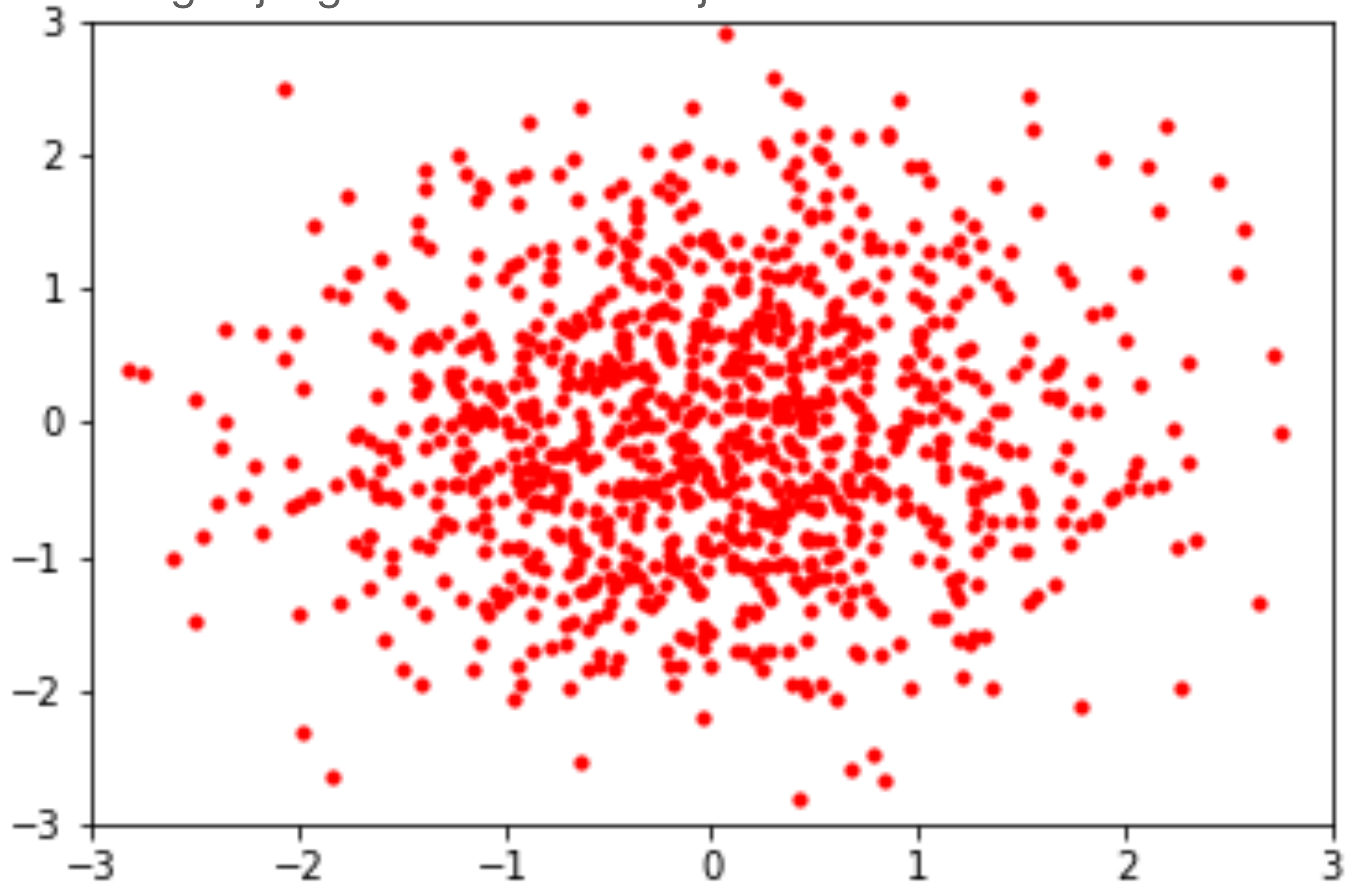
$$U = f(V)$$

- Output samples follow computable “model density”

$$q(U) = r(V) \det \left| \frac{\partial f(V)}{\partial V} \right|^{-1}$$

- Flow $f$ can be **trained** to match target density!

blog.evjang.com/2019/07/nf-jax.html




Simple prior distribution $r(V)$ → Complex model distribution $q(U)$

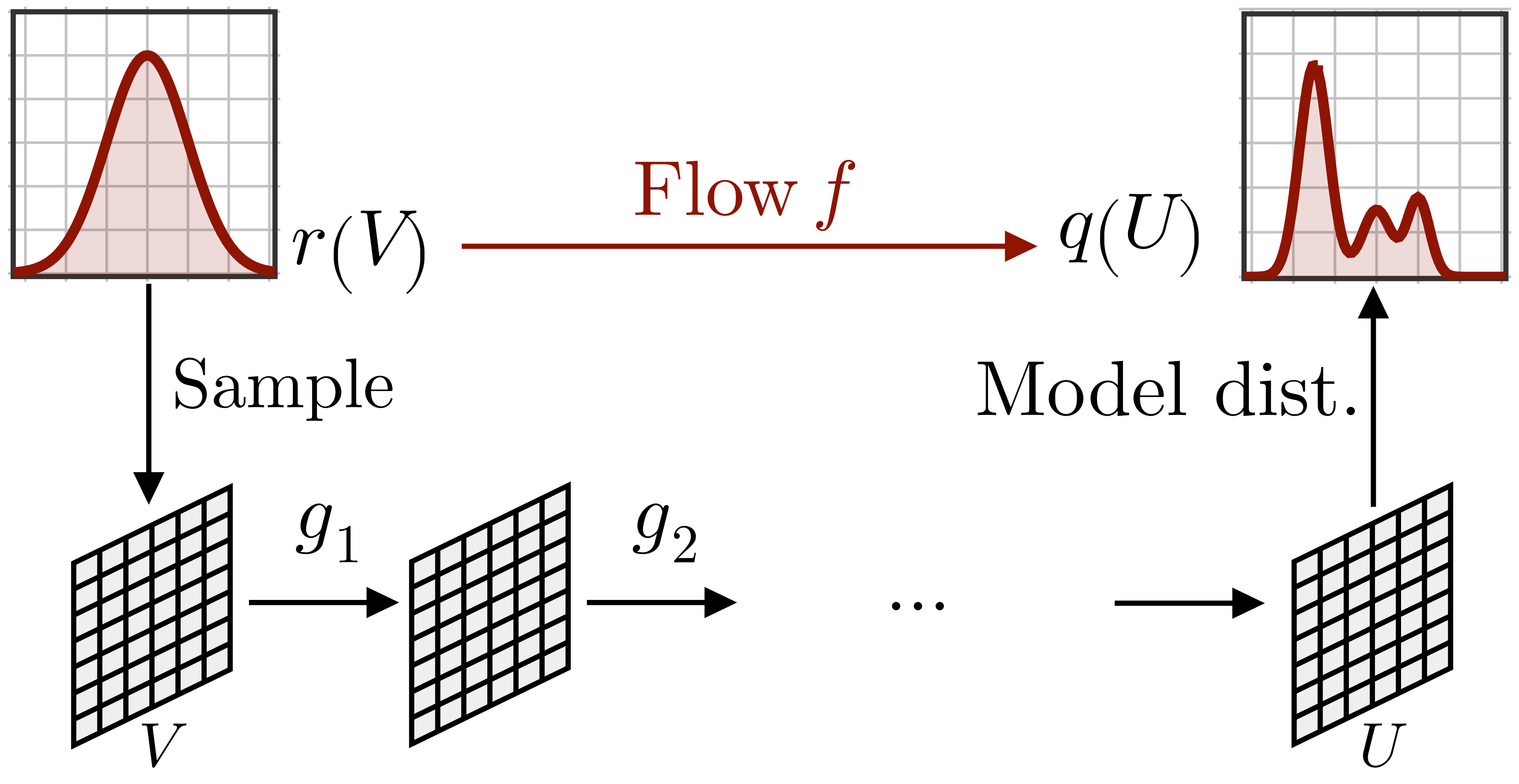
# Defining the flow function

The flow $f$ must be **invertible** and have **tractable Jacobian determinant**

- For LQFT, don't know what $f$ needs to be *a priori*
- Expressive parameterized ansatz + optimization

$$q(U) = r(V) \left| \det_{ij} \frac{\partial [f(V)]_i}{\partial V_j} \right|^{-1}$$

Key to expressivity — **Use composition.**




Each layer is **invertible**, has **tractable Jac.**

Simple individual layers combine to give complex transformations.

# Self-training scheme

**Machine learning jargon**

**Training** = optimization, typically by stochastic gradient descent

**Loss function** $\mathscr{L}$ = target function to be minimized

*Optimization designed for inverted data hierarchy in the lattice problem.* Lesson 1

Kullback & Leibler Ann. Math. Statist. 22 (1951) 79

1. Define **"Reverse" Kullback-Leibler (KL)** divergence between $q(\phi)$ and $p(\phi) = e^{-S(\phi)}/Z$

$$D_{\mathrm{KL}}(q\,||\,p) := \int \mathscr{D}\phi\, q(\phi) \left[\log q(\phi) - \log p(\phi)\right] \geq 0$$

2. Measure using samples $\phi_i$ **from the model**

$$D_{\mathrm{KL}}(q\,||\,p) \approx \frac{1}{M} \sum_{i=1}^{M} \left[\log q(\phi_i) + S(\phi_i)\right]$$

3. Minimize by stochastic gradient descent

Inspired by:

- Self-Learning Monte Carlo (SLMC) [Huang, Wang PRB95 (2017) 035105; Liu, et al. PRB95 (2017) 041101; …]

- Self-play reinforcement learning [Silver, et al. Science 362 (2018), 1140]

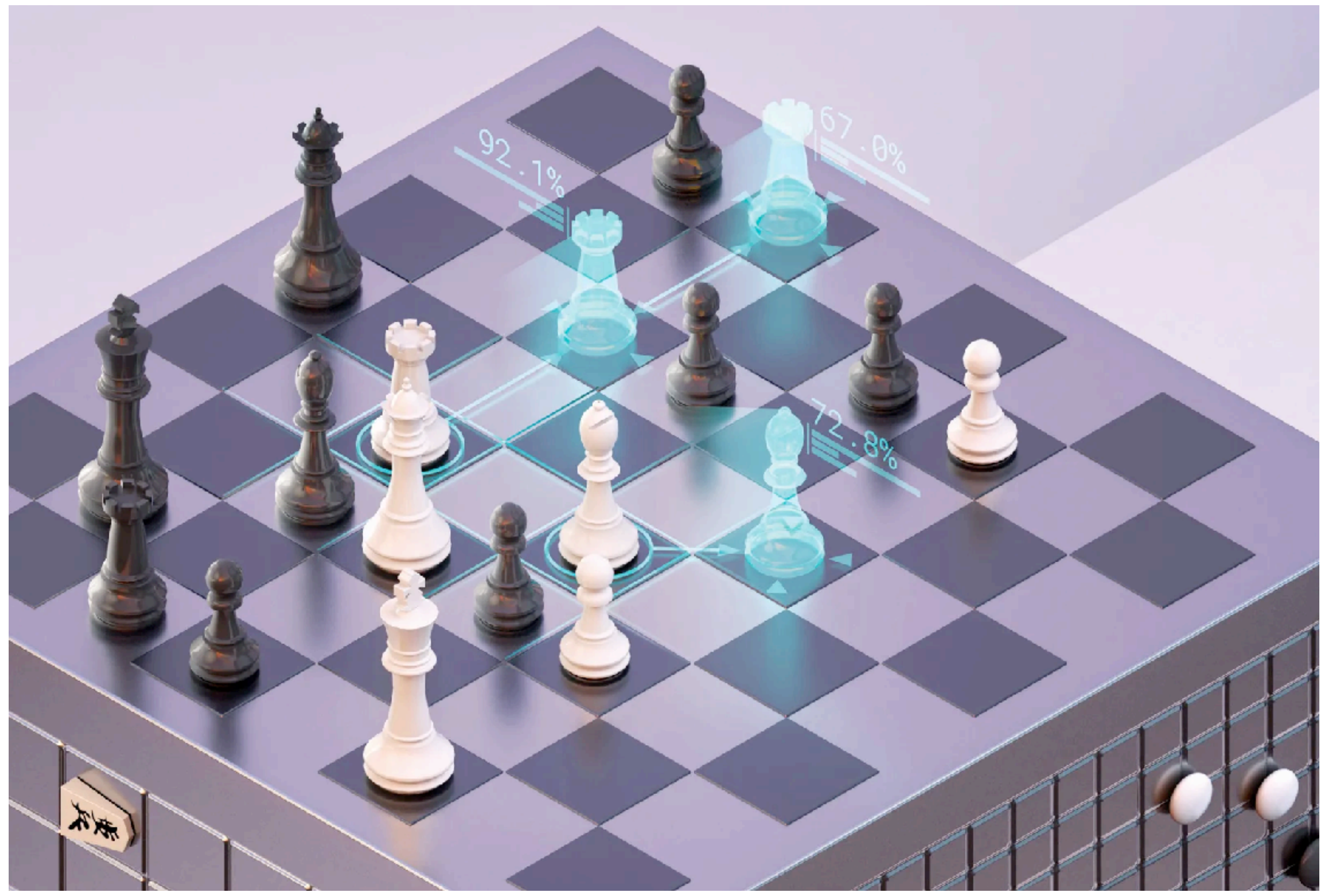
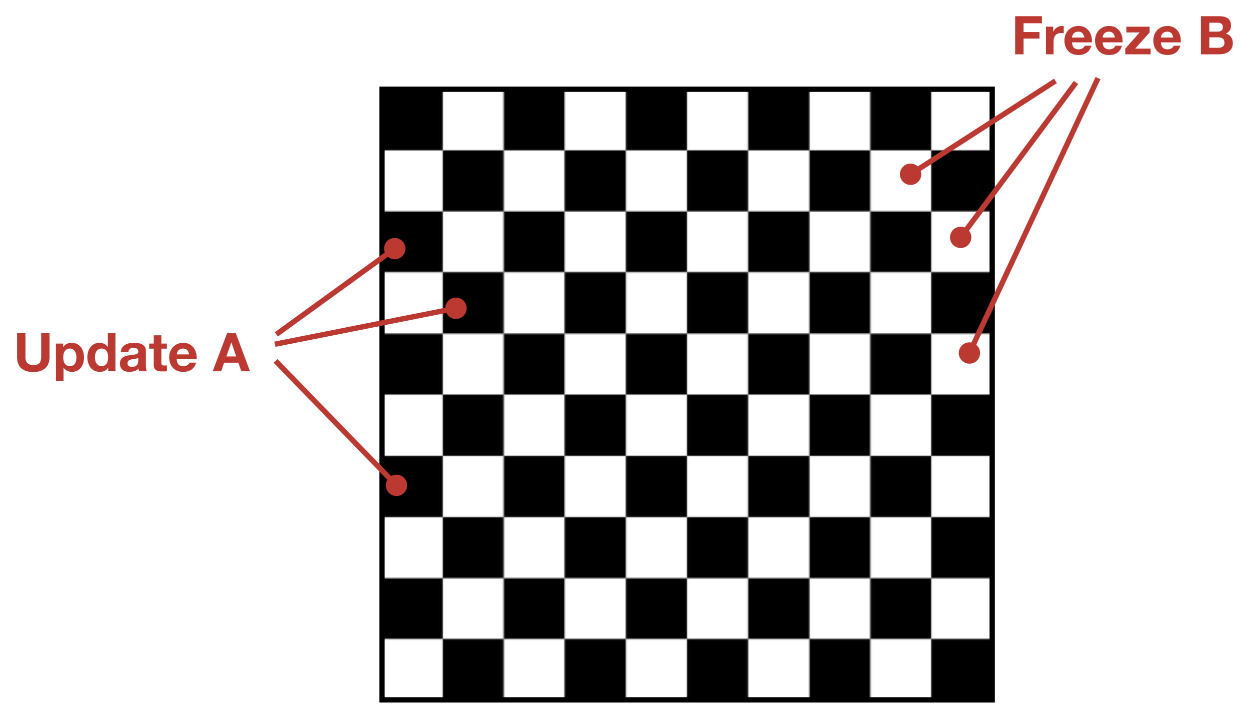


Image credit: DeepMind

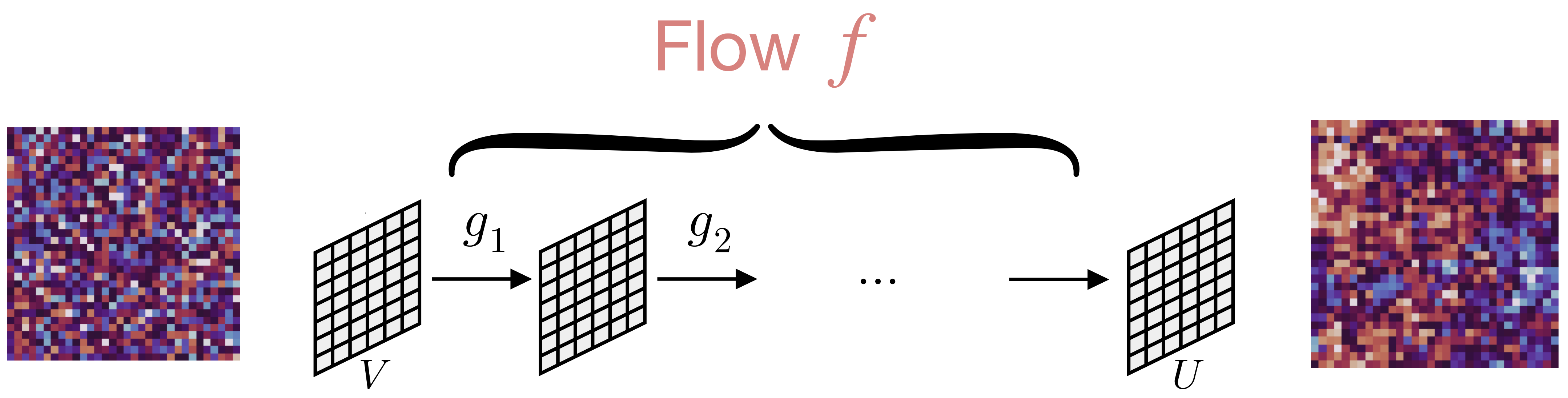
# Flows for scalar $\phi^4$ theory

Scalar field $\phi(x) \in \mathbb{R}$, 1+1D spacetime

$$S[\phi] = \sum_x \partial_\mu \phi(x) \partial^\mu \phi(x) + \frac{M^2}{2}\phi(x)^2 + \lambda \phi(x)^4$$




