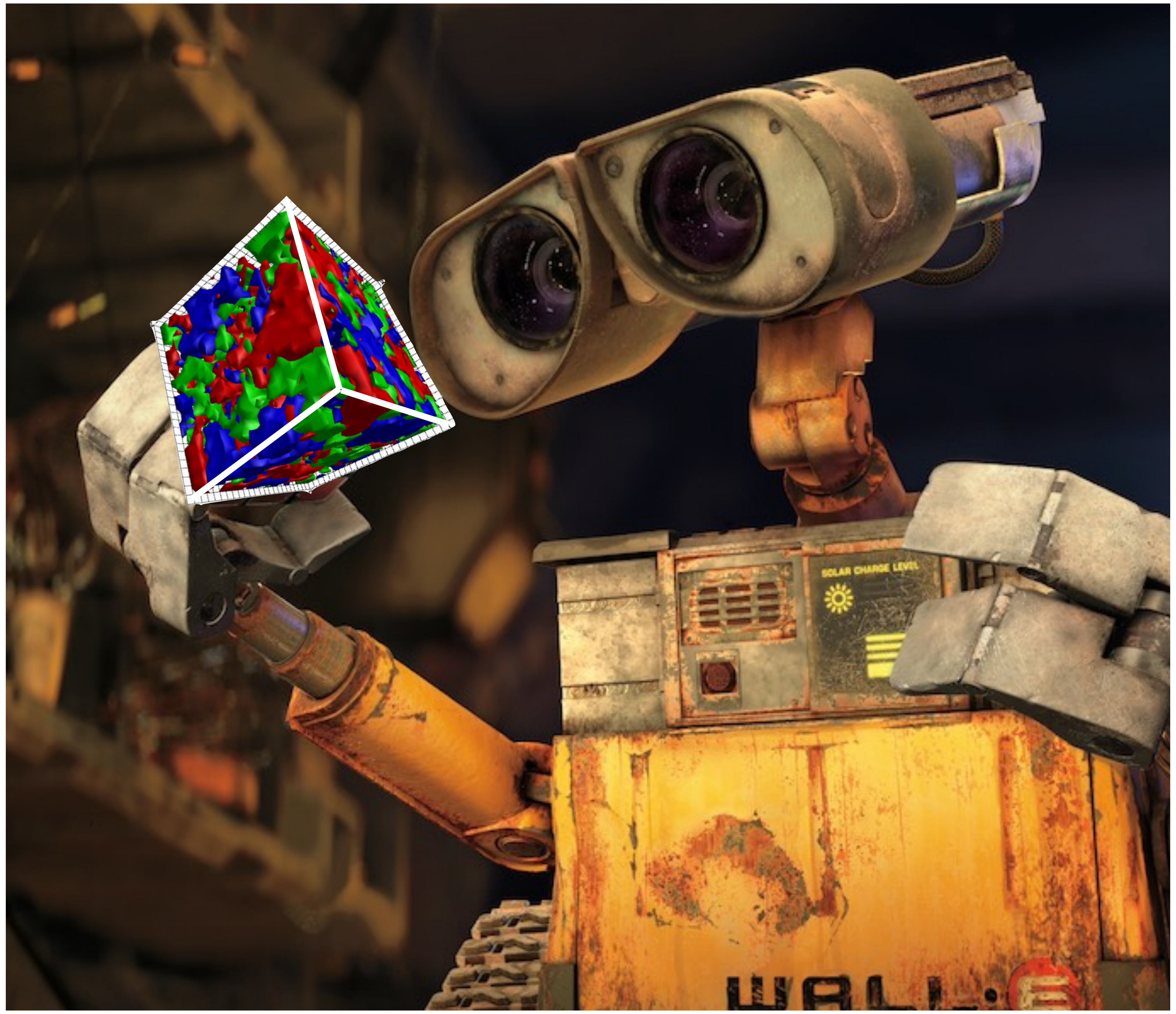
Machine Learning Methods

in

Lattice Gauge Theories



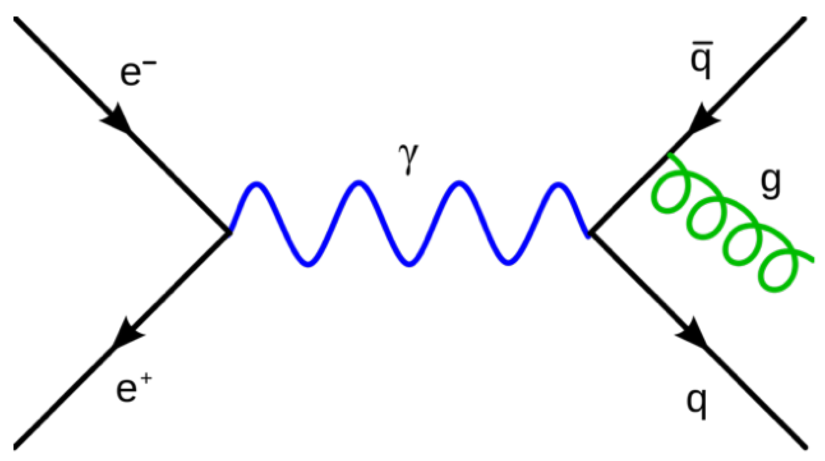
Stokes, Kamleh, Leinweber 1312.0991
WALL-E (2008) *Pixar, please don't sue me*

Gurtej Kanwar
Institute for Theoretical Physics, AEC, U. Bern

Dec 12-14, 2023
UK Annual Theory Meeting

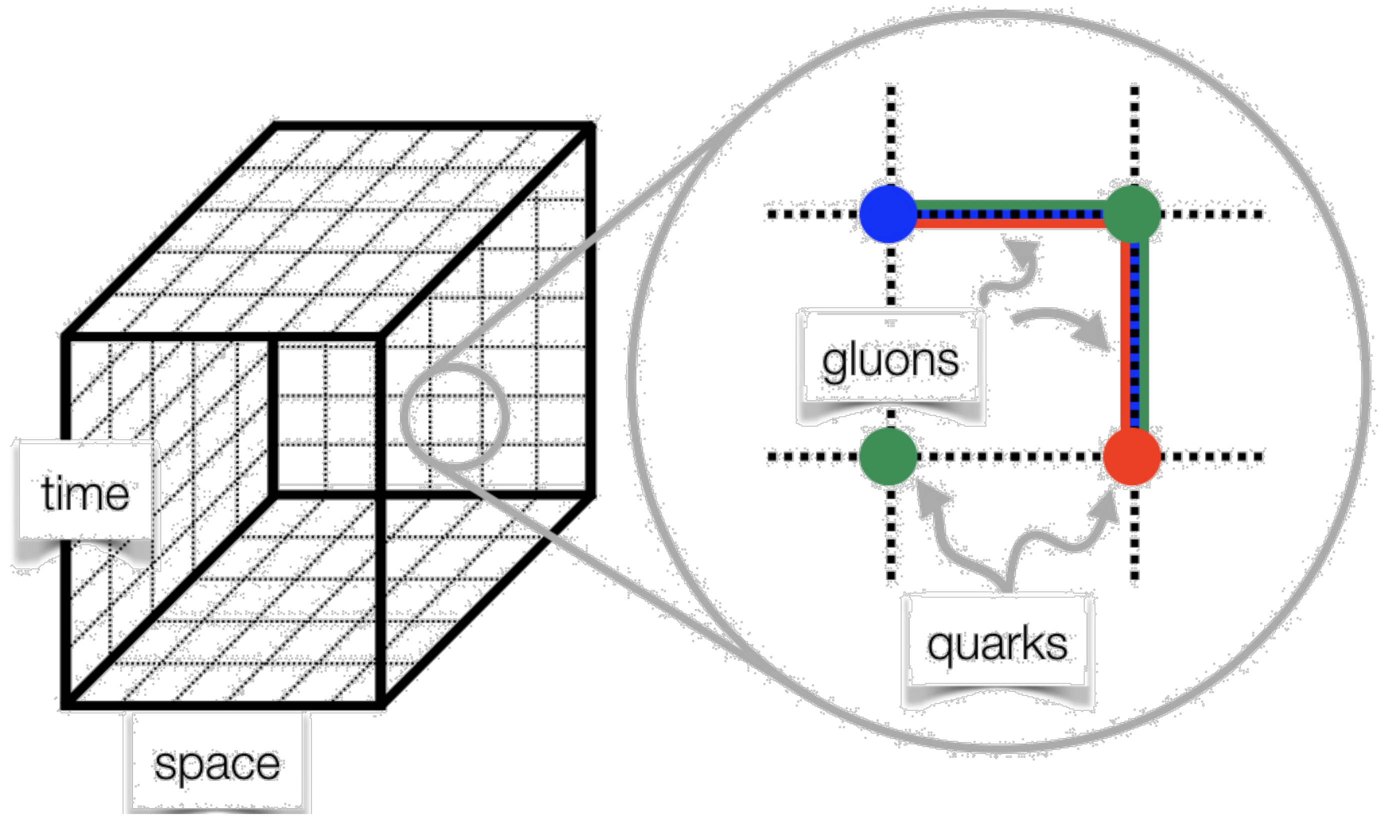
Lattices to regulate QFTs

- Electroweak effects and hard QCD processes can be treated **perturbatively**



- Low-energy QCD effects must be treated **non-perturbatively**

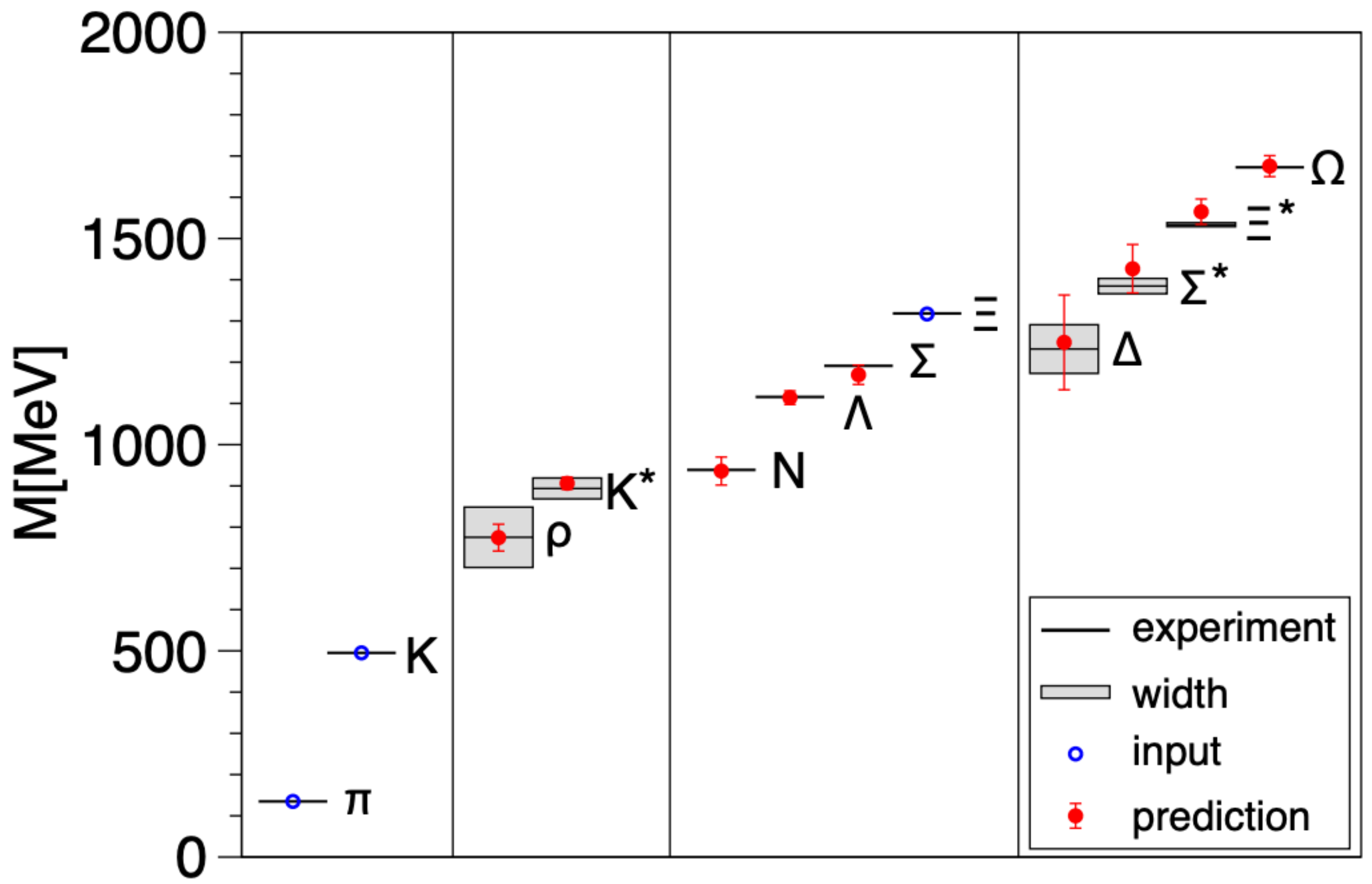
- Lattice field theory
  - Euclidean path integral on a spacetime lattice
  - Lattice spacing a cuts off UV divergences
  - Numerically evaluate, then extrapolate $a \to 0$



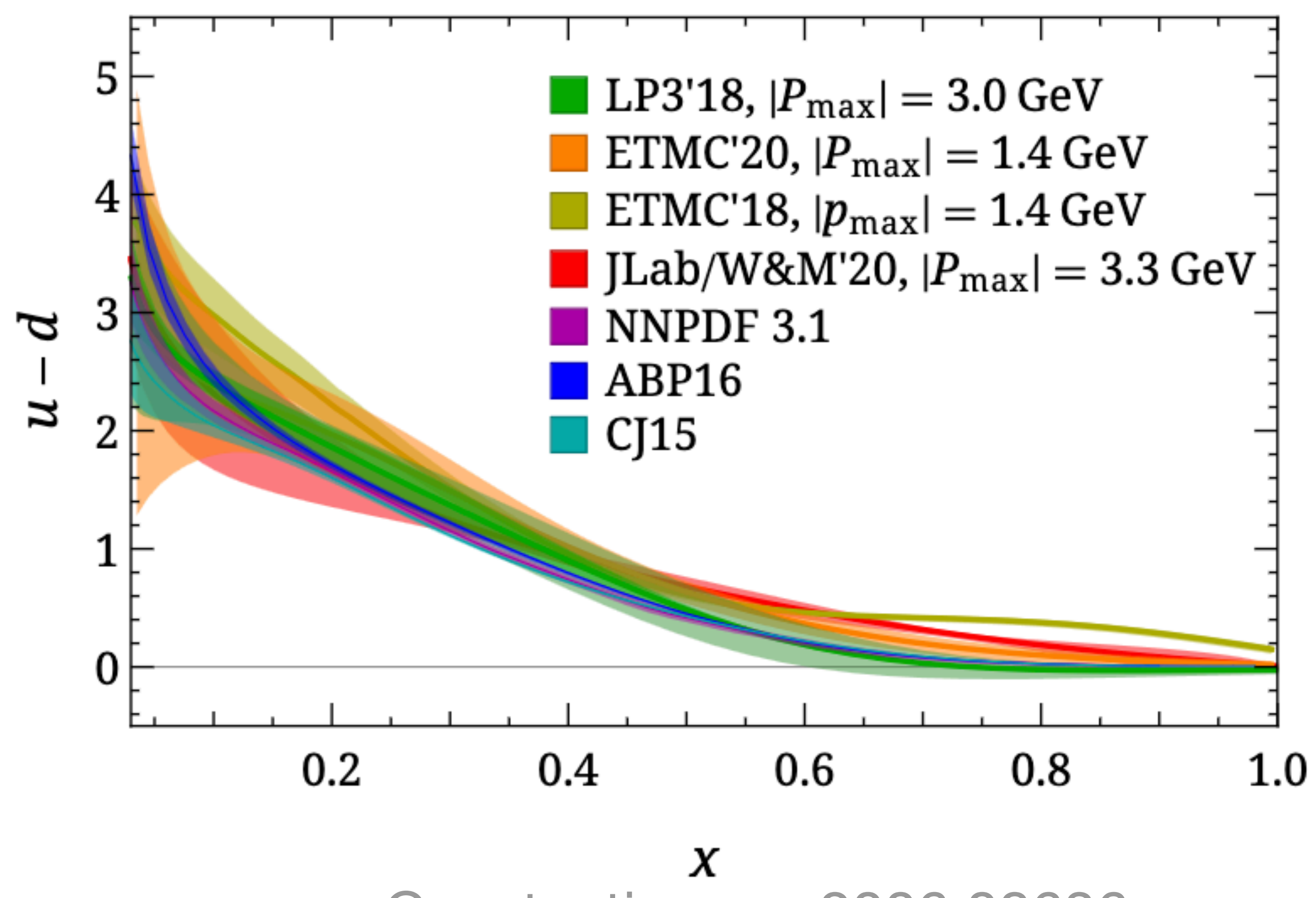
https://evanberkowitz.com/2018/05/30/gA.html

Lattice QCD

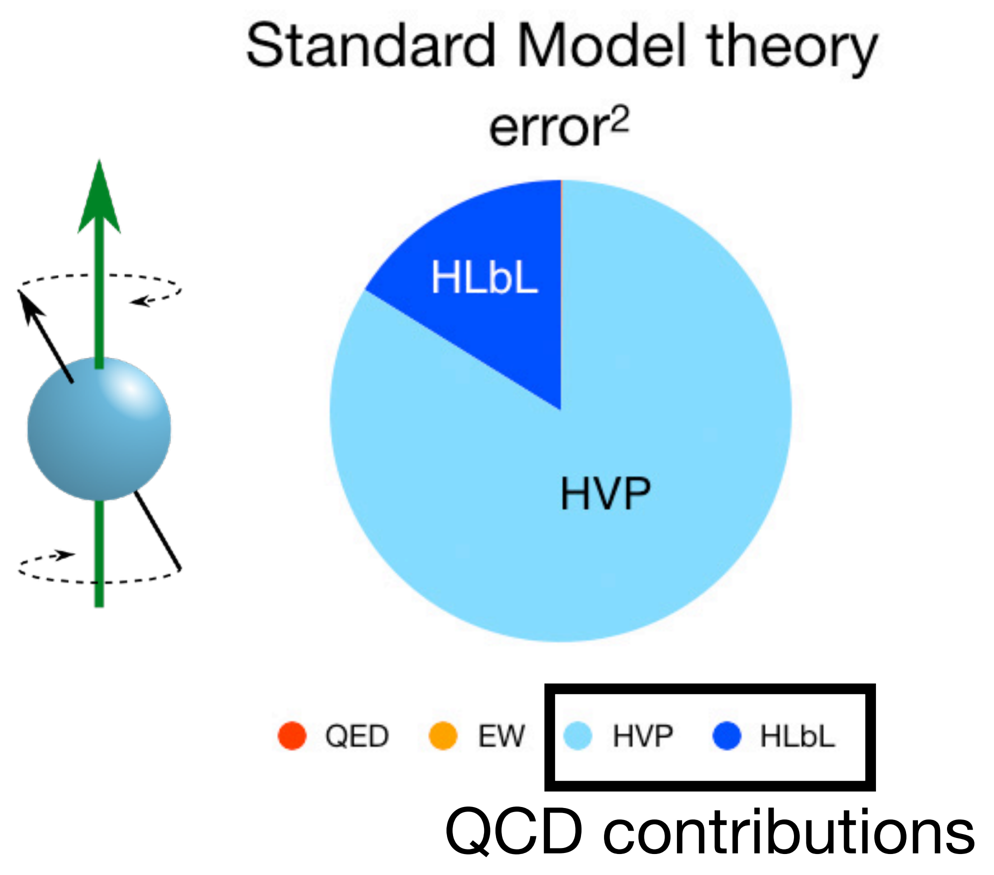
- Hadronic spectrum
  - Heavy resonances
- Hadronic structure
  - PDFs and their generalizations
  - Form factors
- New physics searches
  - Muon g-2
  - Heavy meson decays
- …



Fodor & Hoelbling RMP84 (2012) 449



Constantinou+ 2006.08636



Muon g-2 Press release (2023)

Lattice field theory

Discretized path integral:

- Degrees of freedom assigned to points and edges of a lattice

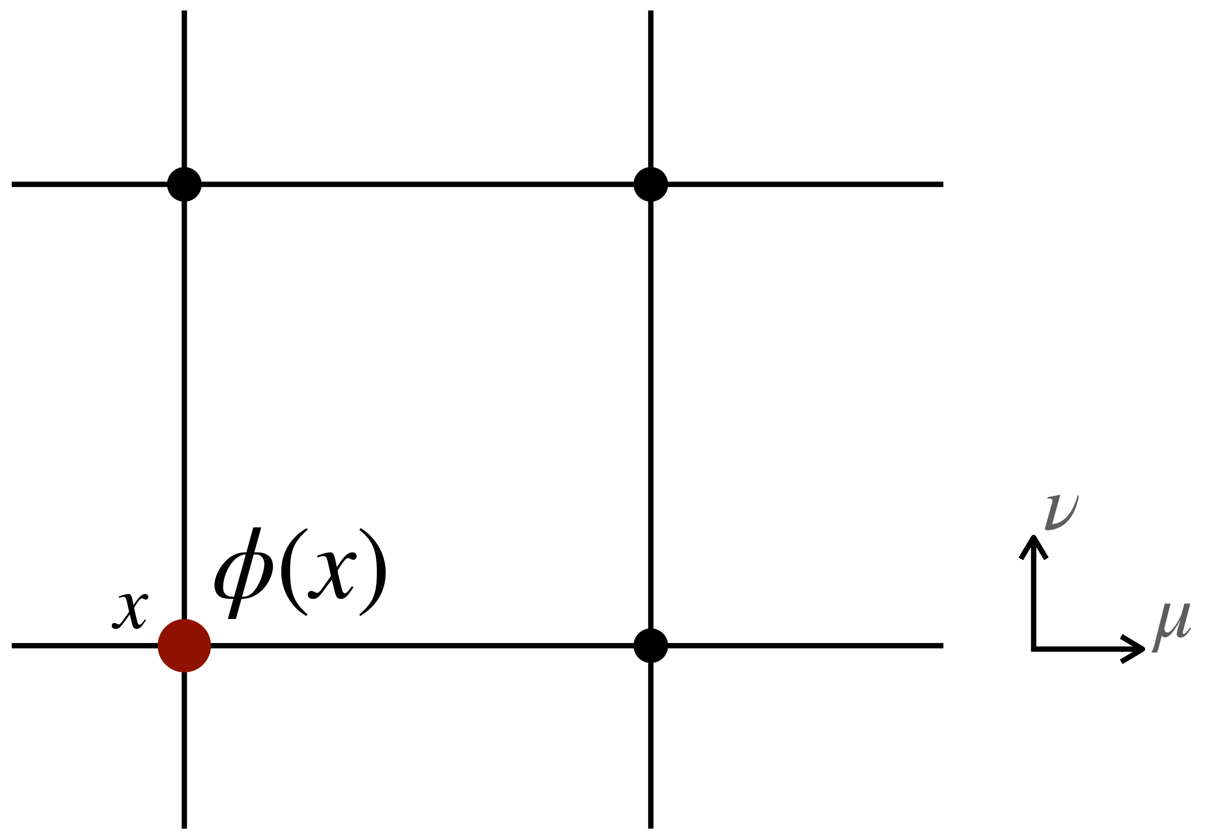
- Boltzmann weight $e^{-S(\phi)}$ encodes distribution over "typical" configurations

Partition function

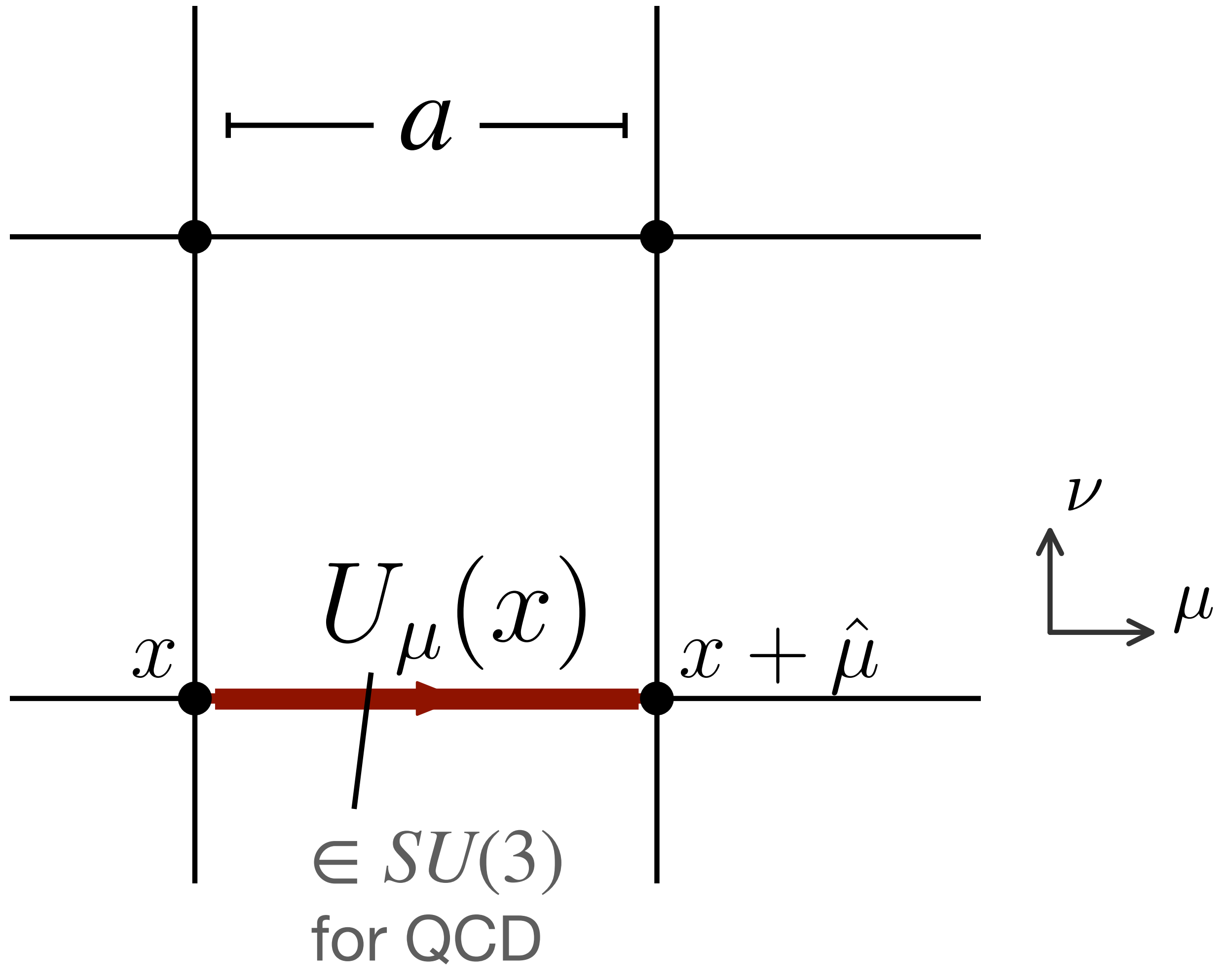
$$Z \equiv \left[\prod_x \int_{-\infty}^{\infty} d\phi(x)\right] e^{-S(\phi)}$$

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

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



Lattice scalar field configuration (2D slice)



Lattice gauge field configuration (2D slice)

Monte Carlo simulation

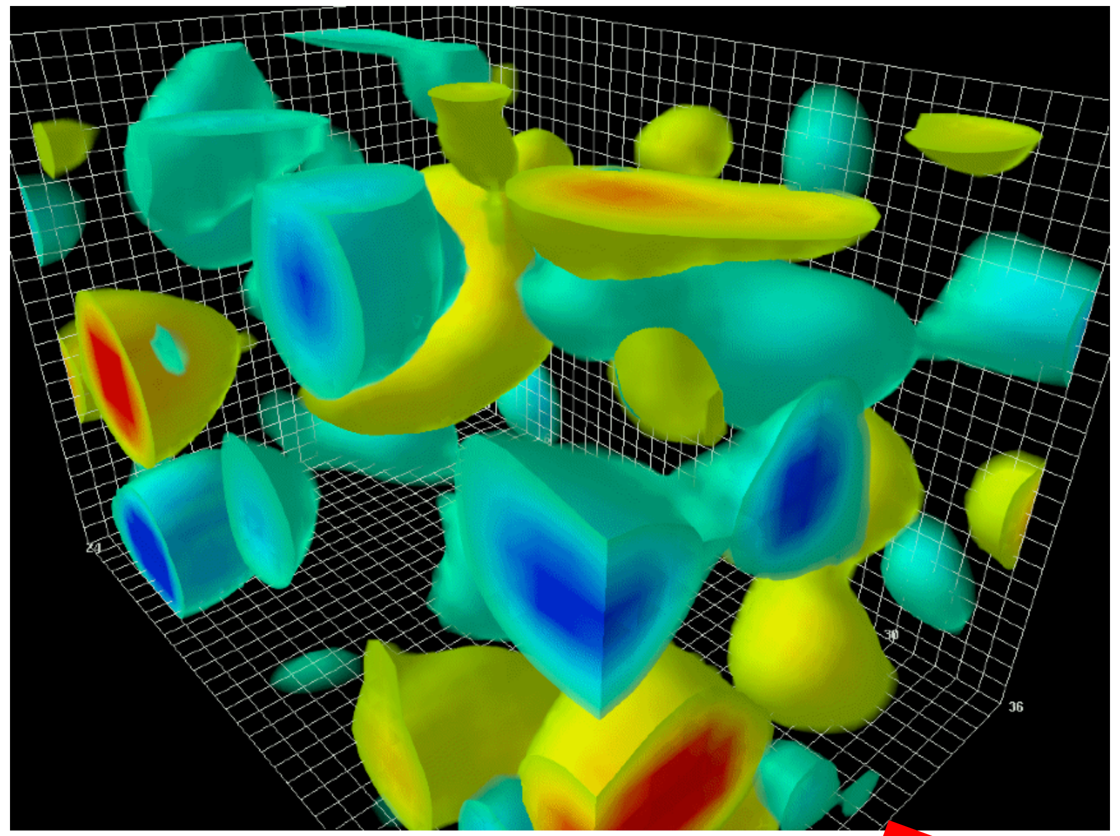
$$\langle \mathcal{O} \rangle = \frac{1}{Z} \Big[\prod_x \int_{-\infty}^{\infty} d\phi(x) \Big] \, \mathcal{O}(\phi) \, e^{-S(\phi)}$$

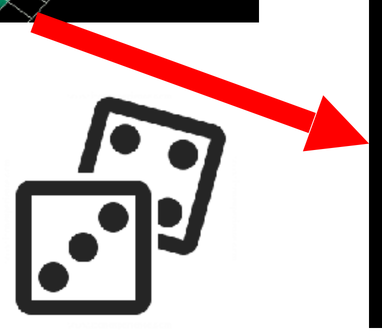
Approximate the path integral using **Markov chain Monte Carlo**