Checkerboard masking pattern $m$

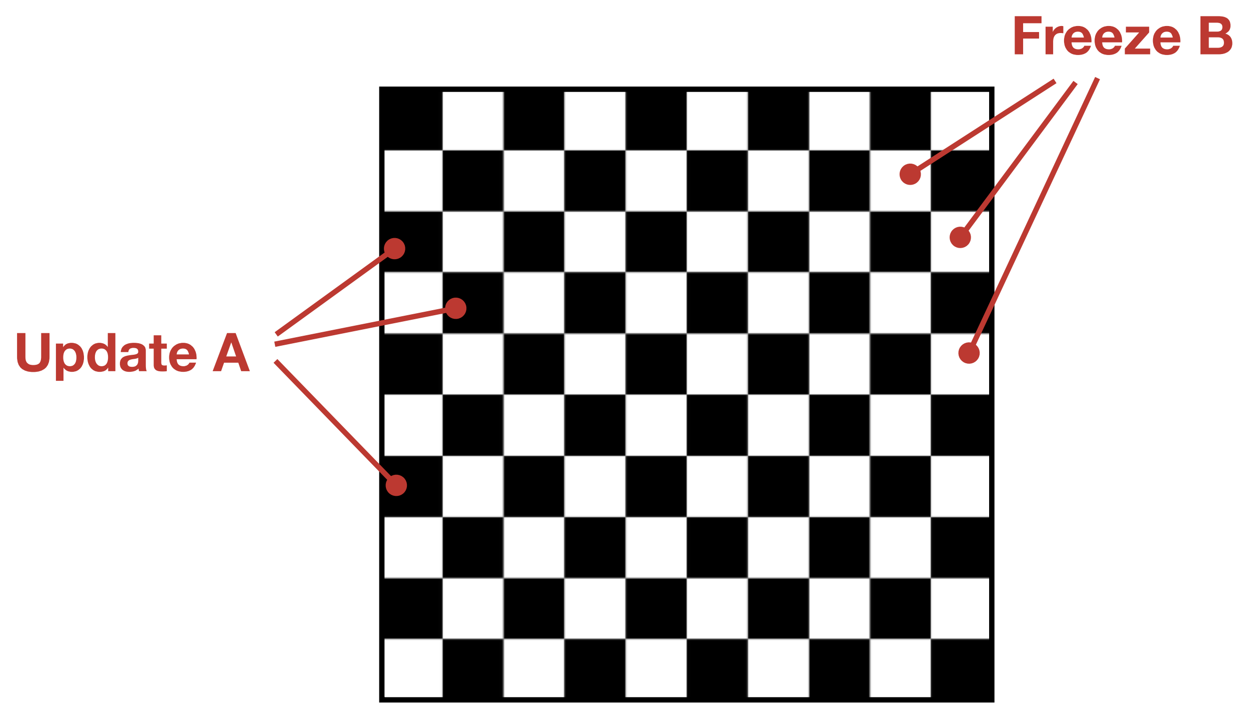



Albergo, GK, Shanahan PRD100 (2019) 034515

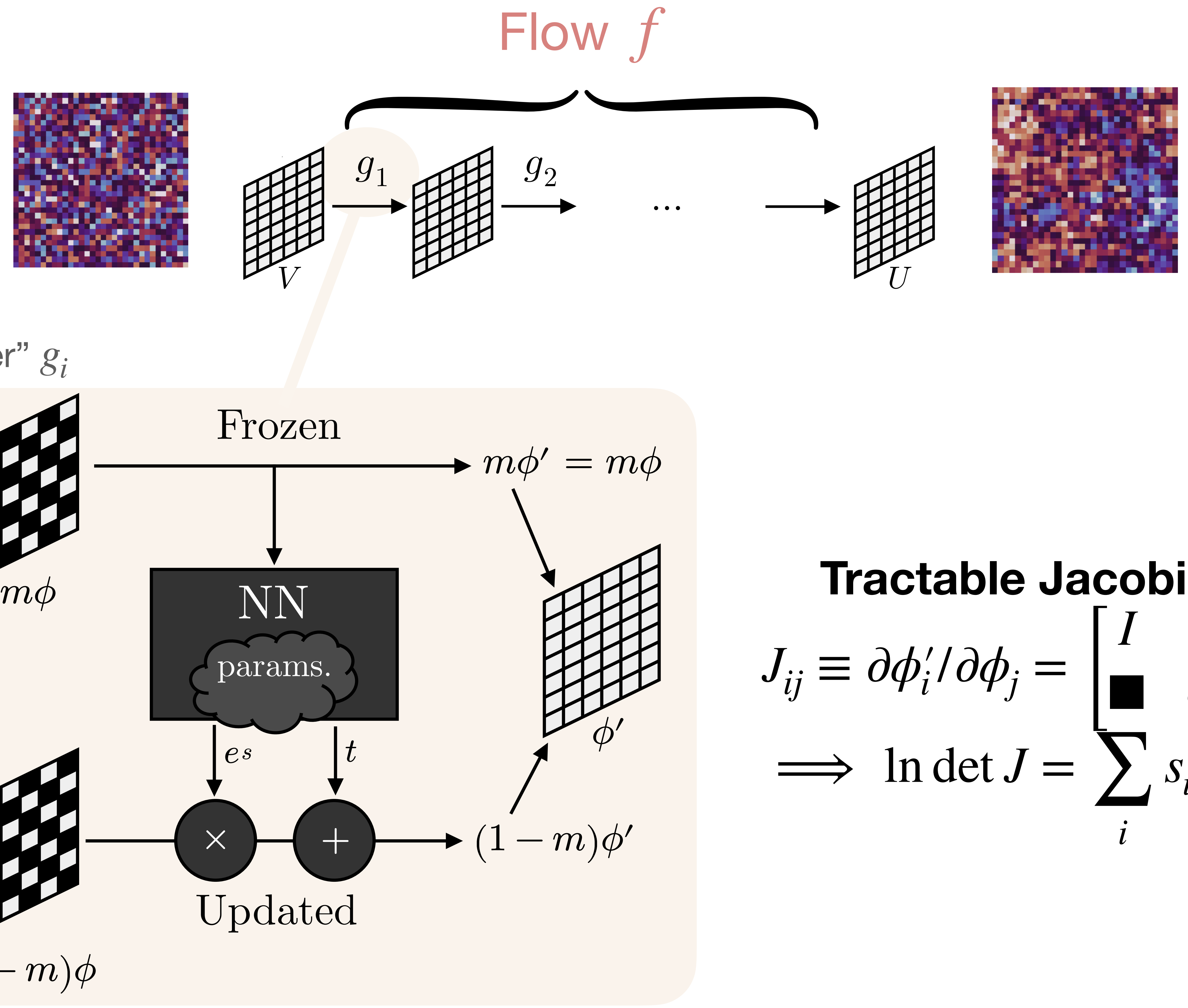
# Flows for scalar $\phi^4$ theory

Scalar field $\phi(x) \in \mathbb{R}$, 1+1D spacetime

$$S[\phi] = \sum_x \partial_\mu \phi(x) \partial^\mu \phi(x) + \frac{M^2}{2} \phi(x)^2 + \lambda \phi(x)^4$$




Checkerboard masking pattern $m$




**Tractable Jacobian**

$$J_{ij} \equiv \partial \phi'_i / \partial \phi_j = \begin{bmatrix} I & \\ \blacksquare & \delta_{ij} e^{s_i} \end{bmatrix}$$

$$\implies \ln \det J = \sum_i s_i$$

Albergo, GK, Shanahan PRD100 (2019) 034515

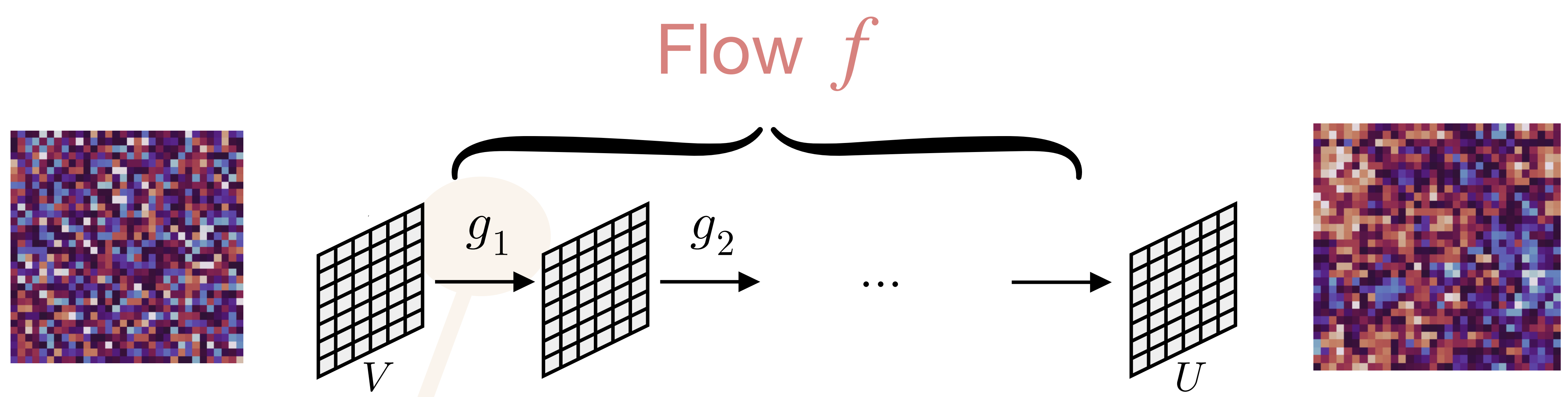
# Flows for scalar $\phi^4$ theory

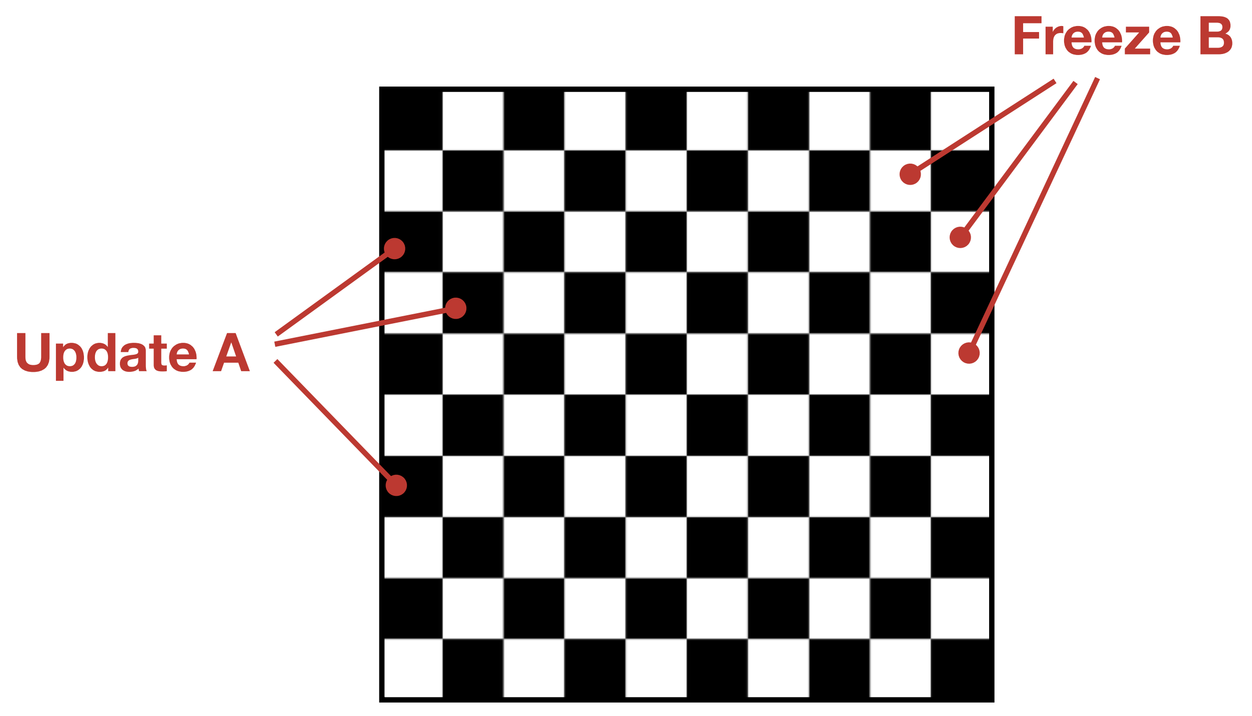
Scalar field $\phi(x) \in \mathbb{R}$, 1+1D spacetime

$$S[\phi] = \sum_x \partial_\mu \phi(x) \partial^\mu \phi(x) + \frac{M^2}{2} \phi(x)^2 + \lambda \phi(x)^4$$

**Machine learning jargon**

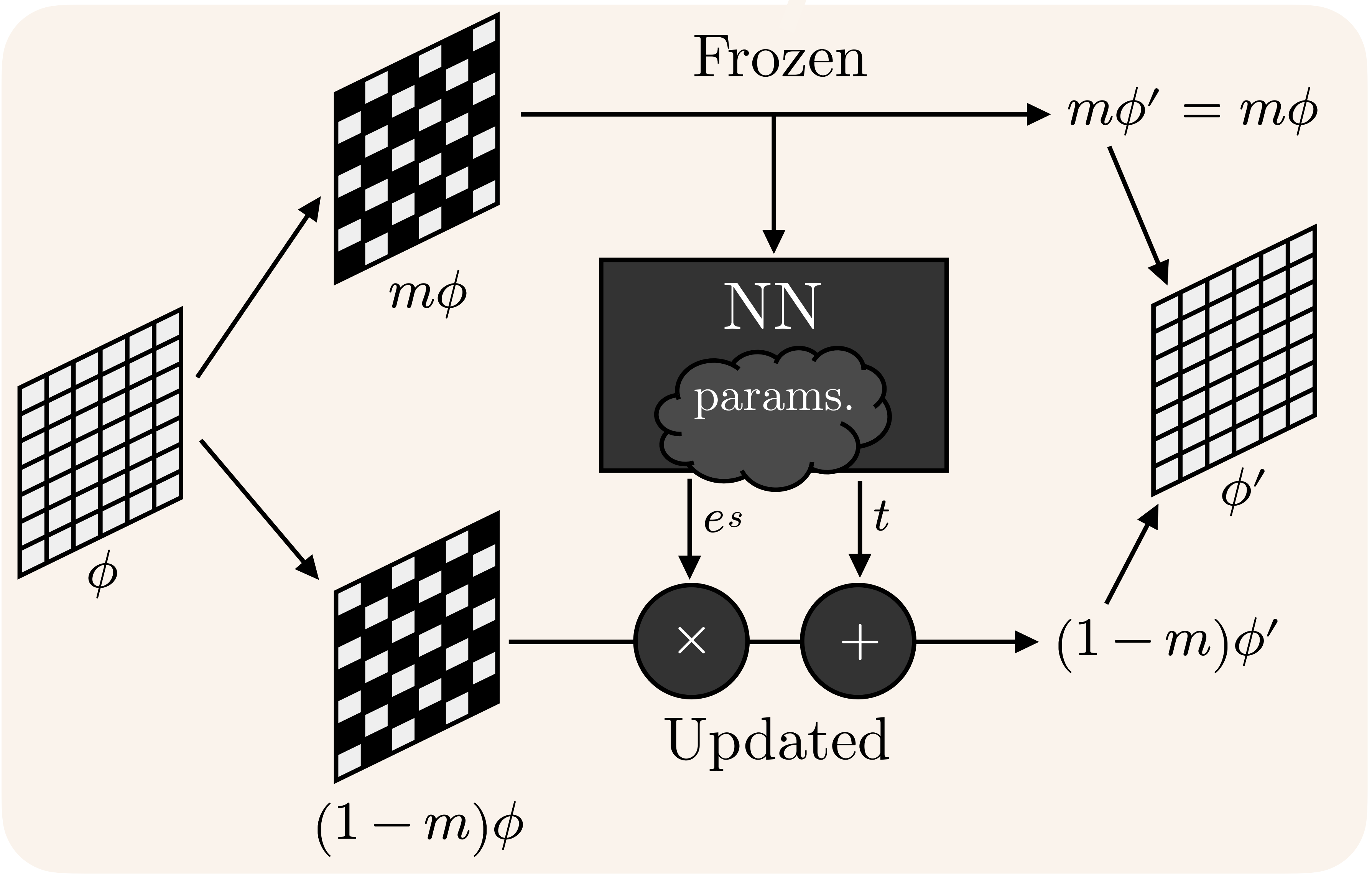
**Neural network (NN)** = highly parameterized function approximator, usually a composition of linear + elementwise non-linear transformations




Checkerboard masking pattern $m$

"Coupling layer" $g_i$




**Tractable Jacobian**

$$J_{ij} \equiv \partial \phi'_i / \partial \phi_j = \begin{bmatrix} I & \\ \blacksquare & \delta_{ij} e^{s_i} \end{bmatrix}$$

$$\implies \ln \det J = \sum_i s_i$$

Albergo, GK, Shanahan PRD100 (2019) 034515

# Flows for scalar $\phi^4$ theory

**Machine learning jargon**

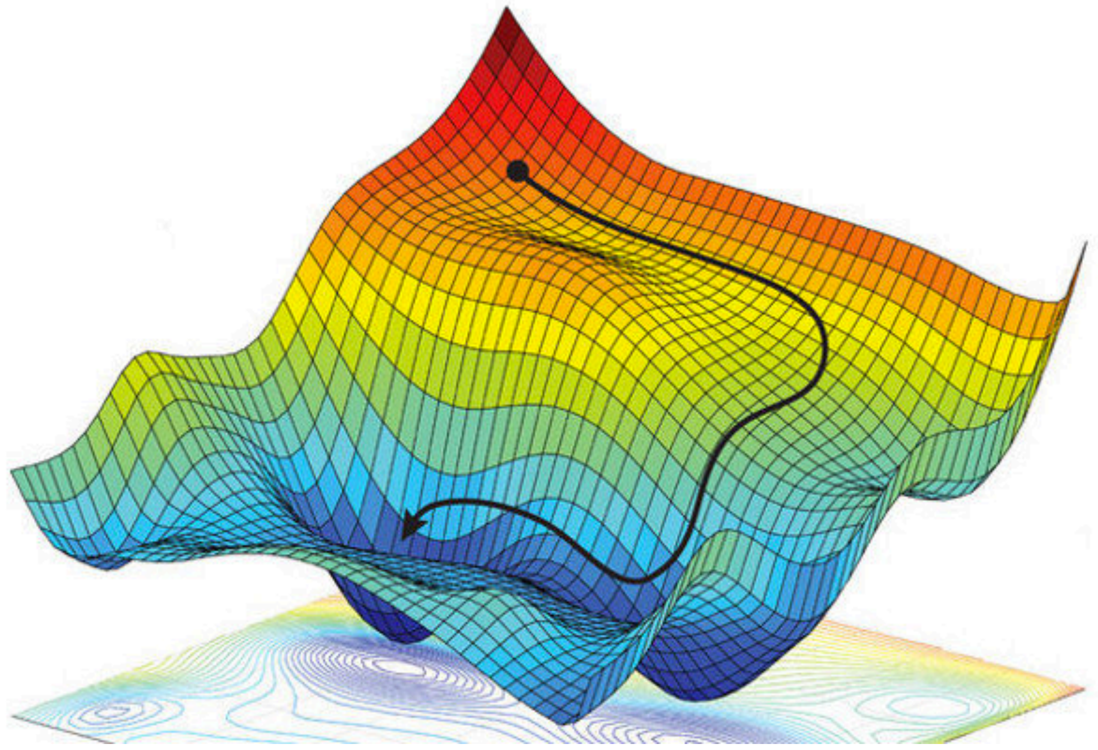
**Training** = optimization, typically by stochastic gradient descent