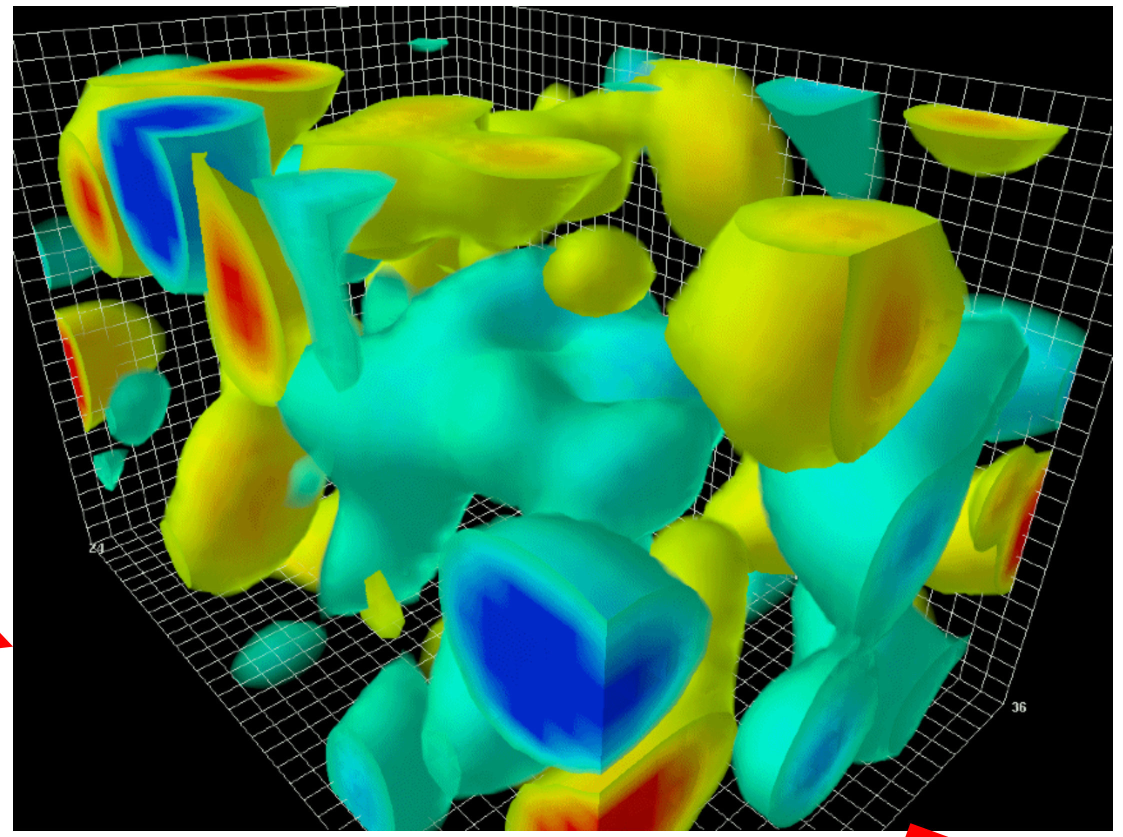
Positive integrand allows interpreting path integral weights as a probability measure:

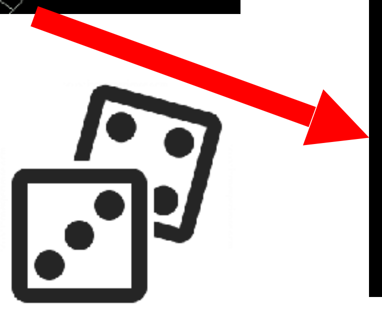
$$\phi_i \sim p(\phi) = e^{-S(\phi)}/Z$$

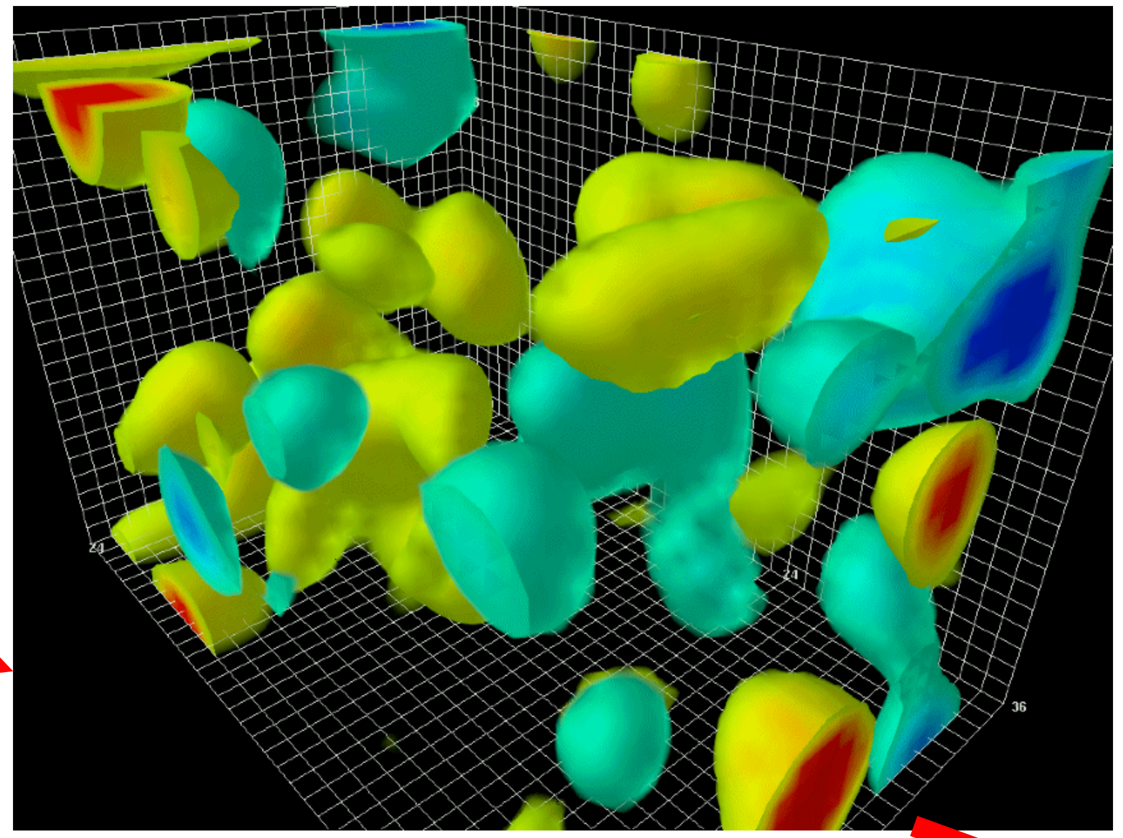
$$\langle \mathcal{O} \rangle \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{O}(\phi_i)$$

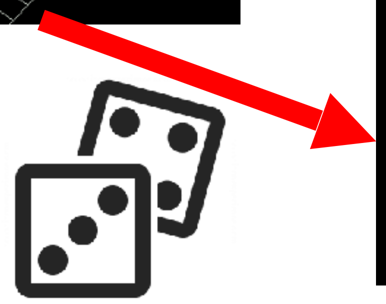


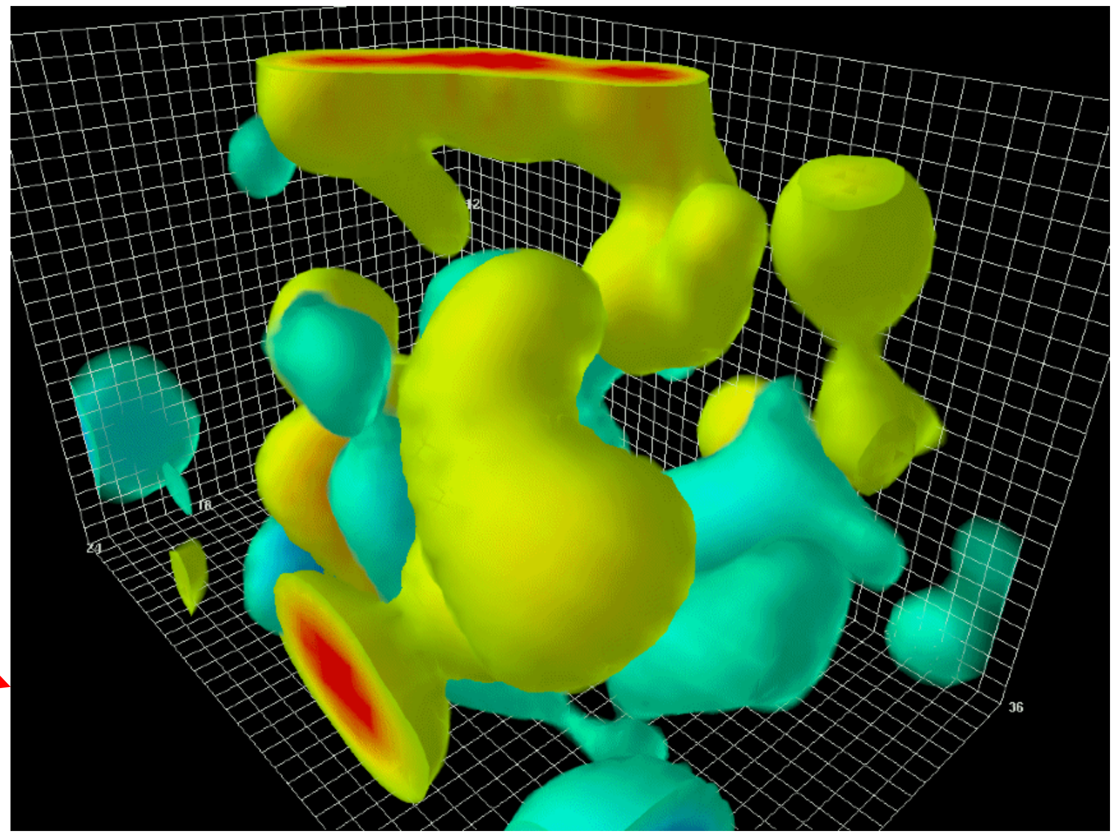












Leinweber, "Visualizations of Quantum Chromodynamics", 2004

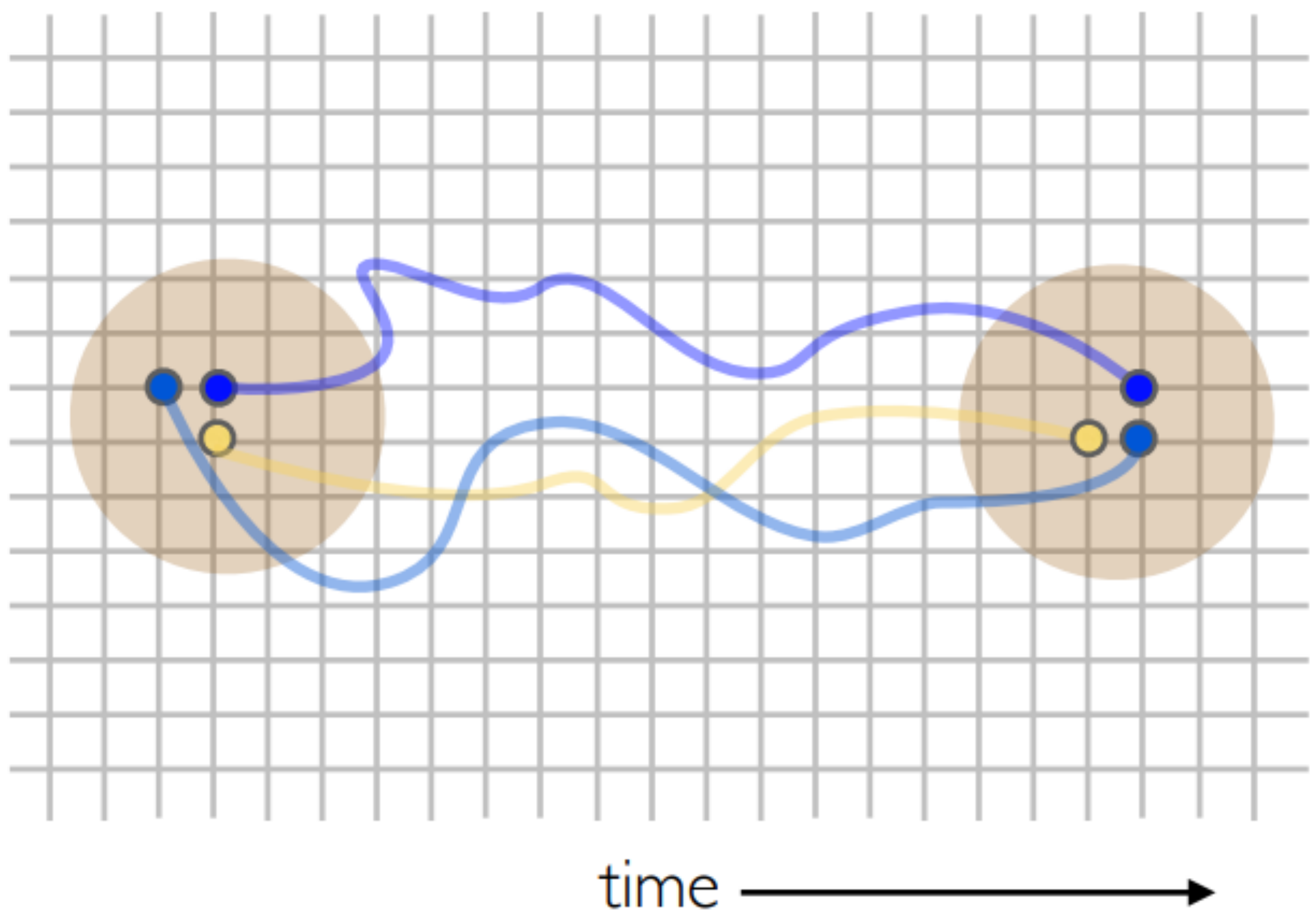
Measuring observables

Imaginary-time correlation functions inform us of the spectrum of the theory

$$\langle \mathscr{A}(t)\mathscr{A}^\dagger(0)\rangle = \sum_n Z_n e^{-E_n t} \xrightarrow{t \gg (\Delta E)^{-1}} Z_0\, e^{-E_0 t}$$

Operators designed to create/ annihilate state(s) of interest

Ground state energy (e.g. particle mass)



Nucleon correlator in lattice QCD

Matrix elements, form factors, etc. accessible via additional operator insertions.

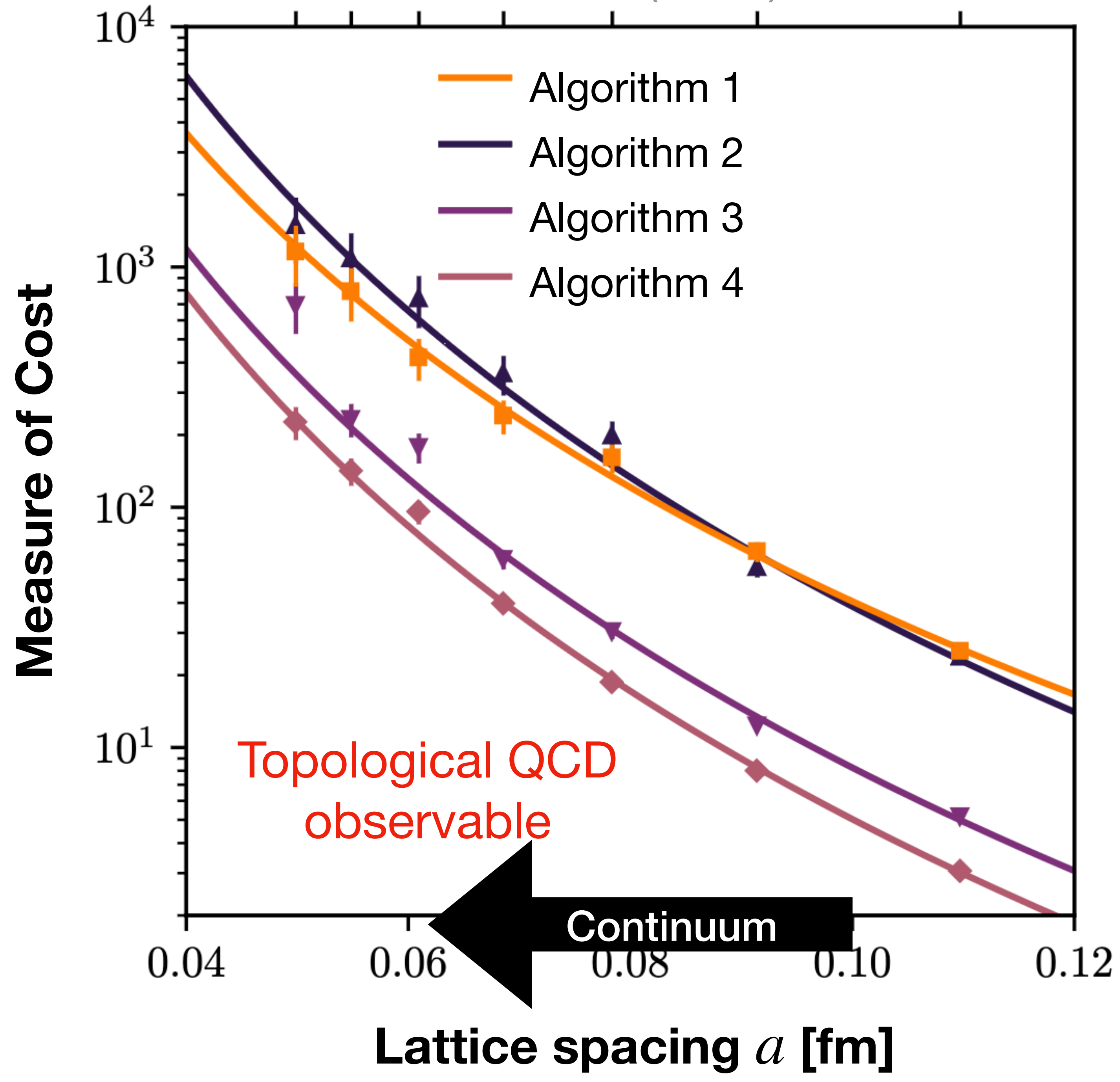
Why ML for Lattice (Gauge) Theories?

State-of-the-art LGT calculations require **enormous computational cost.**

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

These **limit the precision of physics results**
(e.g. in lattice QCD accumulated uncertainties from $a \to 0$, $m_\pi \to \sim 140\mathrm{MeV}$, and $V \to \infty$ limits!)

Eichhorn, et al. (2023) 2307.04742



Why ML for Lattice (Gauge) Theories?

Lattice field theories may be well-suited for application of ML

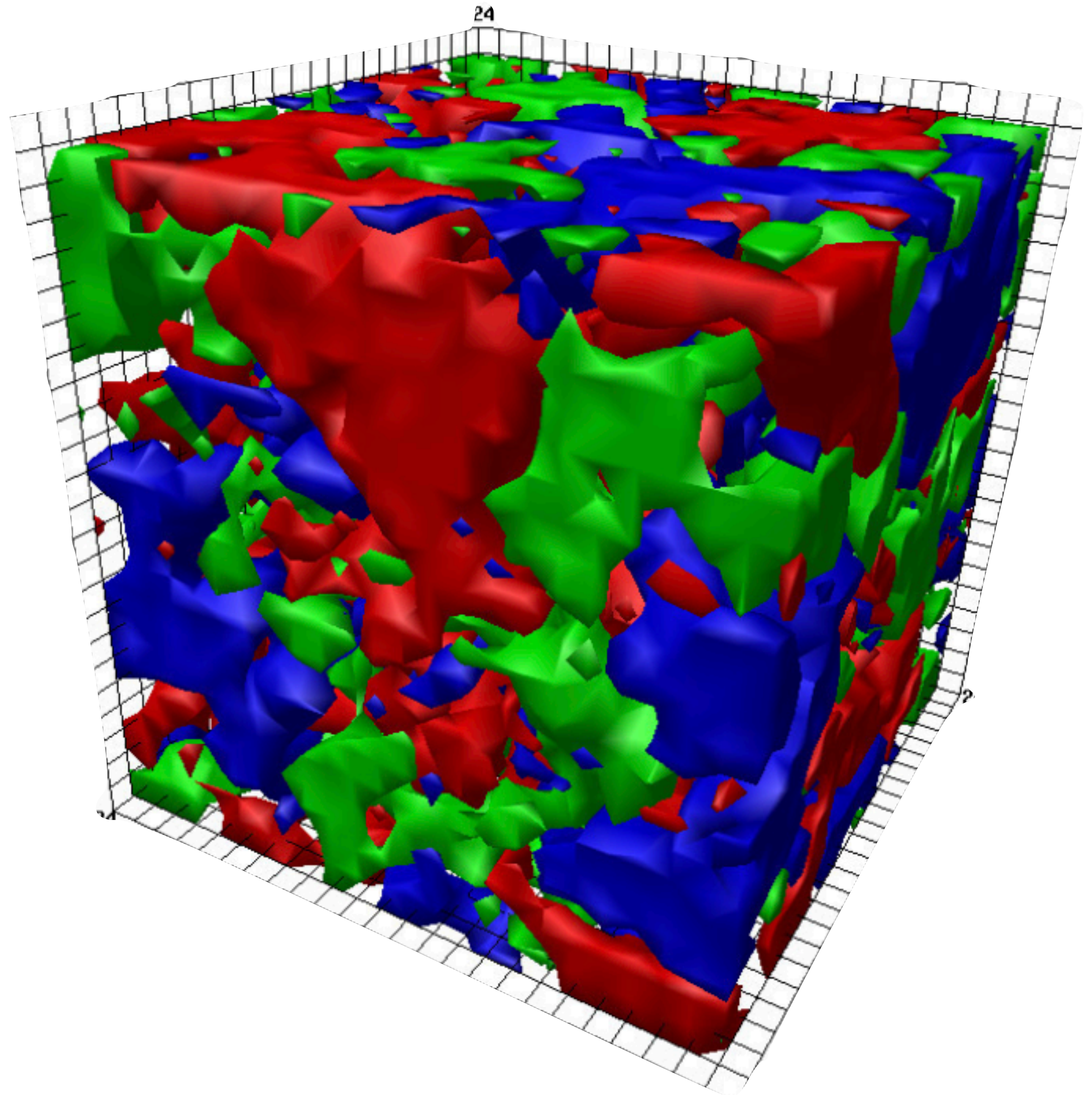
- Problem involving **lots** of well-structured data (lattice cfgs ~ images)