**Loss function** $\mathscr{L}$ = target function to be minimized

**Self-training** using Kullback-Leibler divergence between $p(U) = e^{-S[U]}/Z$ and $q(U)$

$$\mathscr{L} \equiv D'_{\mathrm{KL}}(q\,||\,p) = \int \mathscr{D}U\, q(U)\left[\log q(U) - \log e^{-S[U]}\right]$$

$$\vec{\omega}' = \vec{\omega} - \epsilon \nabla_{\vec{\omega}} \mathscr{L}$$



[Image credit: 1805.04829]

**Exactness** by reweighting or Metropolis

Albergo, GK, Shanahan PRD100 (2019) 034515
Nicoli+ PRE101 (2020) 023304

$$p_{\mathrm{acc}}(U \to U') = \min\left(1, \frac{p(U')}{q(U')}\frac{q(U)}{p(U)}\right)$$

# Flows for scalar $\phi^4$ theory

**Machine learning jargon**

**Training** = optimization, typically by stochastic gradient descent

**Loss function** $\mathscr{L}$ = target function to be minimized

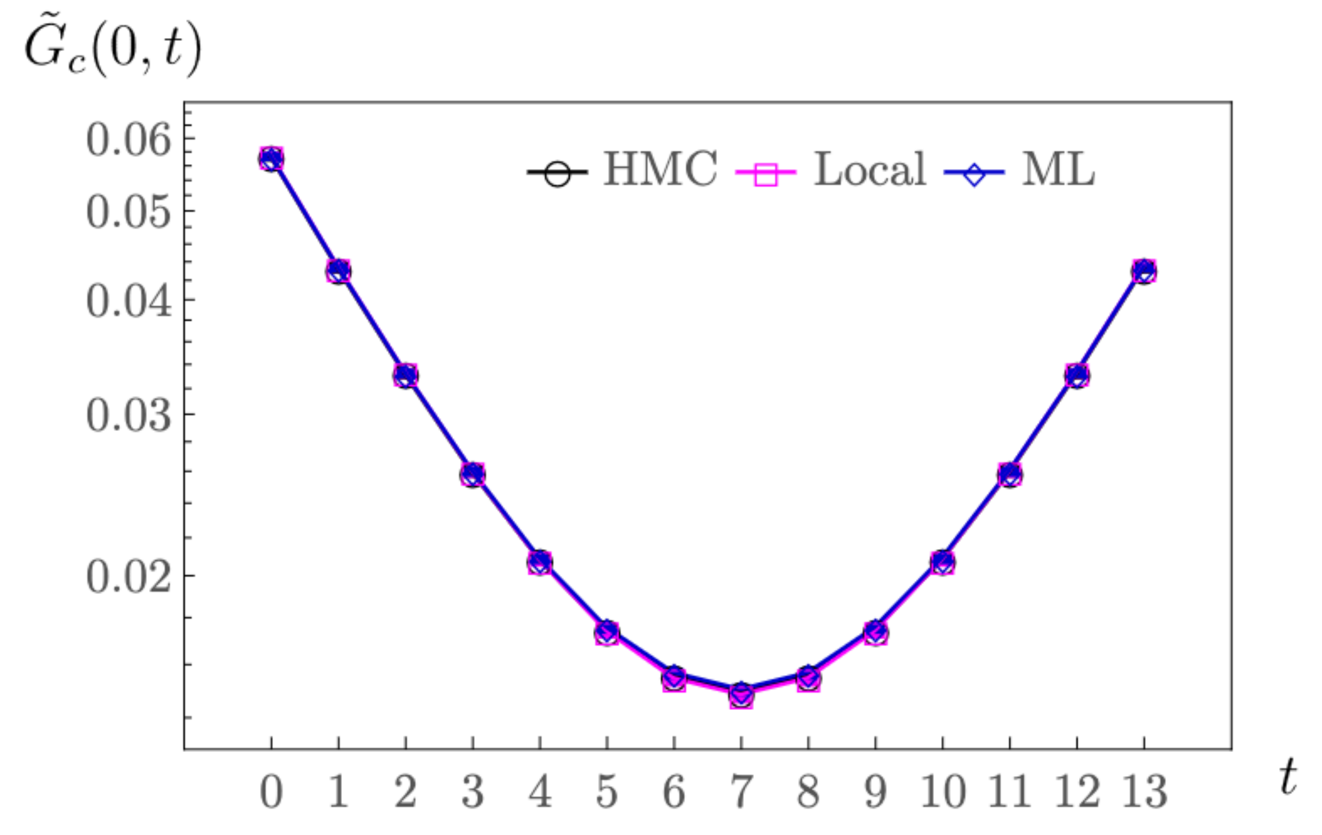
**Self-training** using Kullback-Leibler divergence between $p(U) = e^{-S[U]}/Z$ and $q(U)$

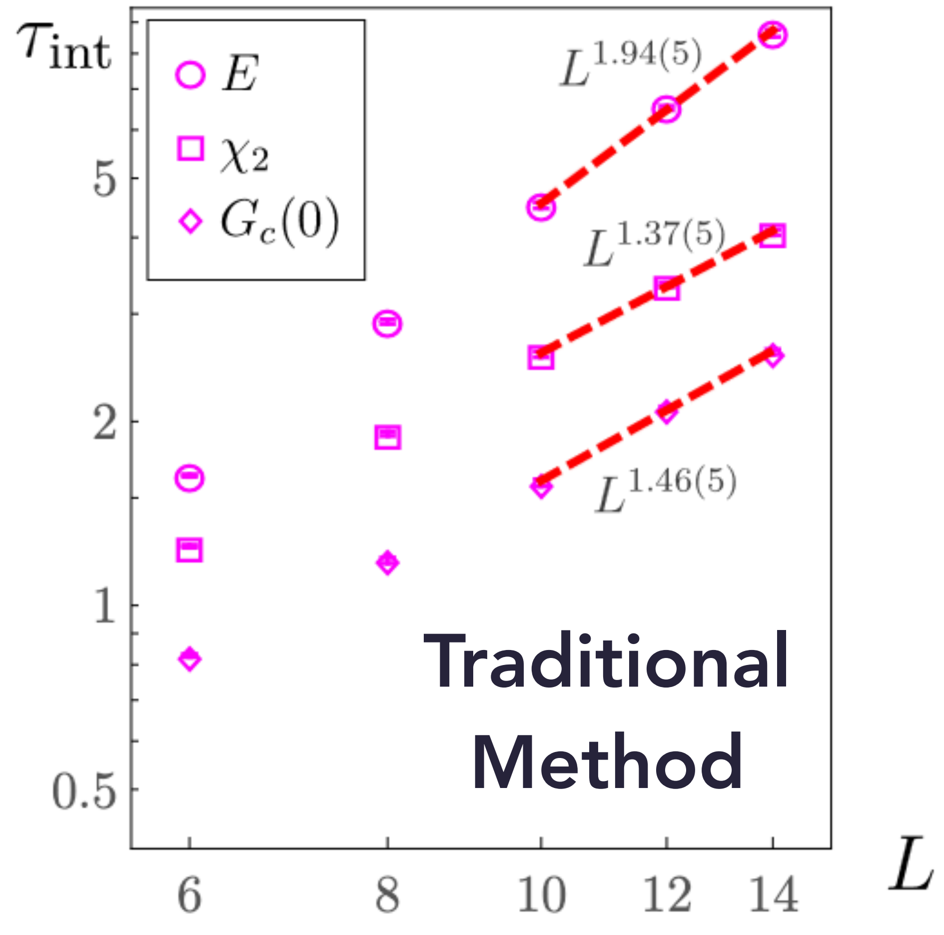
$$\mathscr{L} \equiv D'_{\mathrm{KL}}(q\,||\,p) = \int \mathscr{D}U\, q(U)\left[\log q(U) - \log e^{-S[U]}\right]$$

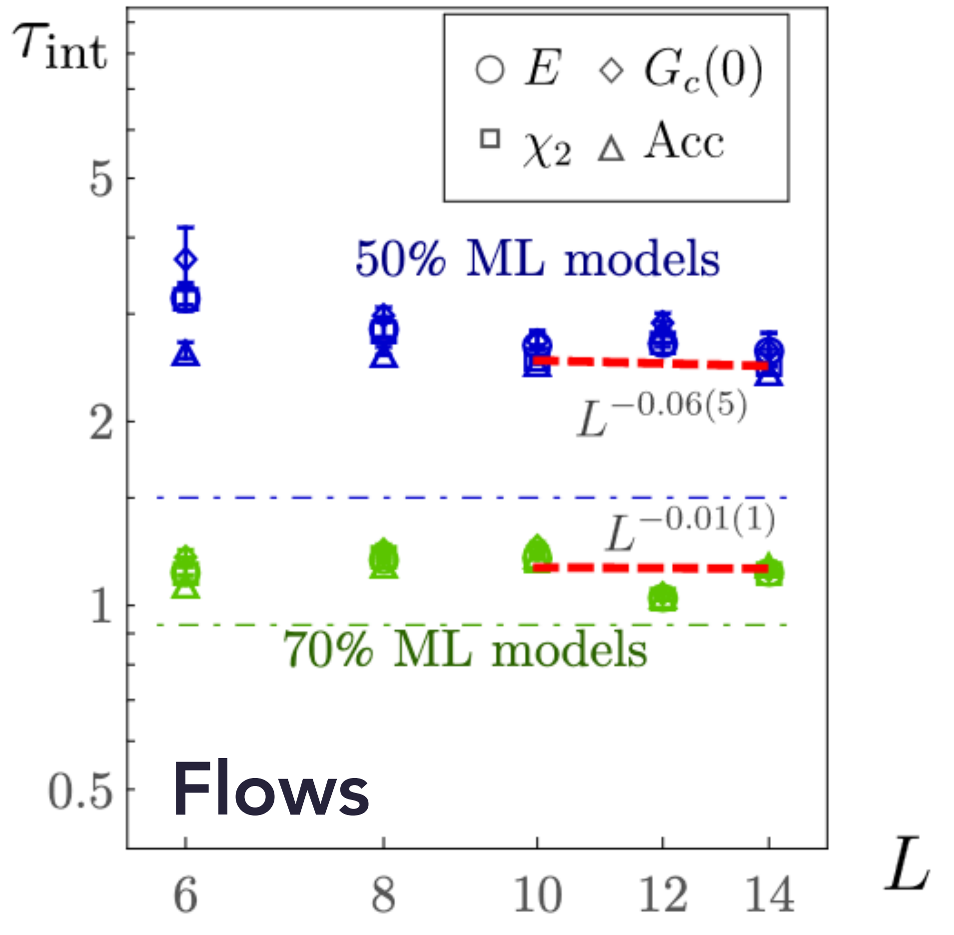
**Exactness** by reweighting or Metropolis

Albergo, GK, Shanahan PRD100 (2019) 034515
Nicoli+ PRE101 (2020) 023304

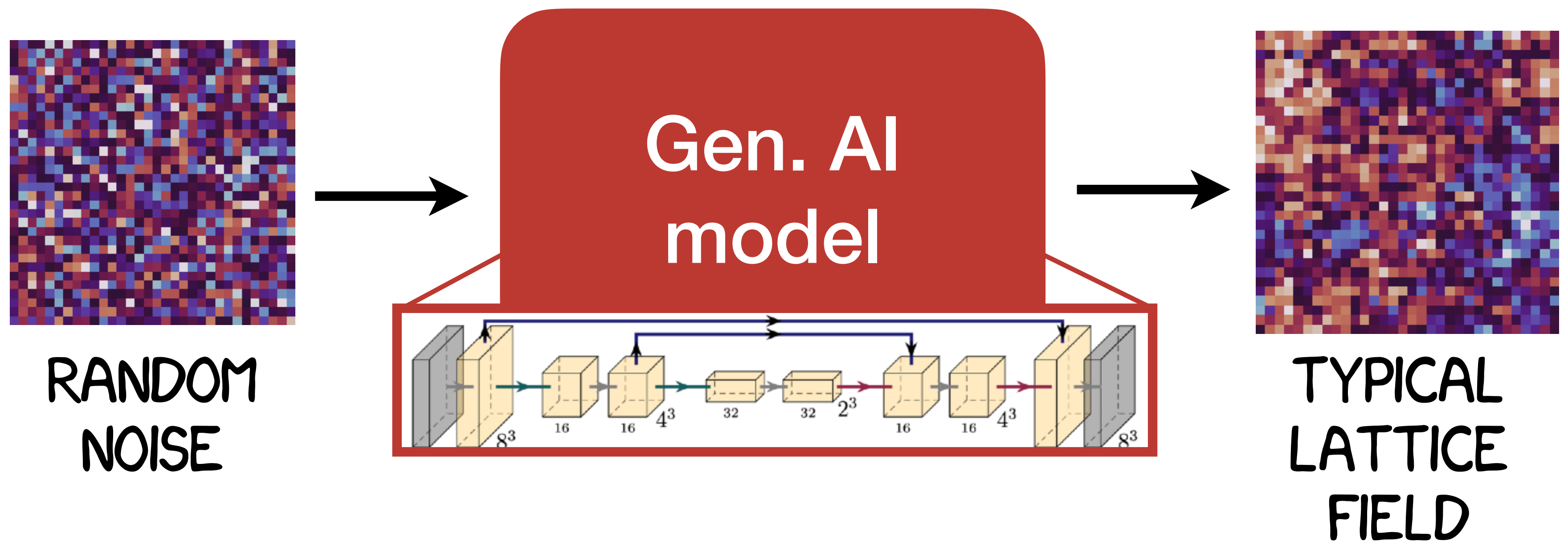
$$p_{\mathrm{acc}}(U \to U') = \min\left(1, \frac{p(U')}{q(U')}\frac{q(U)}{p(U)}\right)$$