- Analytically-known Boltzmann distribution

- Flexibility to choose interpolating operators

- Flexibility to make model choices during analysis

- Ill-posed inverse problems



Stokes, Kamleh, Leinweber 1312.0991

Why ML for Lattice (Gauge) Theories?

Two major components to a lattice calculation.
Might be interesting in applying ML to any/all of these.

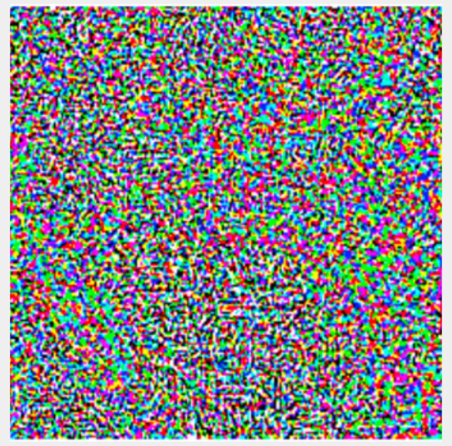
See e.g. Boyda, et al. Snowmass 2022, 2202.05838

1. Ensemble generation
2. Observable measurements & analysis

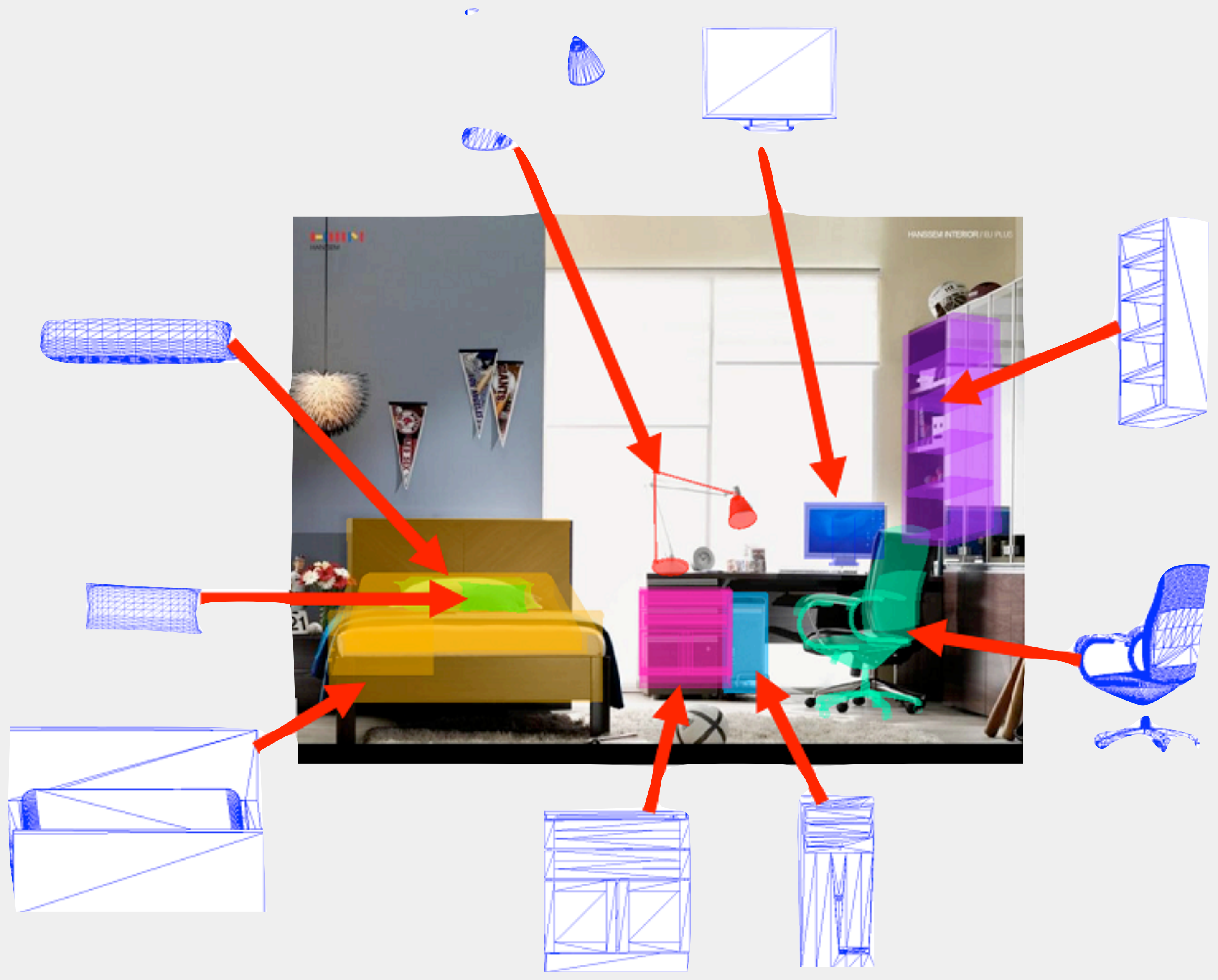
[Karras, Lane, Aila / NVIDIA 1812.04948]

few typical "configurations"

many atypical "configurations"



Not real people!



[github.com/timzhang642/3D-Machine-Learning]

Introduction to machine learning methods

What is machine learning?

Neural networks

+

Stochastic gradient descent
Backpropagation

+

Large training datasets

+

…

Methods

Image classification

+

Language processing

+

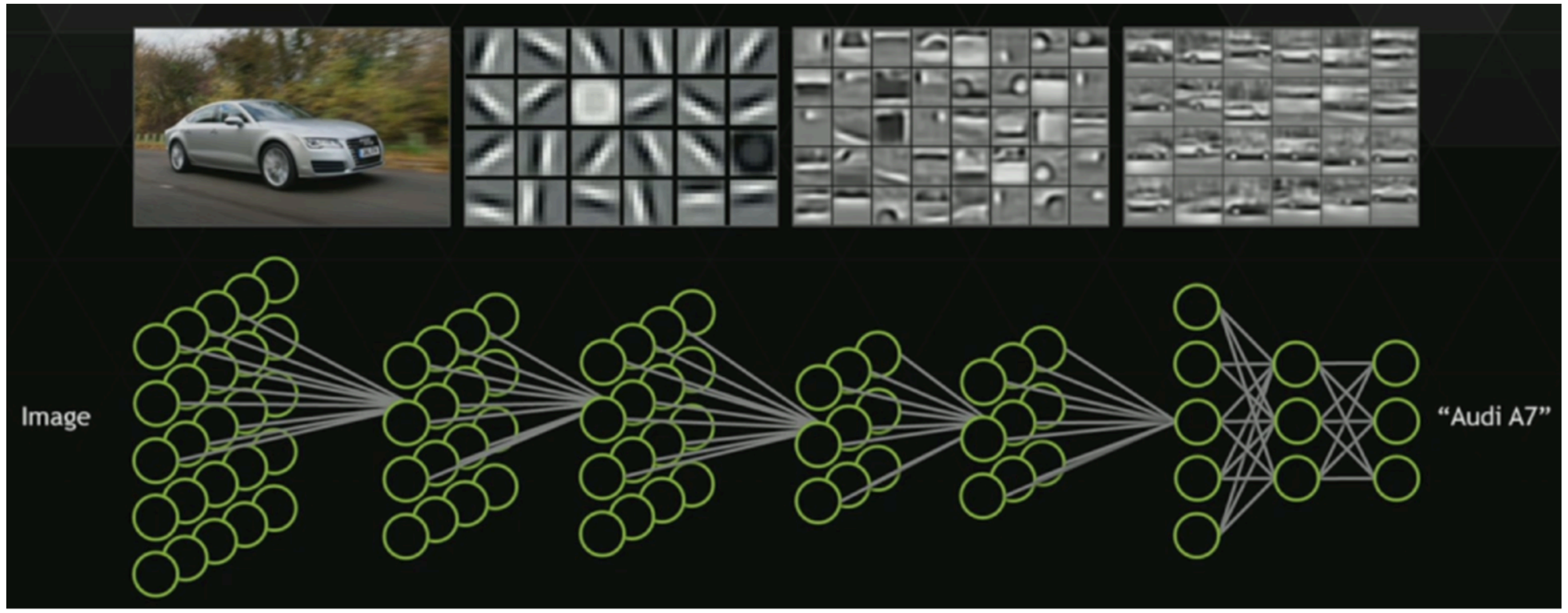
Generative models

+

…

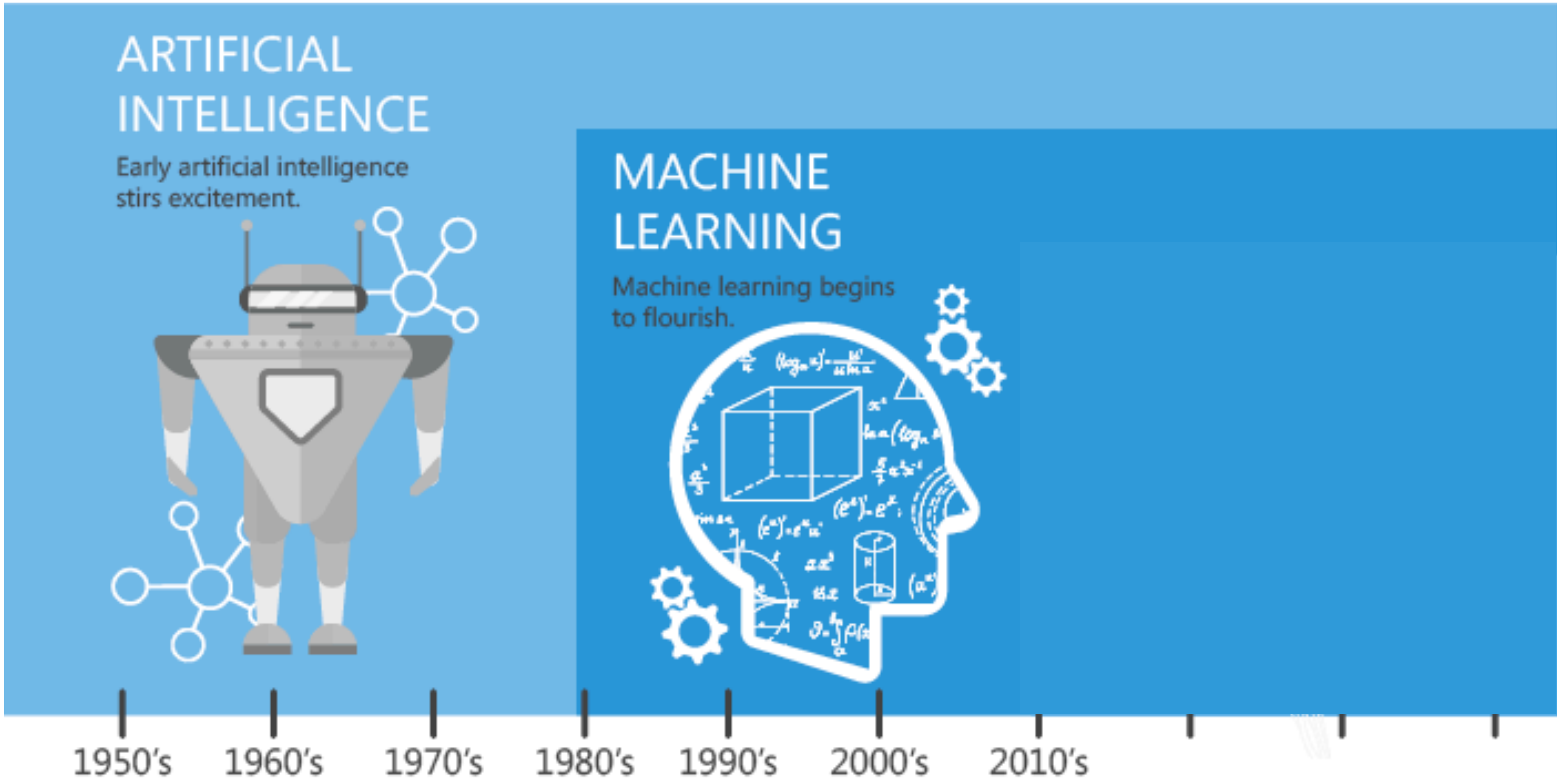
Applications

Artificial intelligence vs. machine learning



Pixar, WALL-E (2008)
NVIDIA CEO Jen-Hsun Huang, "Visual Computing: The Road Ahead" (2015)

Artificial intelligence vs. machine learning



https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08

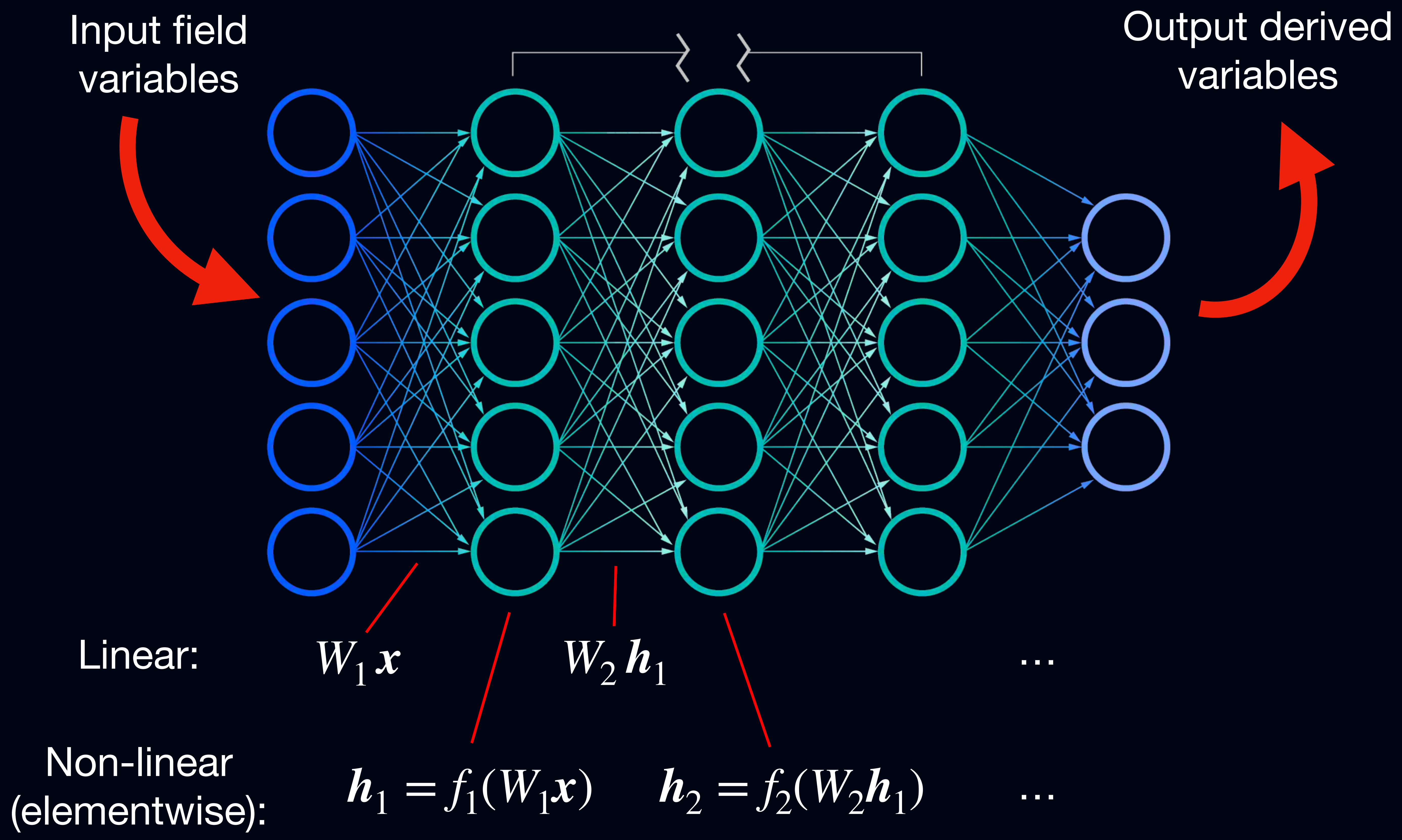
Neural networks or: How I Learned to Stop Worrying and Love the Black Box

Parametrized **linear transforms** + elementwise **non-linear functions**

→ Universal function approximators

K. Hornik, Neural Networks 4, 251–257 (1991)

- Matrices of weights W_1, W_2 are the (optimizable) model parameters ω
- Convolutional neural networks particularly useful on the lattice



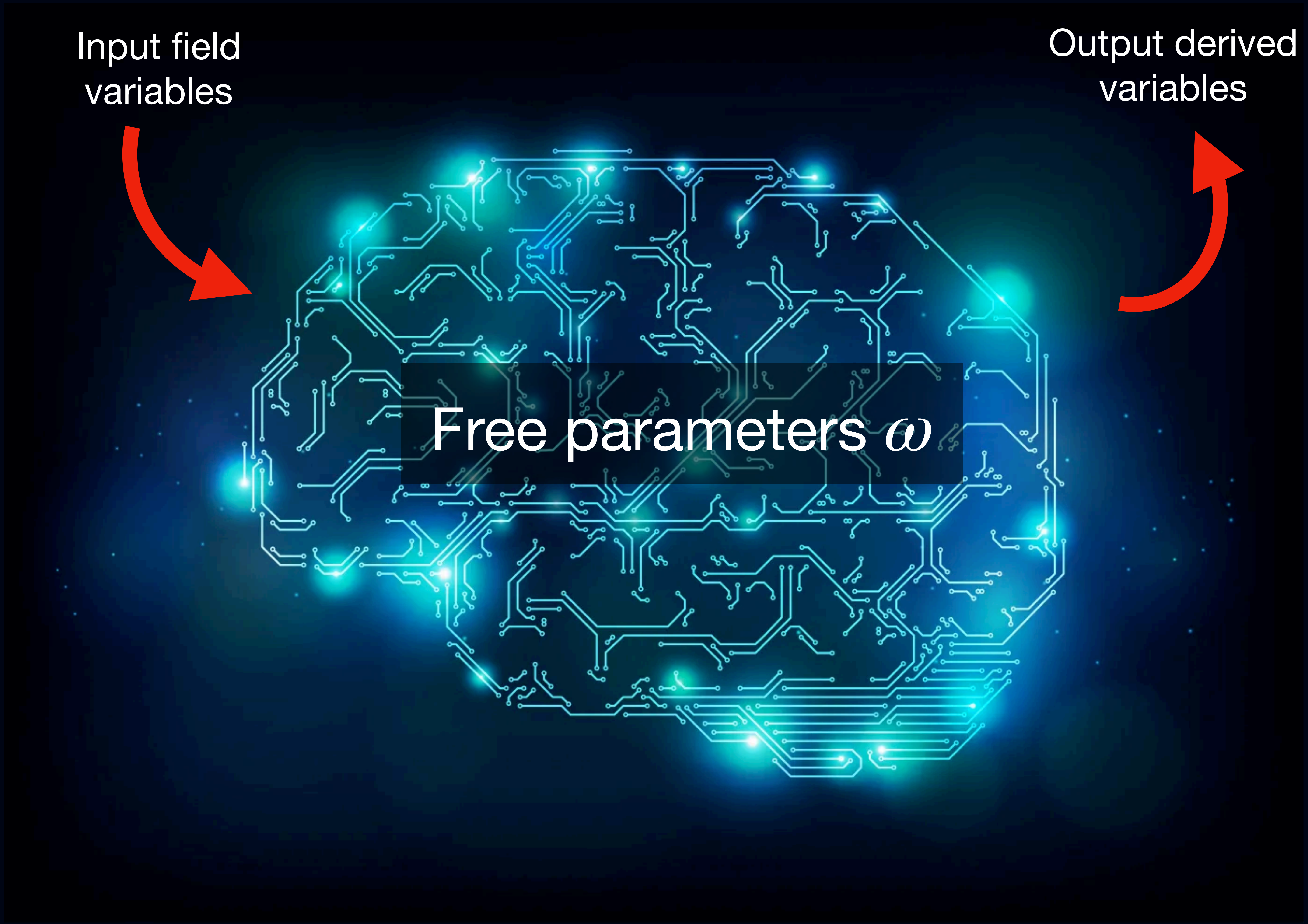
Neural networks or: How I Learned to Stop Worrying and Love the Black Box