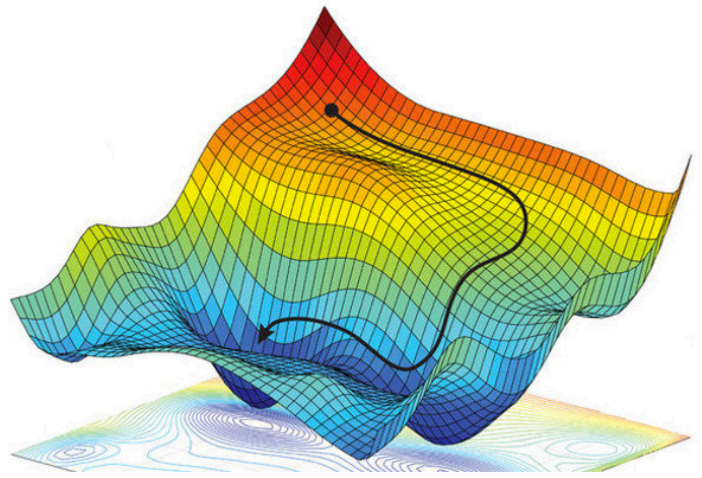


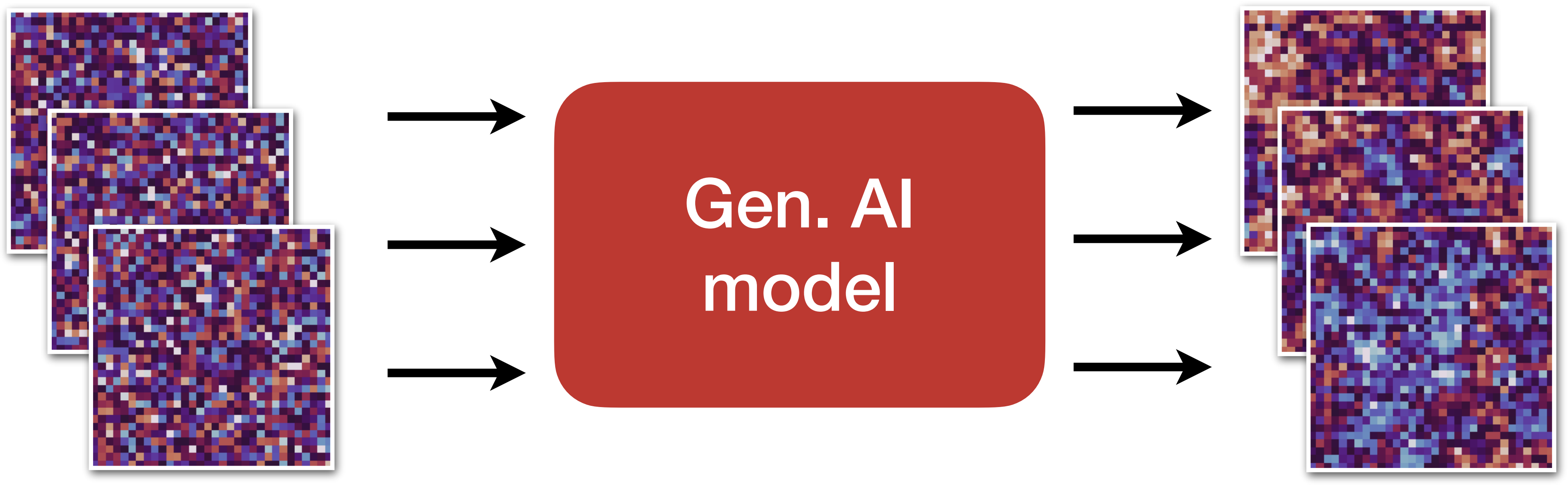



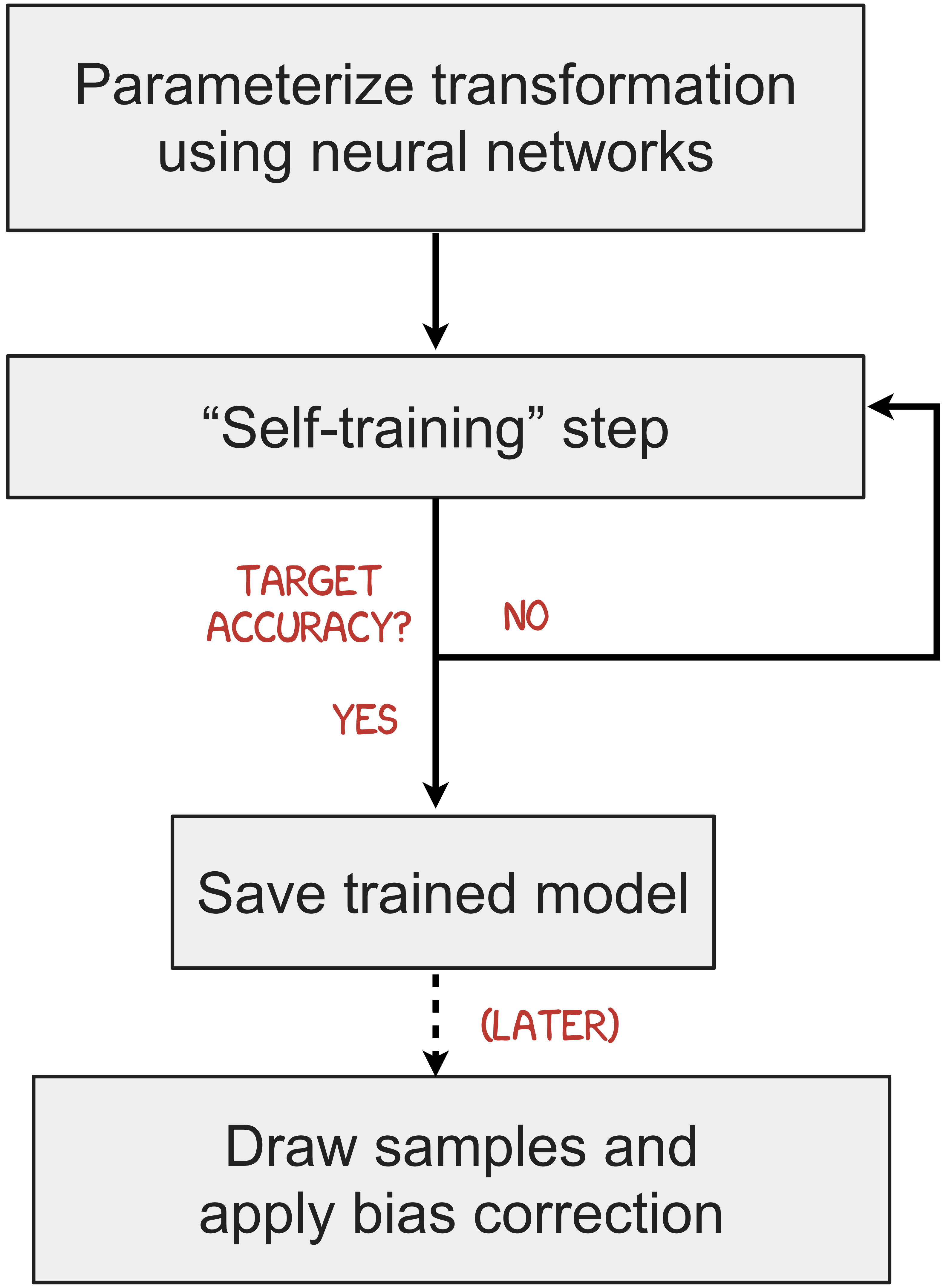



# Birds-eye view





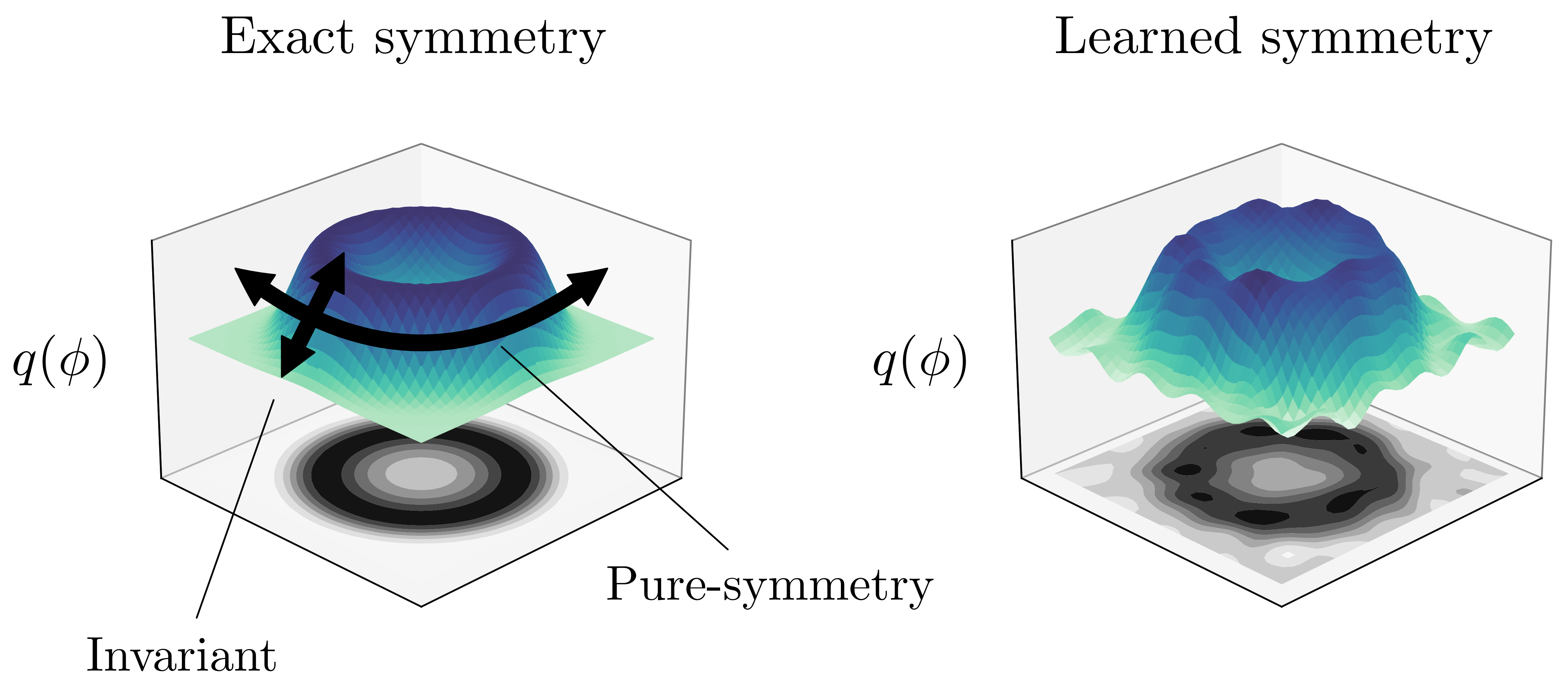
# Symmetries in flows

**Motivation:** Since target $p(\phi)$ is invariant under symmetries, natural to also make $q(\phi)$ invariant.

Lesson 2

Symmetries…

- ✓ Reduce data complexity of training
- ✓ Reduce model parameter count
- ✓ May make "loss landscape" easier




**Invariant** prior + **equivariant** flow = symmetric flow model

$r(t \cdot \phi) = r(\phi)$ $\quad$ $f(t \cdot \phi) = t \cdot f(\phi)$

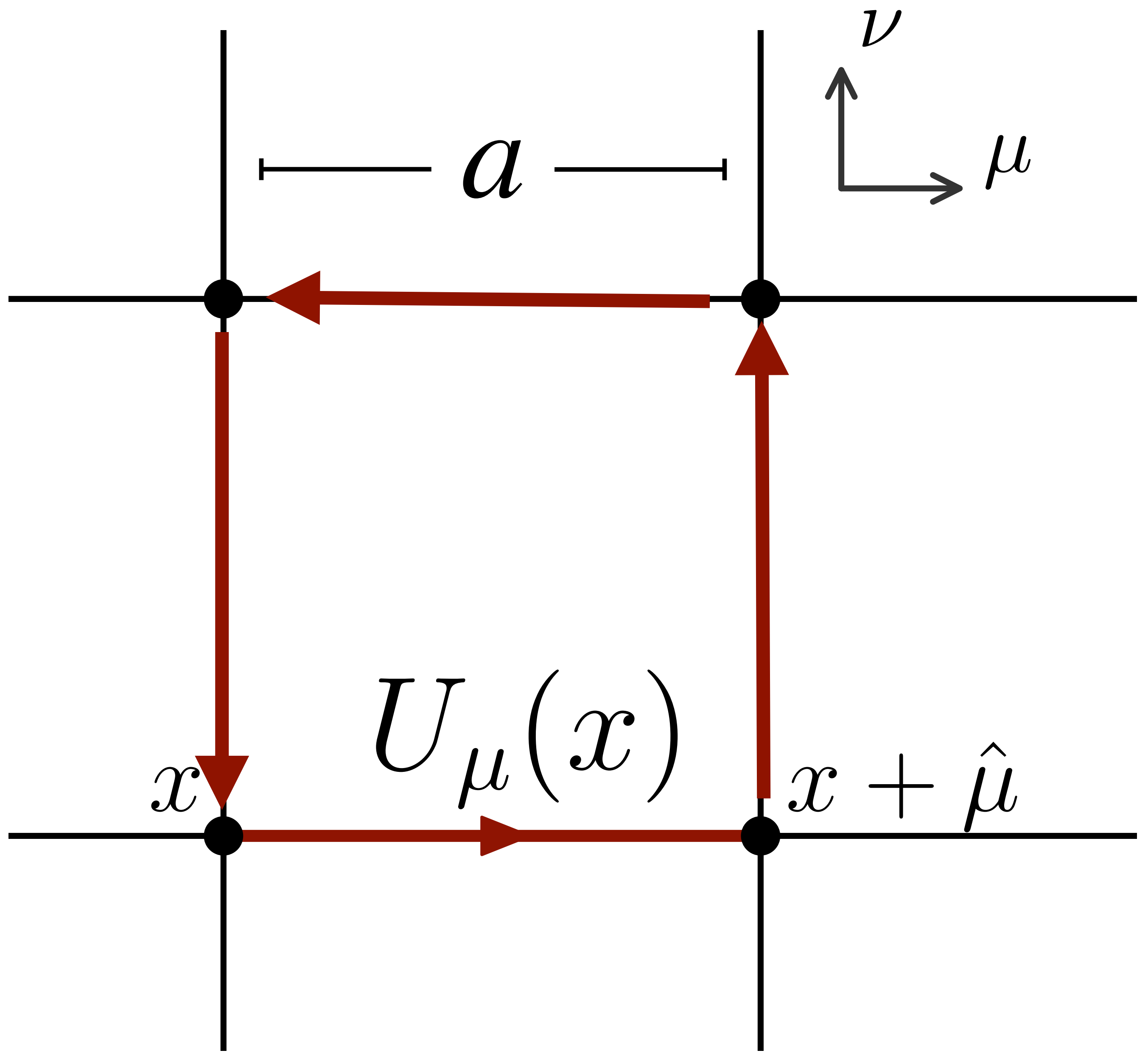
Cohen, Welling 1602.07576

# SU(3) gauge symmetry in QCD

Lattice action in the gluon sector

$$S(U) = -\frac{\beta}{3}\sum_{x}\sum_{\mu<\nu}\mathrm{ReTr}\,P_{\mu\nu}(x)$$

- Gluon self-interaction dynamics (Yang-Mills)
- Confinement, topological instantons




$$P_{\mu\nu}(x) = U_\mu(x)U_\nu(x+\hat{\mu})U_\mu^\dagger(x+\hat{\nu})U_\nu^\dagger(x)$$

Lattice gauge symmetry

$$U_\mu(x) \mapsto \Omega(x)U_\mu(x)\Omega^\dagger(x+\hat{\mu})$$

# Gauge symmetry

*Many lattice QFTs possess a large gauge symmetry group.*

Gauge symmetry for $\mathrm{SU}(3)$
lattice gauge theory

$$U_\mu(x) \mapsto \Omega(x) U_\mu(x) \Omega^\dagger(x+\hat{\mu})$$

GK, et al. PRL125 (2020) 121601

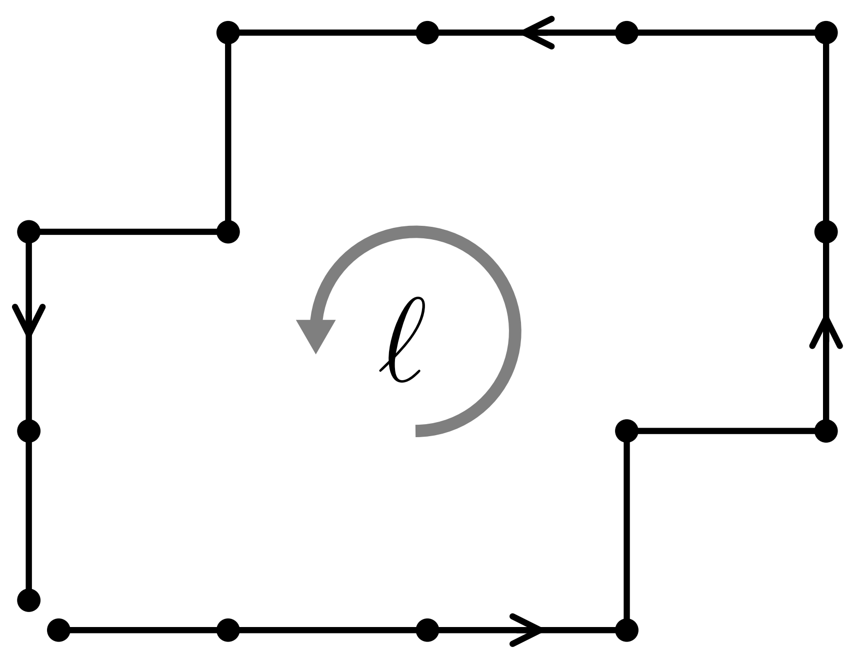
**Gauge-invariant prior:**

Uniform (Haar) distribution
$r(U) = 1$ works.

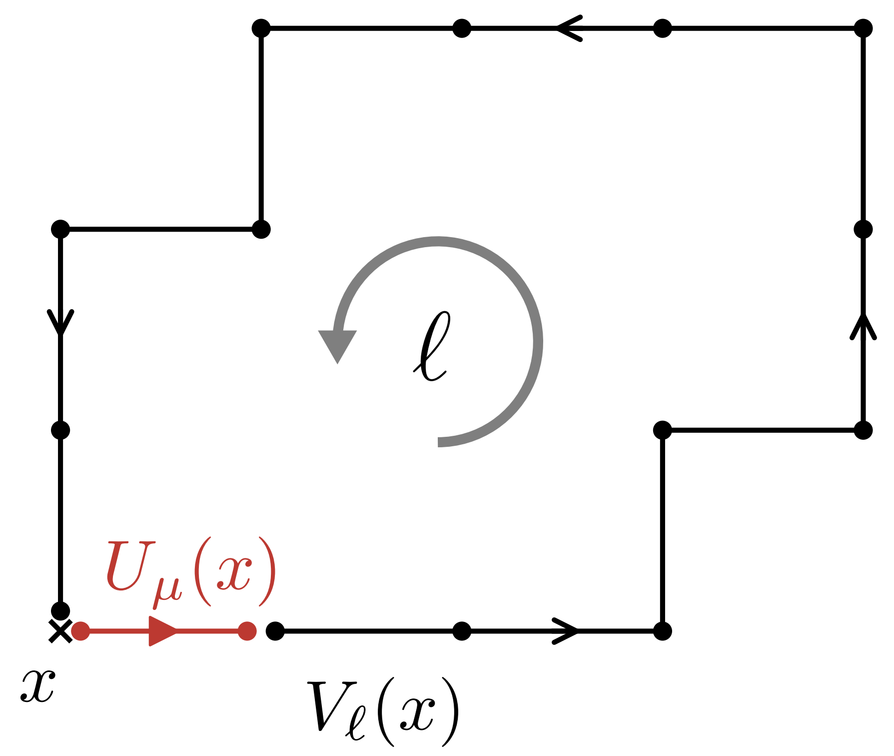
**Gauge-equivariant flow:**

Coupling layers acting on
(untraced) Wilson loops.

Loop transformation easier to satisfy.




$$W_\ell(x) \xrightarrow{\text{Flow}} W'_\ell(x)$$




$$U'_\mu(x) = W'_\ell(x)\, V_\ell^\dagger(x)$$

# Gauge symmetry

*Many lattice QFTs possess a large gauge symmetry group.*

Gauge symmetry for $\mathrm{SU}(3)$ lattice gauge theory

$$U_\mu(x) \mapsto \Omega(x) U_\mu(x) \Omega^\dagger(x+\hat{\mu})$$

**Gauge-invariant prior:**

Uniform (Haar) distribution $r(U) = 1$ works.

**Gauge-equivariant flow:**

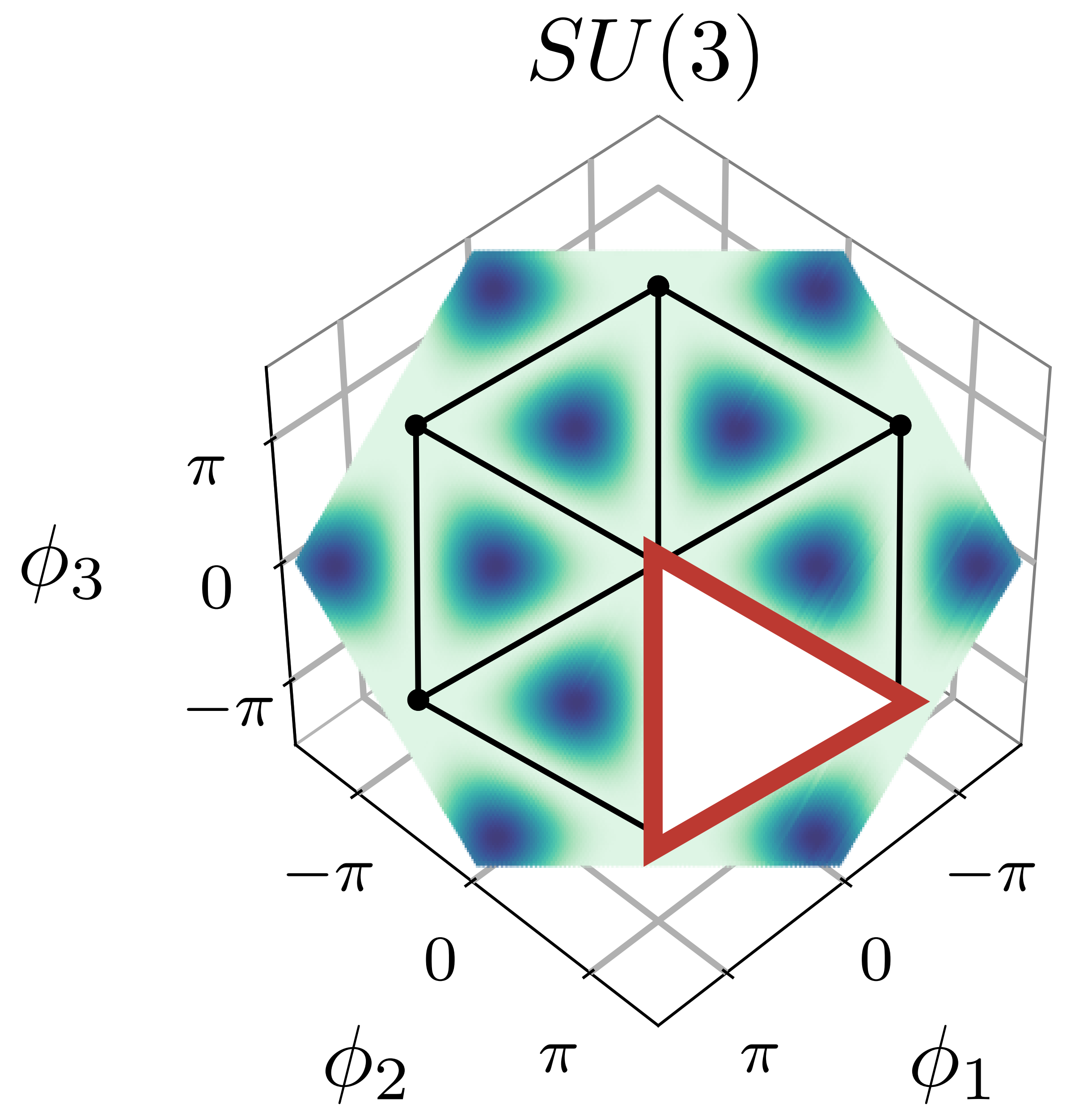
Coupling layers acting on (untraced) Wilson loops.

Loop transformation easier to satisfy.

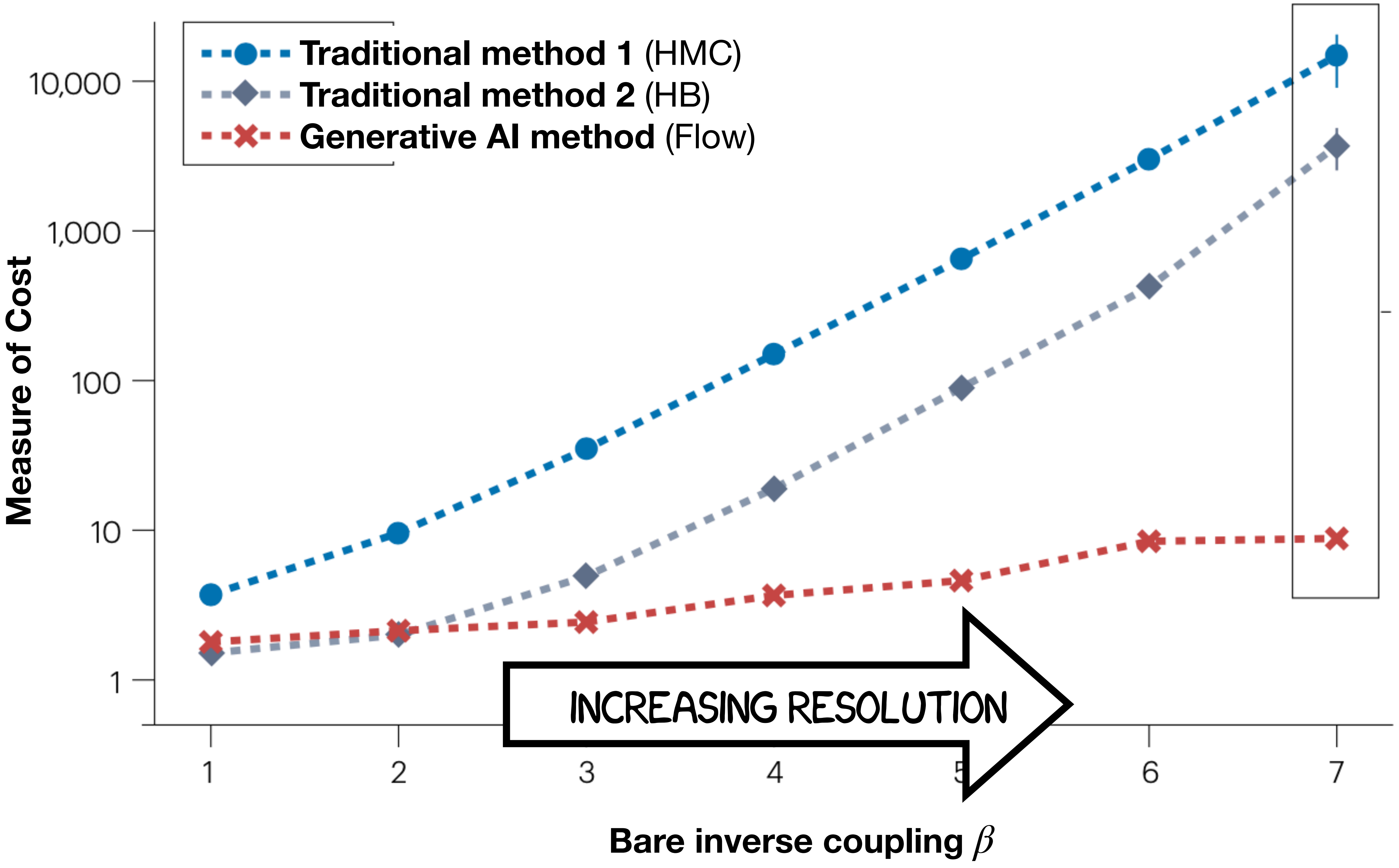
Custom flows designed for $U(1)$ and $SU(N)$ gauge manifolds

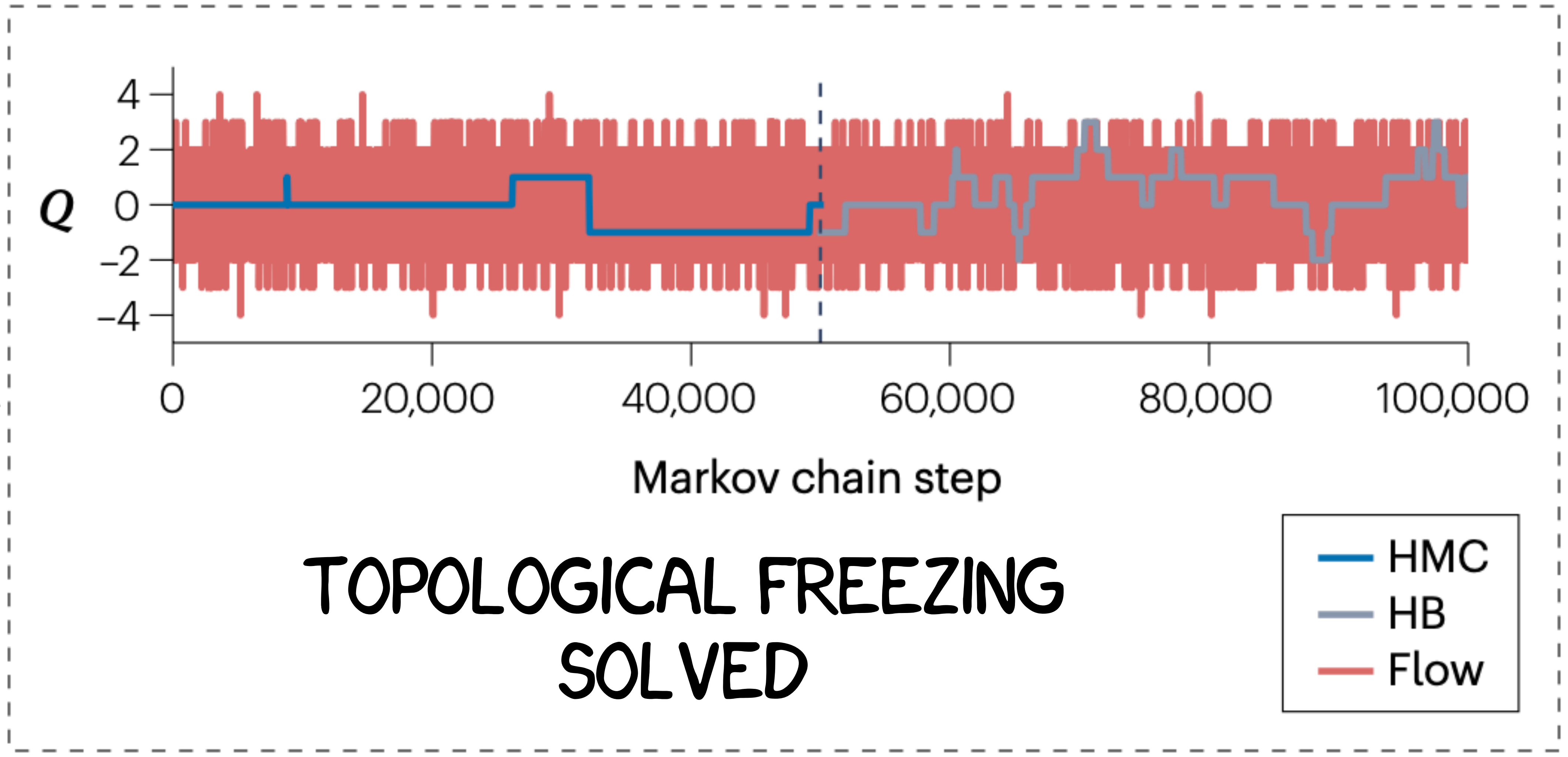
GK, et al. PRL125 (2020) 121601

Rezende, et al. PMLR119 (2020) 8083
Boyda, et al. PRD103 (2021) 074504



# Sampling for U(1) lattice gauge theory



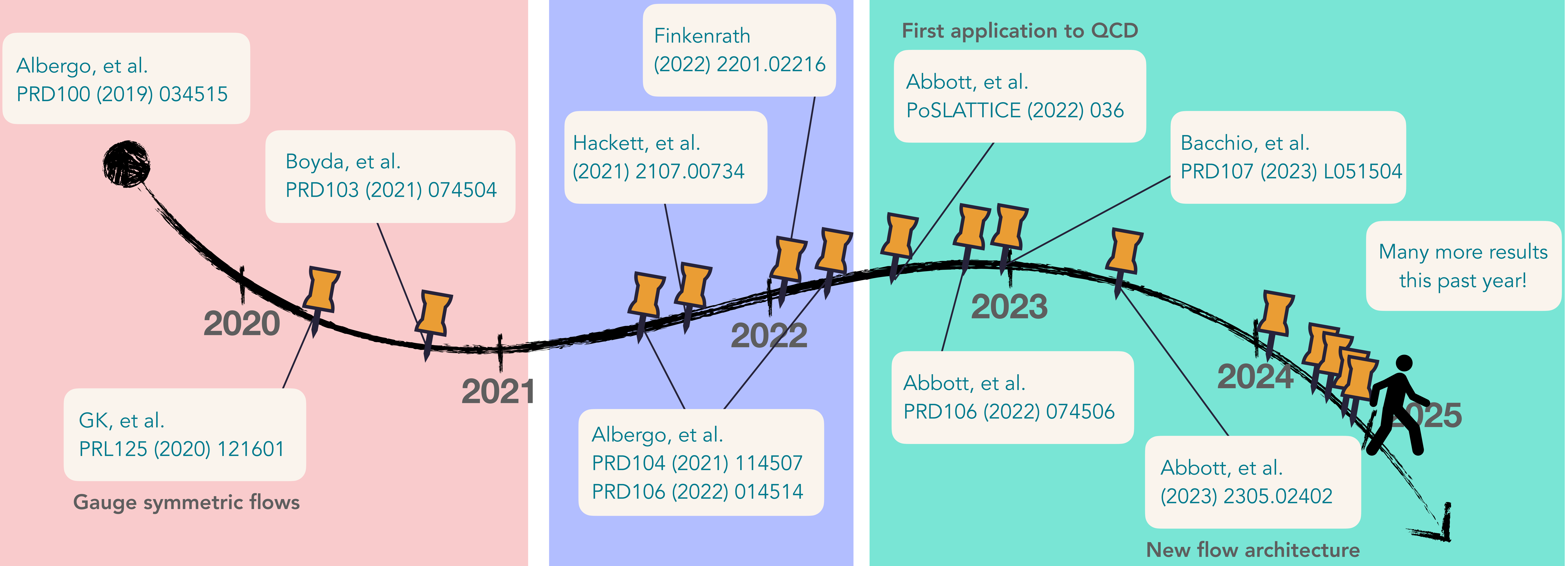

Lesson 3

- **Transfer learning**: Model for previous target $\beta$ used to initialize for next target

- Also applied successfully to $SU(N)$ gauge theories.

Cranmer, GK, Racanière, Rezende, Shanahan, Nature Reviews Physics 5 (2023) 526

# Building up to QCD applications




- Viability to solve CSD
- Gauge theories
- 2D spacetime

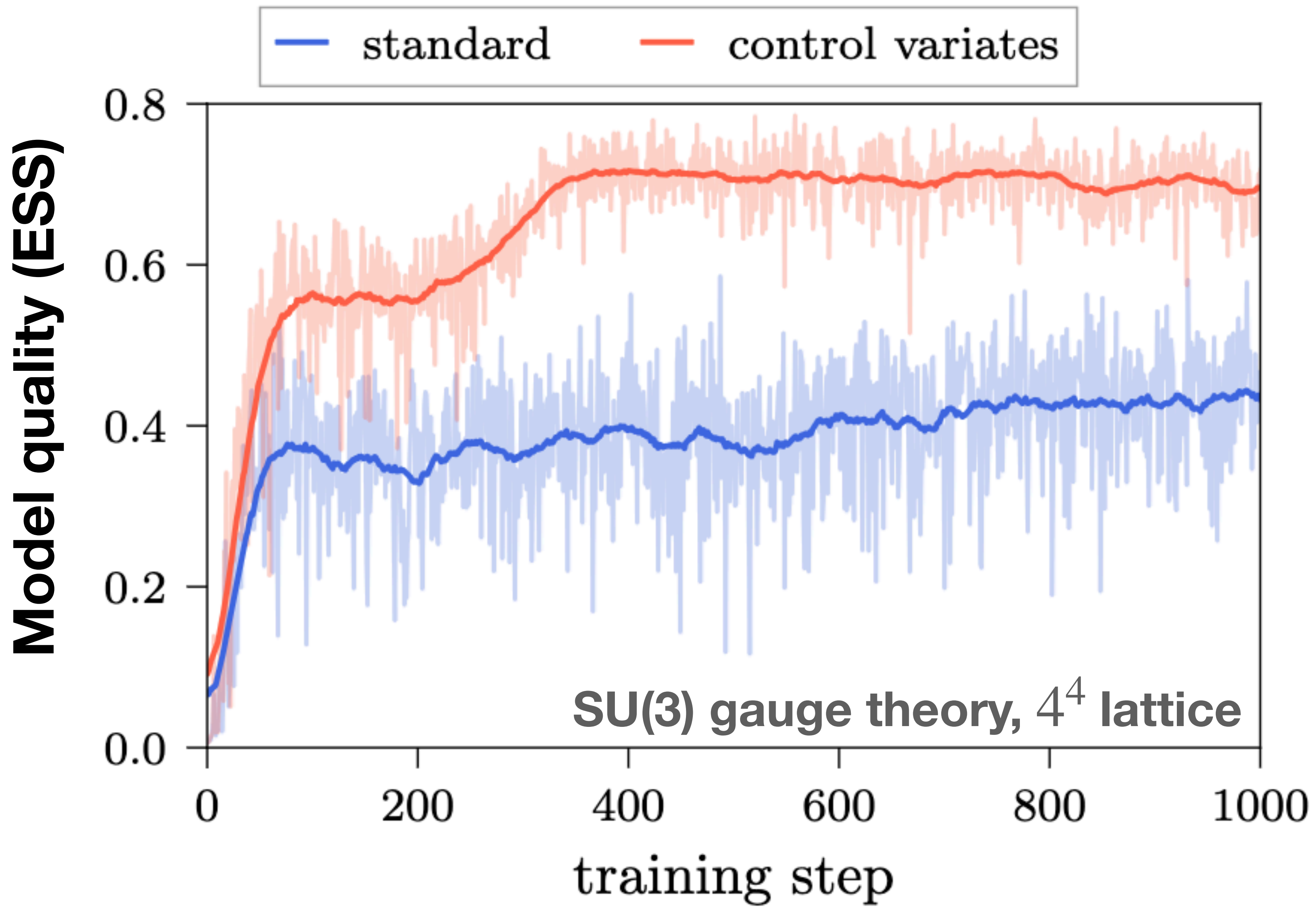
- Fermions
- Broken symmetry

- 4D spacetime
- Refined flow models
- Better training

- Integrating flows into practical simulations

# Recent developments

- Better training procedures
  - Minimize gradient noise with control variates or path gradients
    Vaitl, Nicoli, Nakajima, Kessel (2022) 2207.08219
    Białas, Korcyl, Stebel (2022) 2202.01314




Abbott, et al. (2023) 2305.02402

- "Residual flows"
  - Flow = Discrete steps according to gradient of scalar function $\hat{S}(\phi)$
  - Symmetries easier to encode
  - Relation to trivializing map, continuous flows
    Lüscher CMP293 (2010) 899
    Bacchio, Kessel, Schaefer, Vaitl PRD107 (2023) L051504

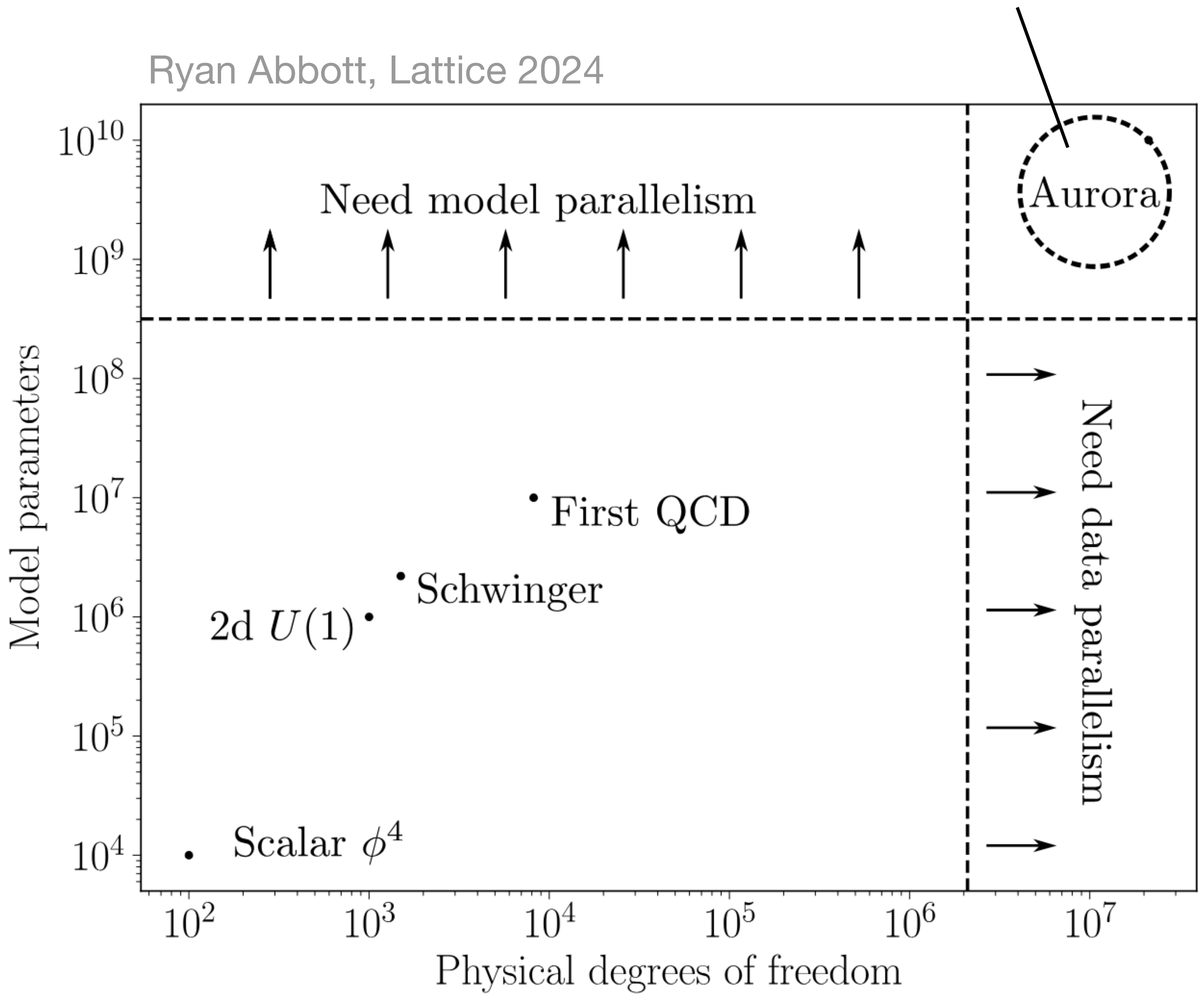
# To the exascale


63,744 Intel GPUs, ~1 exaflop performance

We are running this year

- Significant software effort
- Large models with $O(10^9)$ params
- New simulation targets

Ryan Abbott, Lattice 2024



# Lessons in distilled form

# Lessons in distilled form

**Lesson 1** *Design training schemes around the features of the problem.*

- Self-training very important for future of this method

# Lessons in distilled form

**Lesson 1** *Design training schemes around the features of the problem.*

- Self-training very important for future of this method

**Lesson 2** *Incorporate physics constraints and information when possible.*

- Gauge symmetries were a breakthrough in applying to gauge theories
- Counter to the Bitter Lesson (Richard Sutton)

"We have to learn the bitter lesson that building in how we think we think does not work in the long run"

# Lessons in distilled form

**Lesson 1** *Design training schemes around the features of the problem.*

- Self-training very important for future of this method

**Lesson 2** *Incorporate physics constraints and information when possible.*

- Gauge symmetries were a breakthrough in applying to gauge theories
- Counter to the Bitter Lesson (Richard Sutton)

"We have to learn the bitter lesson that building in how we think we think does not work in the long run"

**Lesson 3** *Amortize costs as much as possible*

- Transfer learning between targets
- Larger models encoding more general information

# Lessons in distilled form

Thank you!

**Lesson 1** *Design training schemes around the features of the problem.*

- Self-training very important for future of this method

**Lesson 2** *Incorporate physics constraints and information when possible.*

- Gauge symmetries were a breakthrough in applying to gauge theories
- Counter to the Bitter Lesson (Richard Sutton)

"We have to learn the bitter lesson that building in how we think we think does not work in the long run"

**Lesson 3** *Amortize costs as much as possible*

- Transfer learning between targets
- Larger models encoding more general information

# Backup slides

# Related approaches

Generative Adversarial Networks (GANs):

- Highly expressive
- Work in the direction of GANs for lattice

  Urban, Pawlowski 1811.03533

  Zhou, Endrődi, Pang, Stöcker 1810.12879

Variational AutoEncoders (VAEs):

- Can also learn meaningful directions in the prior variables

**However:** No access to $q(\phi)$… hard to make exact!

Karras, Lane, Aila / NVIDIA 1812.04948

AI-generated faces (GAN)

Shen & Liu 1612.05363

AI-generated faces (VAE)

# Coupling layers

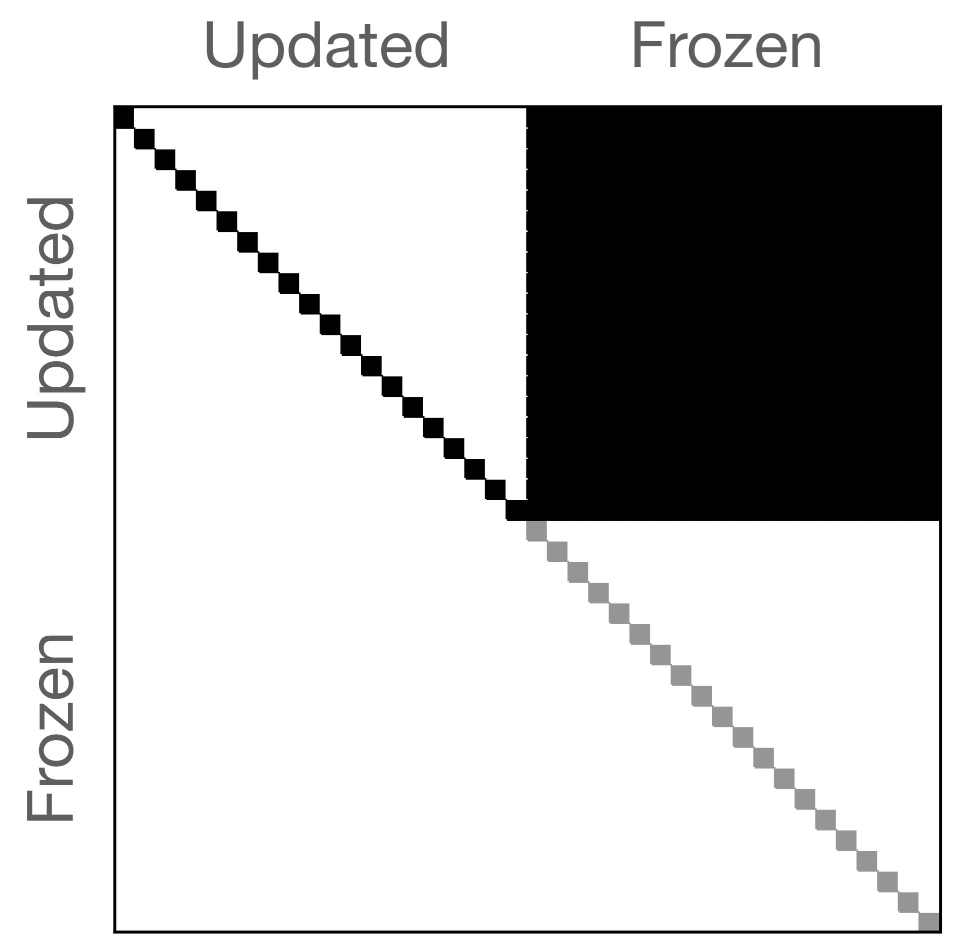
**Idea:** Construct each $g$ to act on a **subset of components**, conditioned only on the complimentary subset. “Masking pattern” $m$ defines subsets.

→ Jacobian is explicitly upper-triangular (get det J from diag elts)

“Updated” $(m_i = 0)$ “Frozen” $(m_i = 1)$

$$\frac{\partial[g(V)]_i}{\partial V_j} = \left(\begin{array}{ccc|ccc} \frac{\partial[g(V)]_1}{\partial V_1} & & & & & \\ & \frac{\partial[g(V)]_2}{\partial V_2} & & & \text{(nonzero)} & \\ & & \ddots & & & \\ \hline & & & 1 & & \\ & 0 & & & 1 & \\ & & & & & \ddots \end{array}\right)$$

Schematically

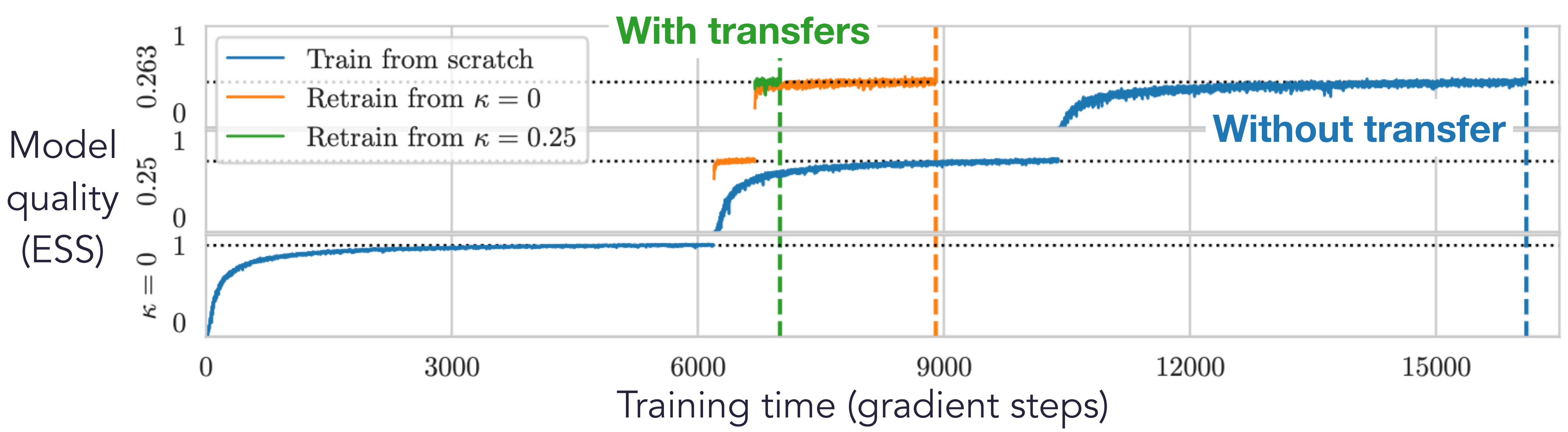



→ Invertible if each diag component invertible, $\partial[g(V)]_i/\partial V_i \neq 0$.

# Transfer learning

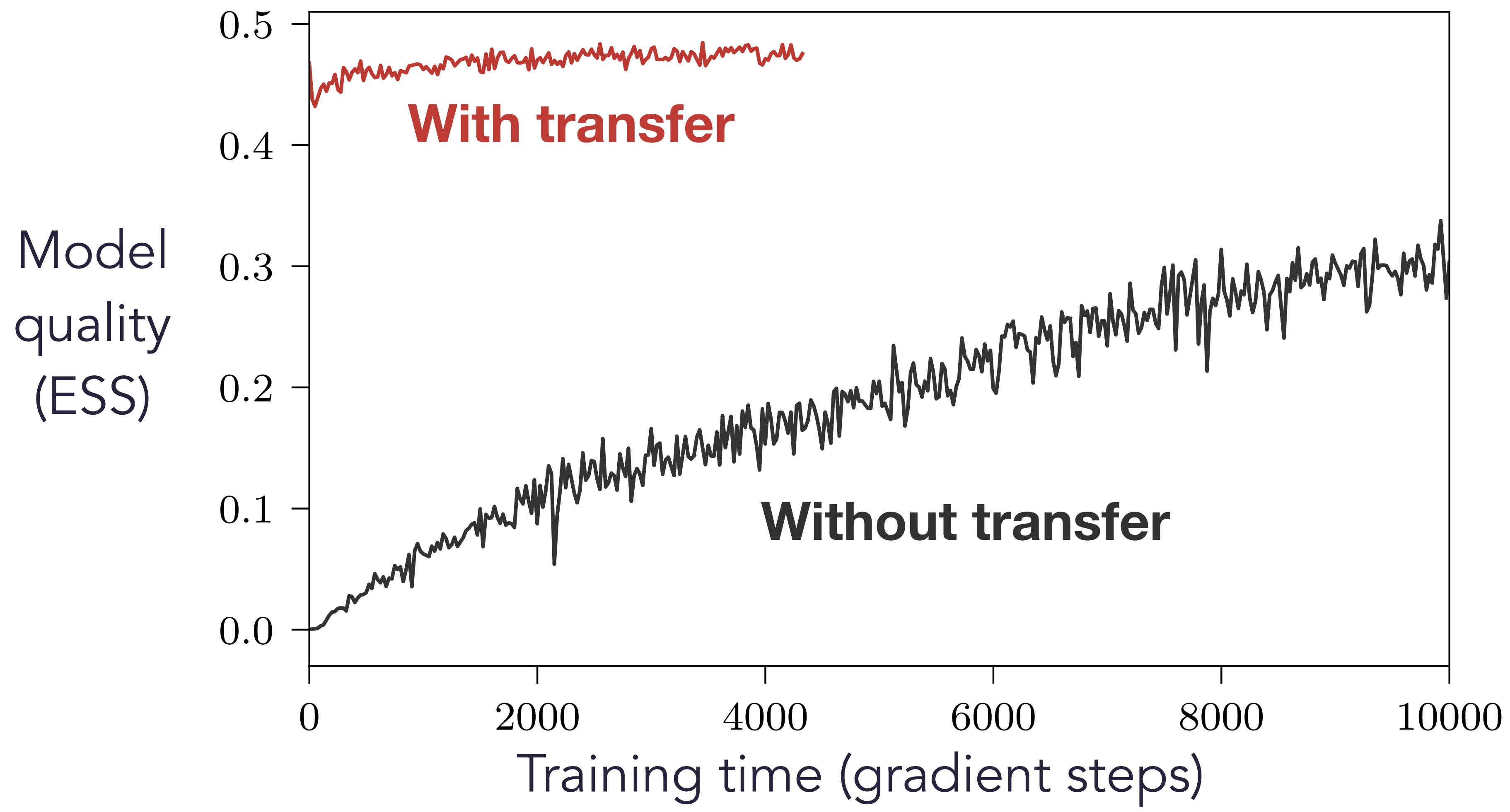
Both parameter transfer and volume transfer are highly effective for lattice field theory.

Abbott, et al. 2211.07541




- Schwinger model [U(1) gauge theory + fermions]
- Parameter transfer $\kappa = 0 \rightarrow 0.25 \rightarrow 0.263(\kappa_{cr})$

Boyda, GK, … PRD103 (2021) 074504

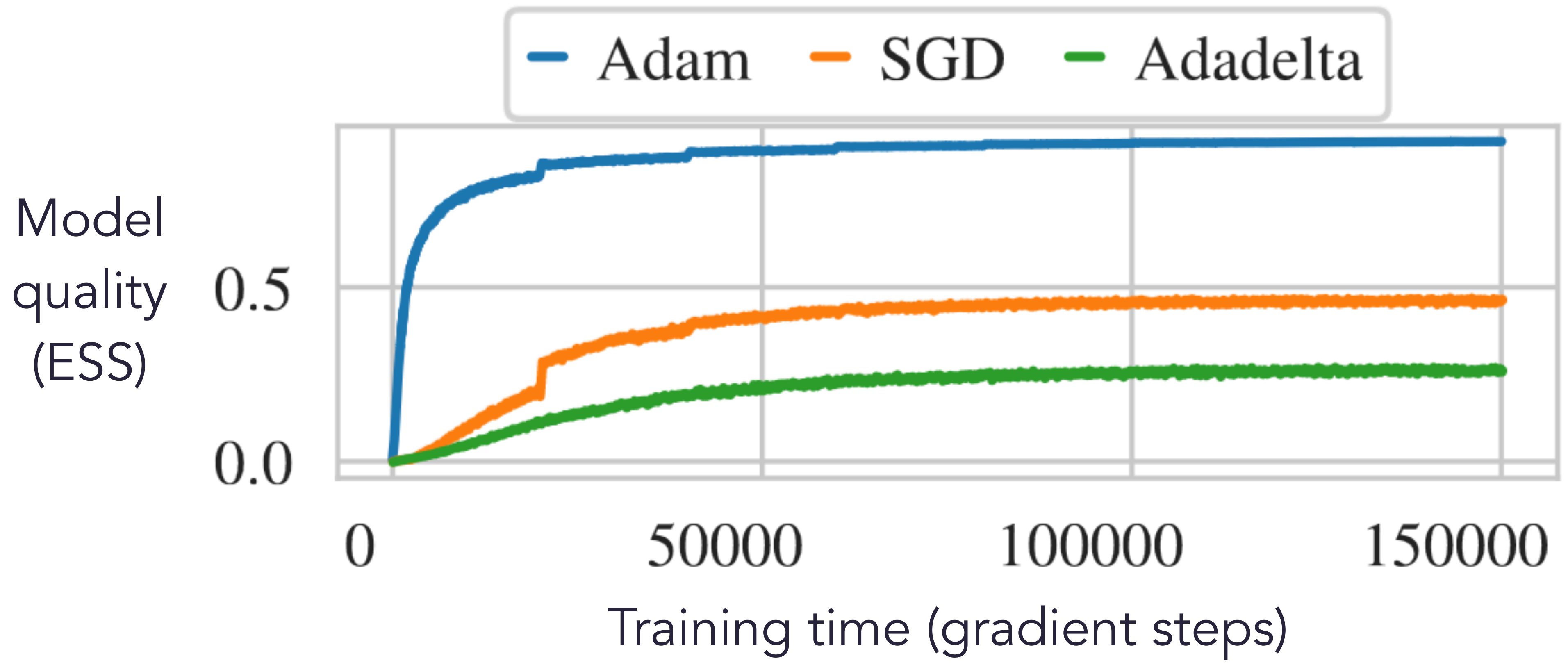


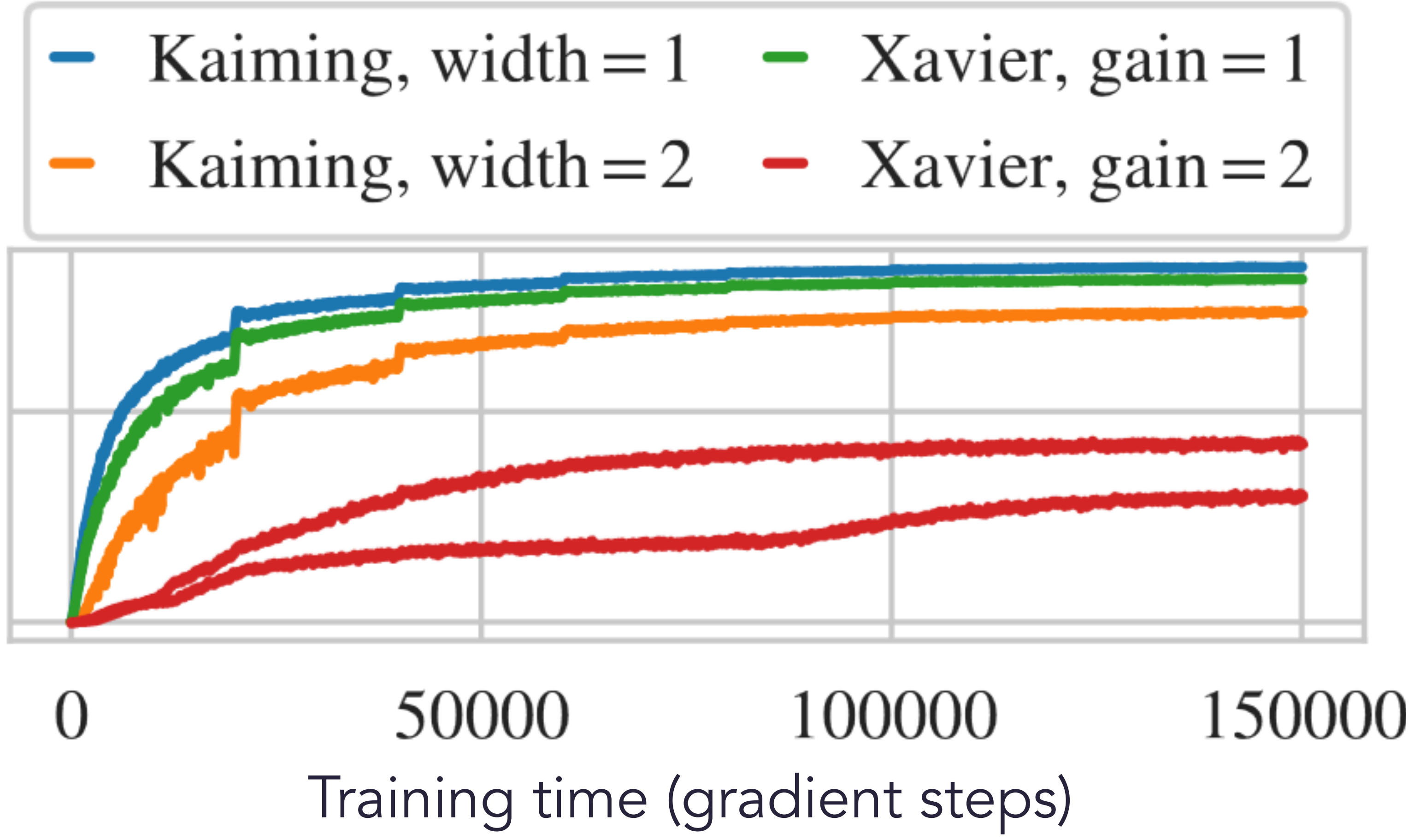

- SU(N) gauge theory
- Volume transfer $8 \times 8 \rightarrow 16 \times 16$ (red)
- Directly start at $16 \times 16$ (black)

# Hyperparameters can make a big difference

Optimization algorithm, hyperparameters,
and initialization have strong effects on training rate.

Abbott, et al. 2211.07541



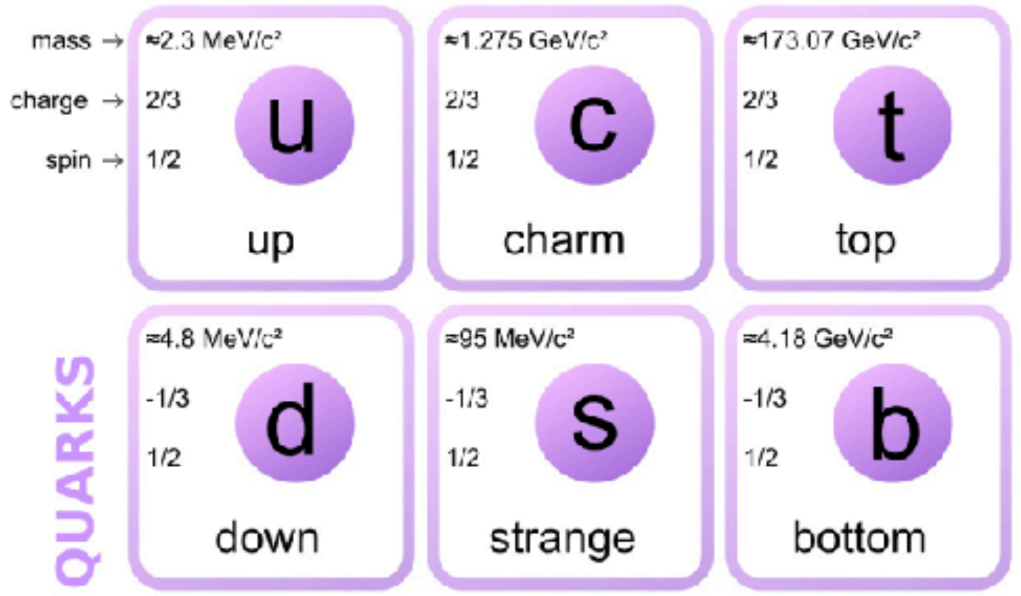



# Including the quarks

Interaction between all quark flavors ($\psi_u, \psi_d, \ldots$) and gluons ($U$):

Action $$S_f = \sum_f \bar{\psi}_f D_f[U] \psi_f$$

Path integral $$\int \prod_f [d\bar{\psi} d\psi] e^{-S_f} = \prod_f \det(D_f[U])$$




- $D_f$ is a sparse $O(V) \times O(V)$ matrix
- Traditional methods use the **pseudofermion** representation

$$|\det(D)|^2 \propto \int d\phi^\dagger d\phi e^{-\phi^\dagger (D^\dagger D)^{-1} \phi}$$

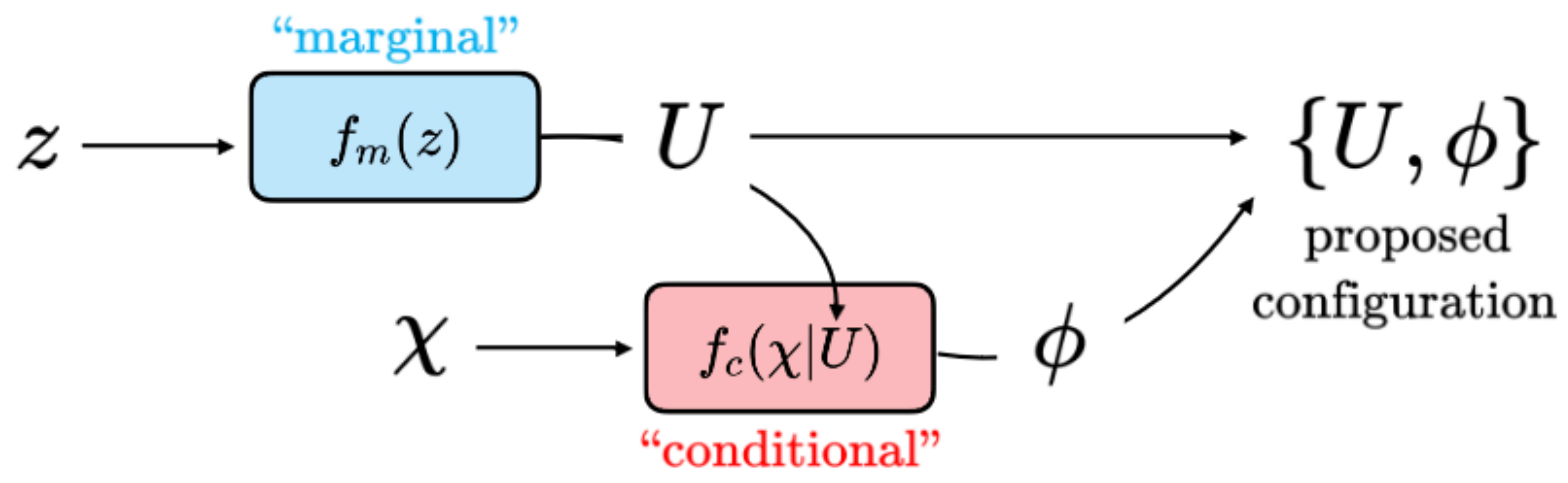
PRD104 (2021) 114507
PRD106 (2022) 014514
PRD106 (2022) 074506

# Flows with pseudofermions

Pseudofermions highly effective in HMC, logical to use for flows also.

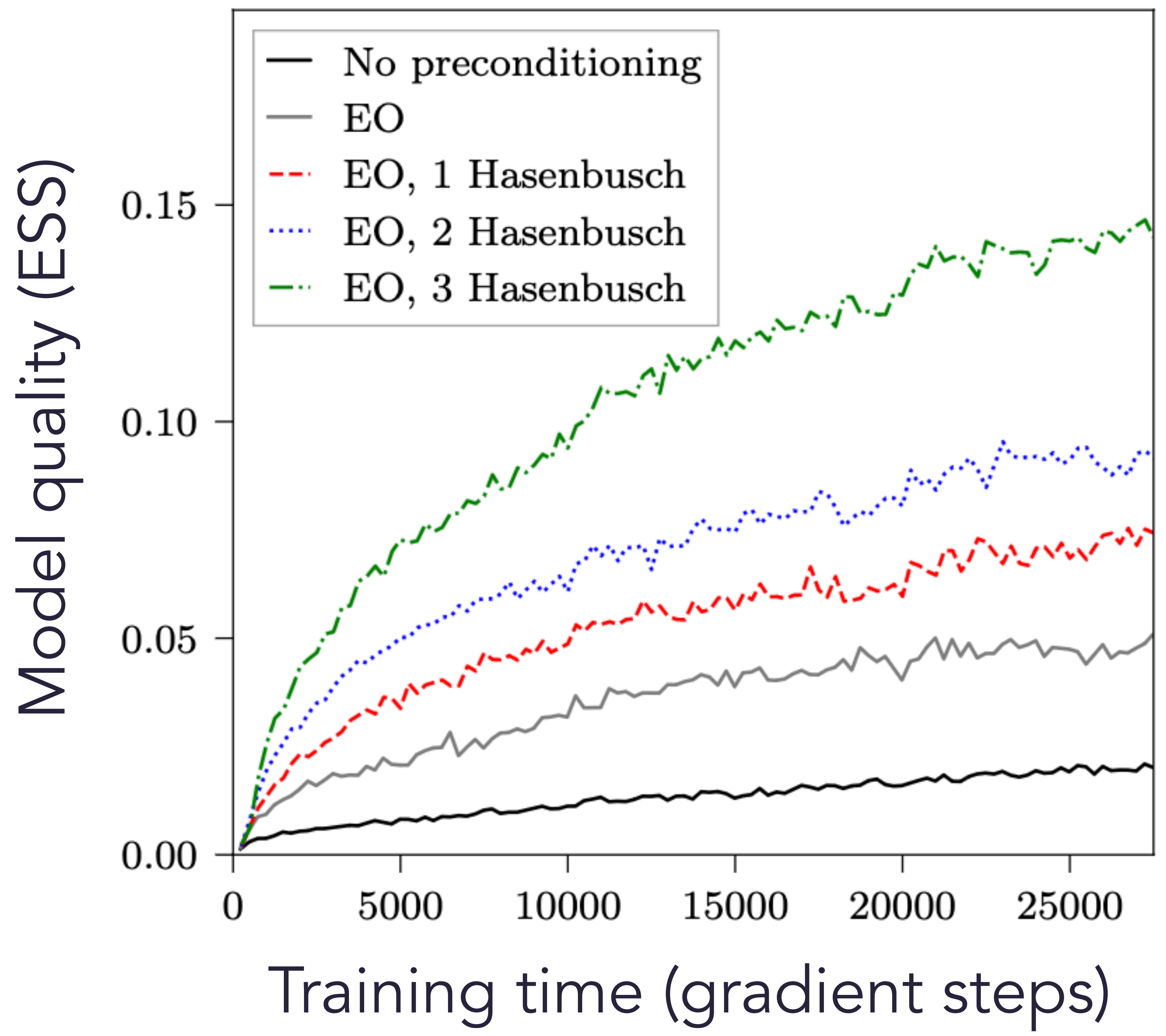
Separate coupling layers for gauge field and PFs can be composed arbitrarily

- **Simplest case:** marginal + conditional model




- **Preconditioning** works equally well for flows

- Modified Metropolis allows averaging away noise in the conditional flow



# Beyond critical slowing down

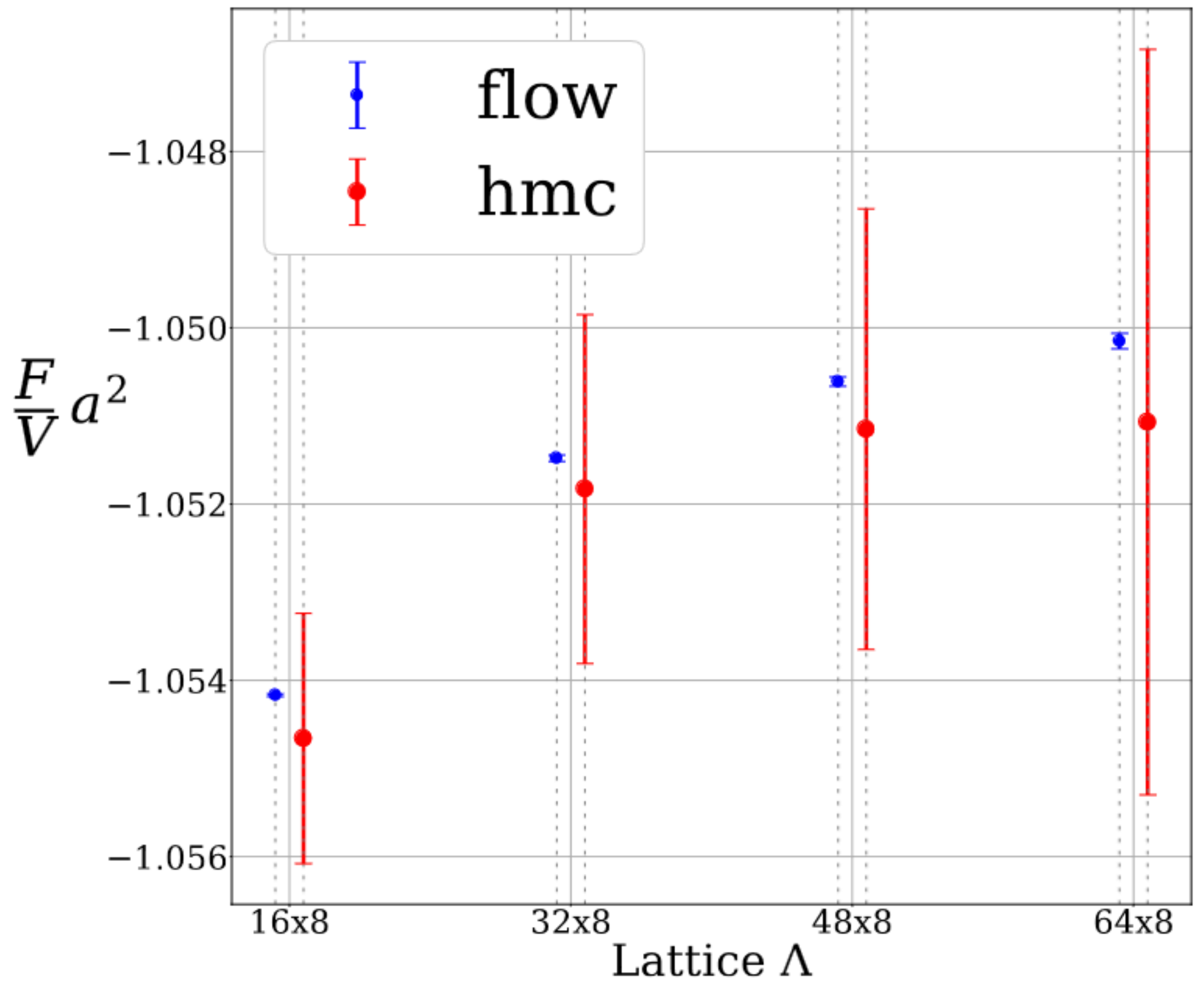
New paradigms

- Partition functions (e.g. for thermodynamics)
- Parameter dependence
  Gerdes+ (2022) 2207.00283
  Singha+ (2022) 2207.00980
- Correlated samples
- Transformed replica exchange
- Sign problems
  Lawrence+ PRD103 (2021) 114509
  Rodekamp+ PRB106 (2022) 125139
  Pawlowski & Urban (2022) 2203.01243

Practical gains

- Embarrassingly parallel sampling
- Storage-free ensembles

Nicoli+ PRE101 (2020) 023304
Nicoli+ PRL126 (2021) 032001




With $U_i \sim q(U)$,

$$\hat{Z} = \frac{1}{N}\sum_{i=1}^{N} e^{-S[U_i]}/q(U_i)$$

and $\hat{F} = -\log \hat{Z}$

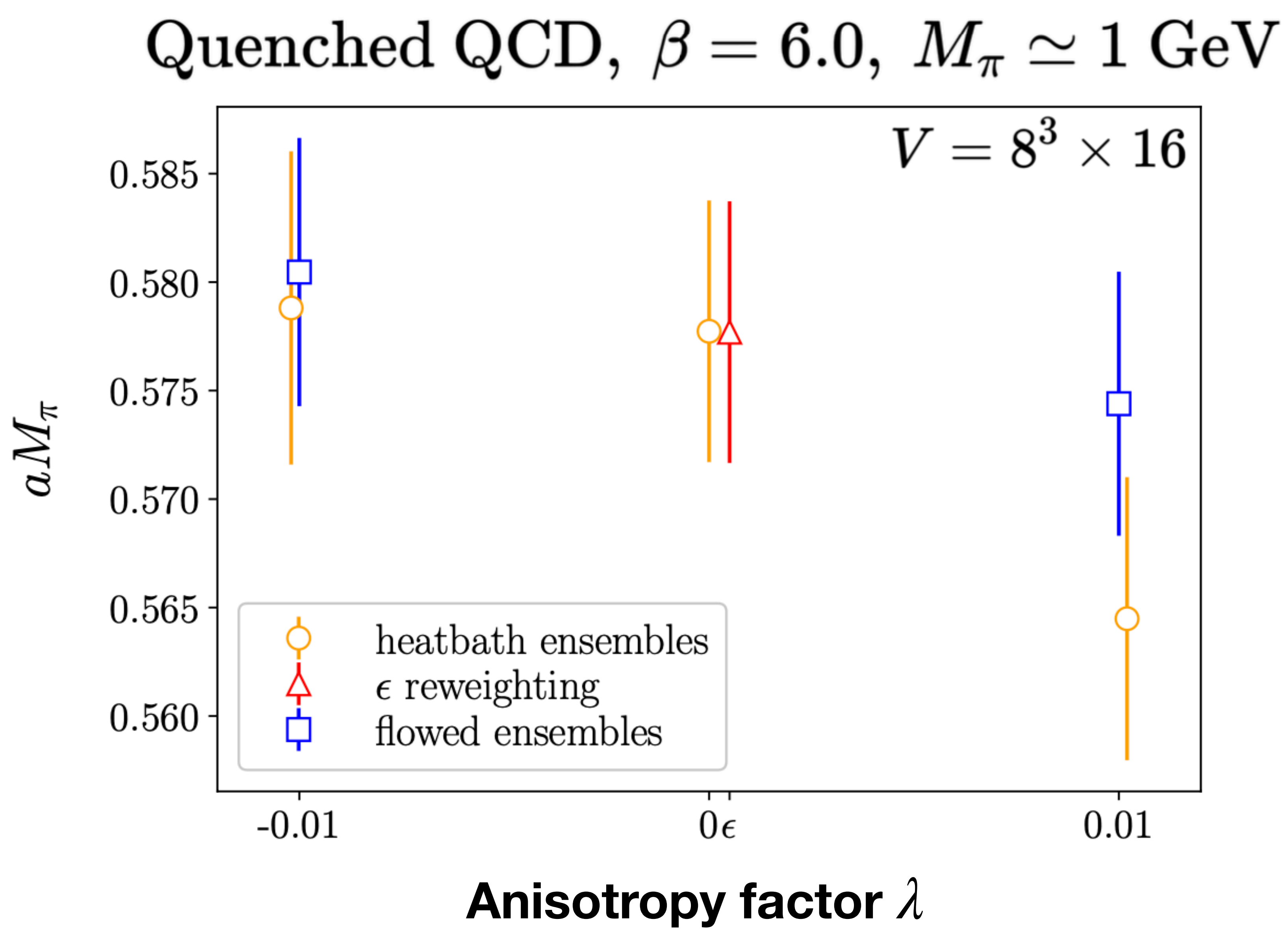
# Near-term applications

Correlated sampling **PRD109 (2024) 094514**
(e.g. Feynman-Hellmann)

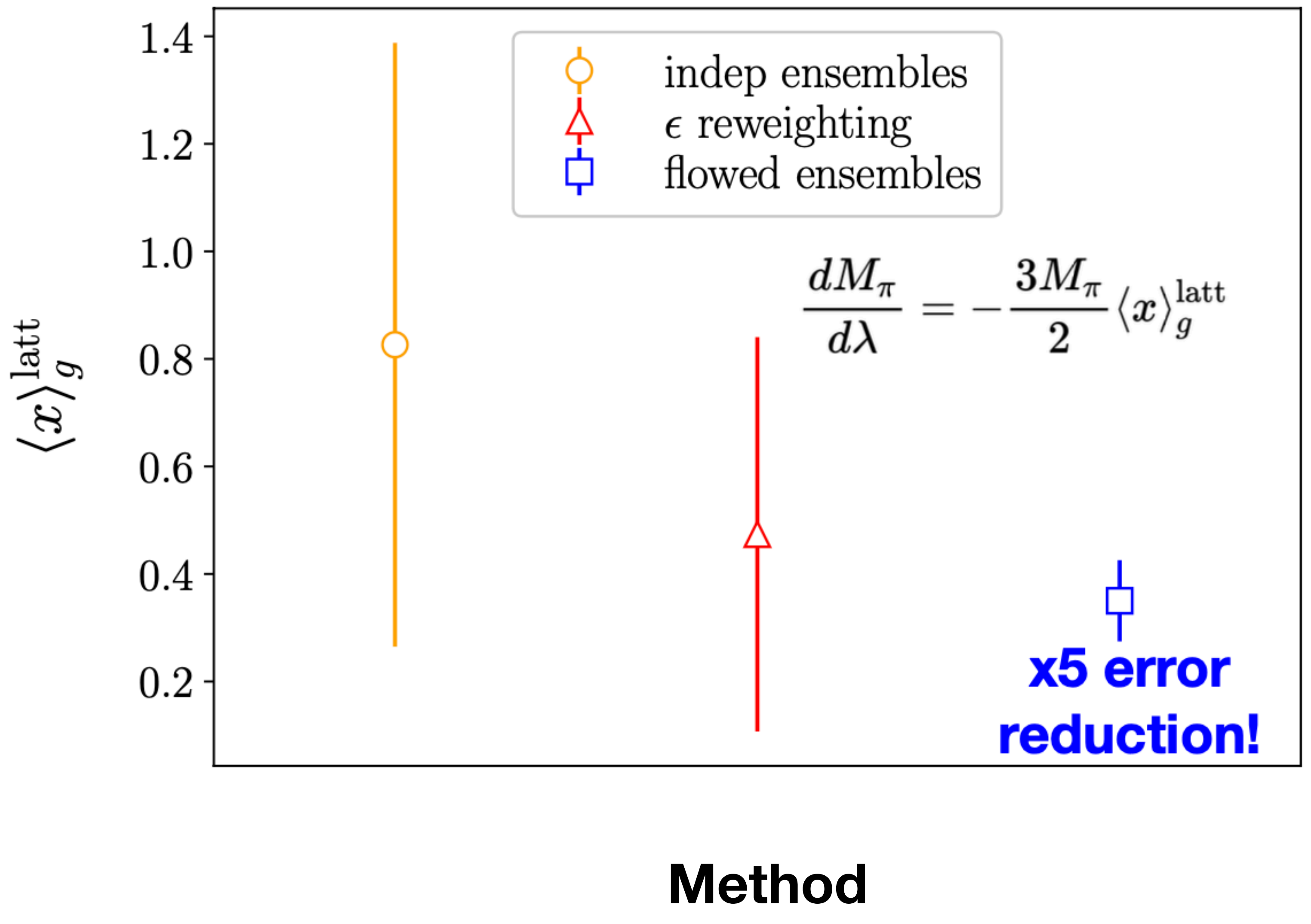
- "Shorter" distance to flow
- Correlations give noise reduction

Replica exchange with flows **2404.11674**

- "Shorter" distance to flow
- Flows can be easily inserted into existing PT procedures



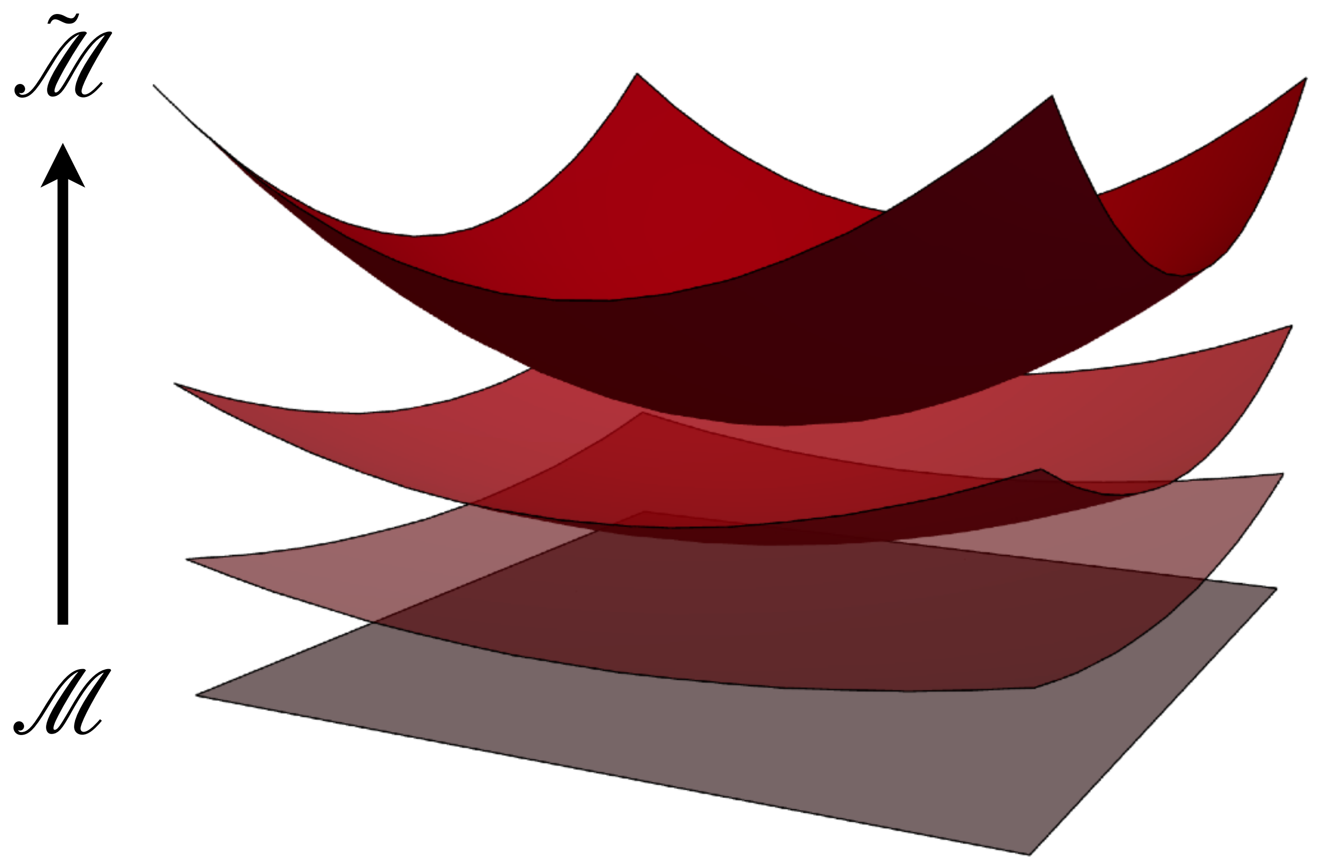

**Estimate derivative**



# Integral deformations for noisy observables

*Lattice integrands are often holomorphic, allowing the integration contour to be deformed without bias.*

$$\langle \mathcal{O} \rangle = \frac{1}{Z} \int_{\mathcal{M}} e^{-S(\phi)} \mathcal{O}(\phi) = \frac{1}{Z} \int_{\tilde{\mathcal{M}}} e^{-S(\tilde{\phi})} \mathcal{O}(\tilde{\phi})$$




- Defines a **modified observable**, which may have improved variance:

$$\mathcal{Q}(\phi) \equiv \det J(\phi) e^{-[S(\tilde{\phi}(\phi)) - S(\phi)]} \mathcal{O}(\tilde{\phi}(\phi))$$

$$\langle \mathcal{Q}(\phi) \rangle = \langle \mathcal{O}(\phi) \rangle$$

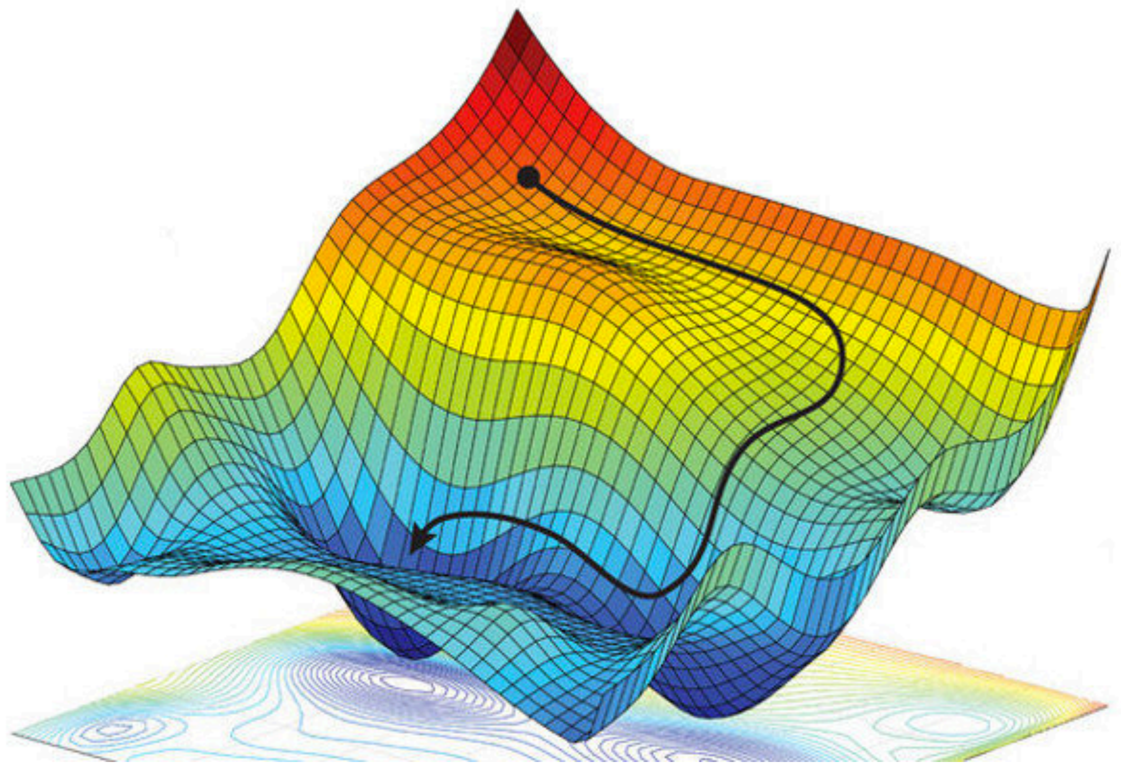
$$\mathrm{Var}[\mathcal{Q}(\phi)] \neq \mathrm{Var}[\mathcal{O}(\phi)]$$

Detmold, GK, Wagman, Warrington PRD102 (2020) 014514

[Image credit: Neill Warrington]

# Learning the integration contour

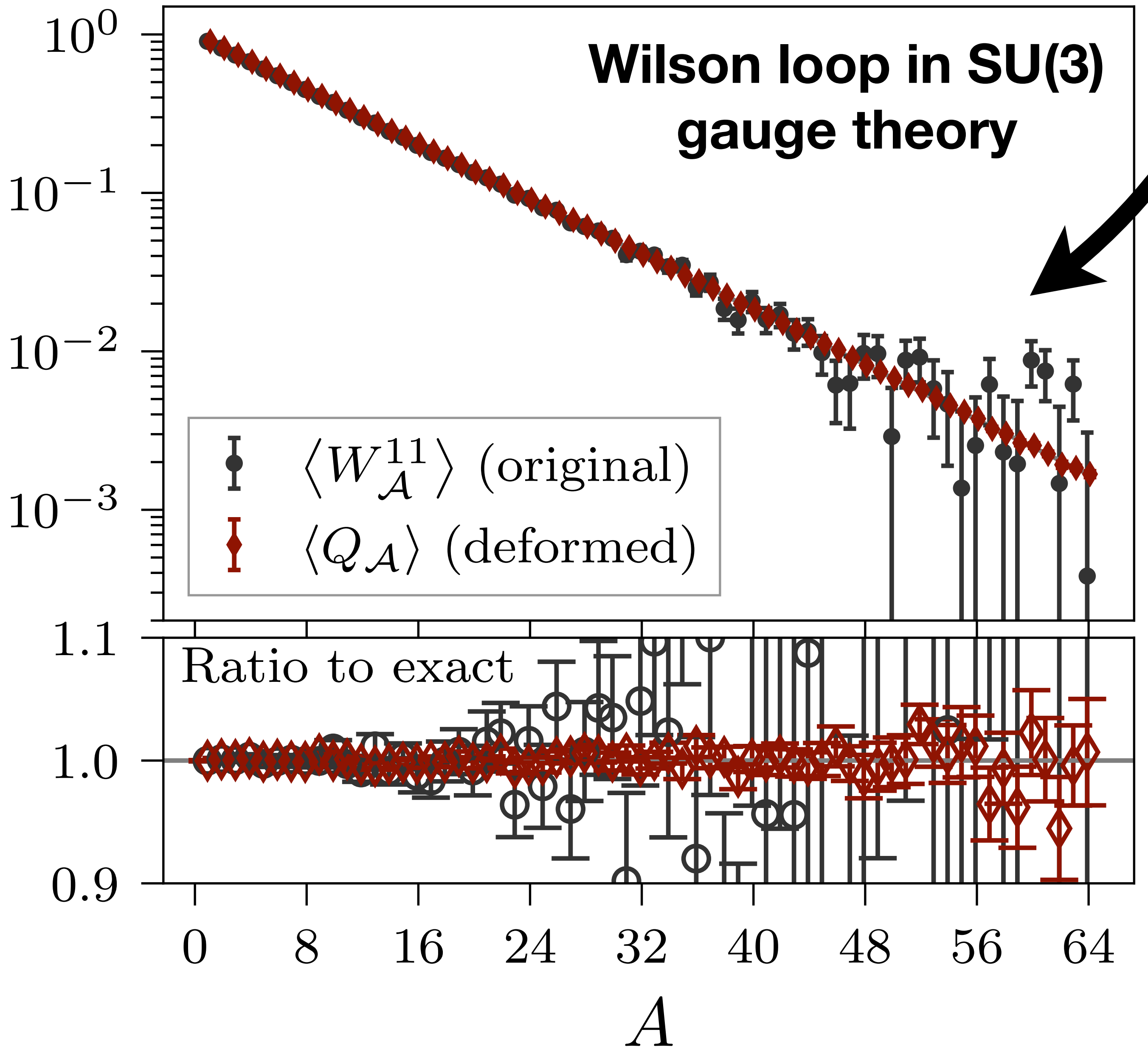
*The choice of $f : \phi \mapsto \tilde{\phi}$ defines $\tilde{\mathscr{M}}$, $\mathcal{Q}(\phi)$, and the variance.*

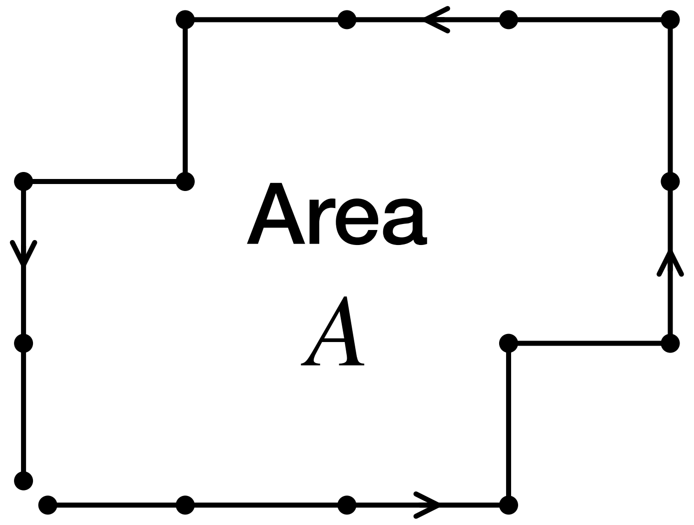


[Image credit: 1805.04829]

Parameterize $f(\phi; \omega)$ then **minimize variance**.

- Caveat: Complex analyticity
- Caveat: $SU(N)$ variables




Detmold, GK, Wagman, Warrington PRD102 (2020) 014514,
Detmold, GK, Lamm, Wagman, Warrington PRD103 (2021) 094517