Parametrized **linear transforms** + elementwise **non-linear functions**

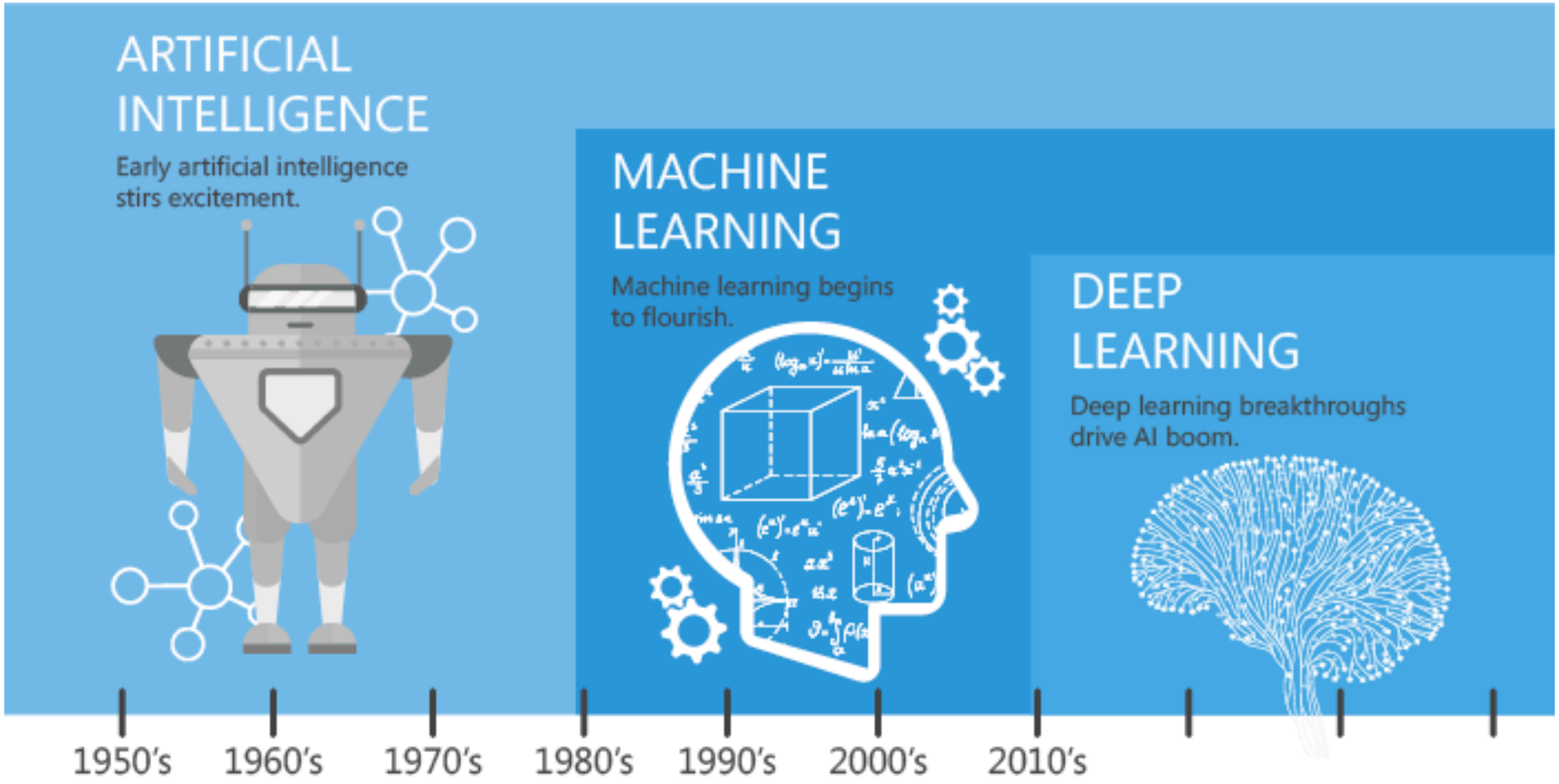
→ Universal function approximators

K. Hornik, Neural Networks 4, 251–257 (1991)

- Matrices of weights W_1, W_2 are the (optimizable) model parameters ω
- Convolutional neural networks particularly useful on the lattice

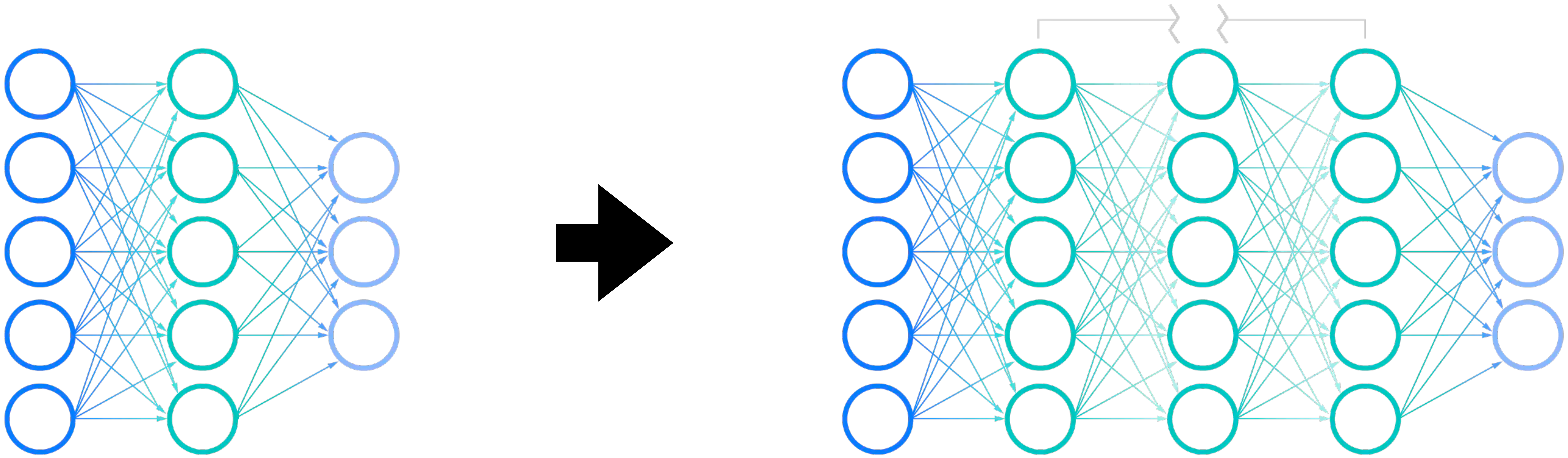


We need to go deeper



https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08

"Deep learning" = many layers



May be able to express more complex functions with fewer nodes per layer.

Could be harder to train.

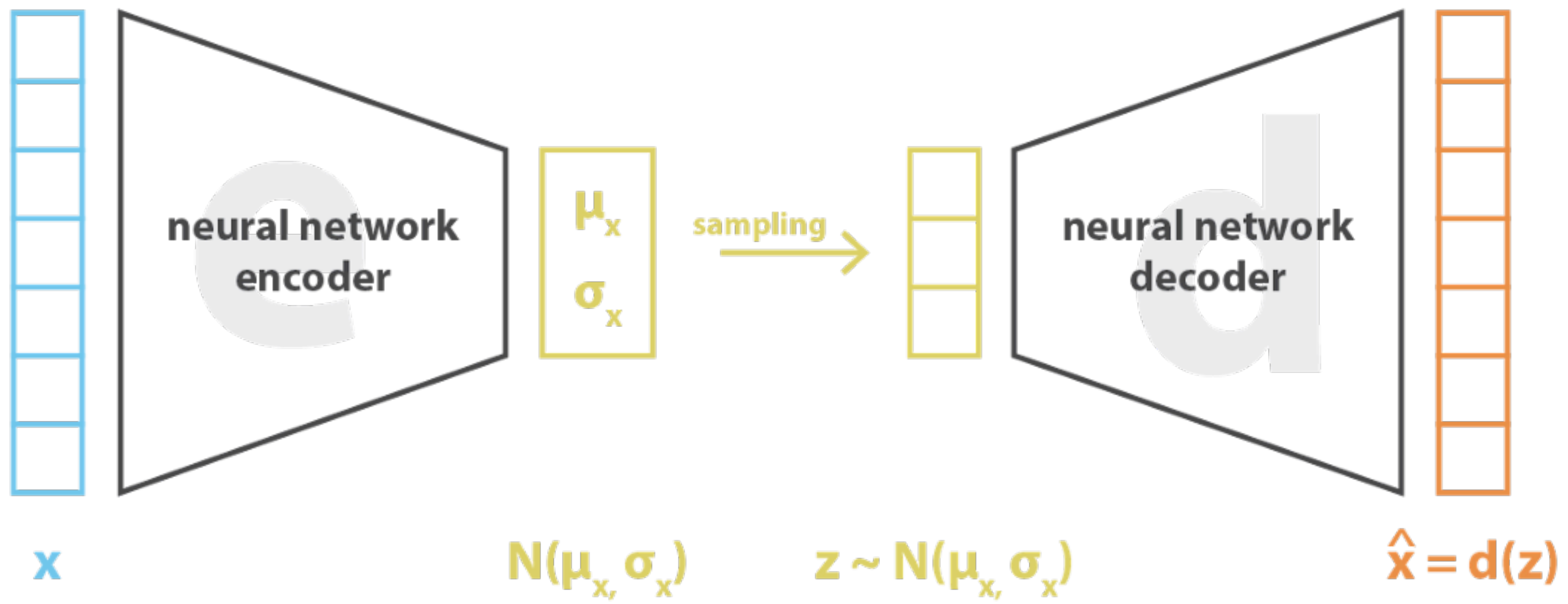
Lattice applications: Unclear whether lessons from standard ML apply. Try things!

https://www.ibm.com/cloud/learn/neural-networks

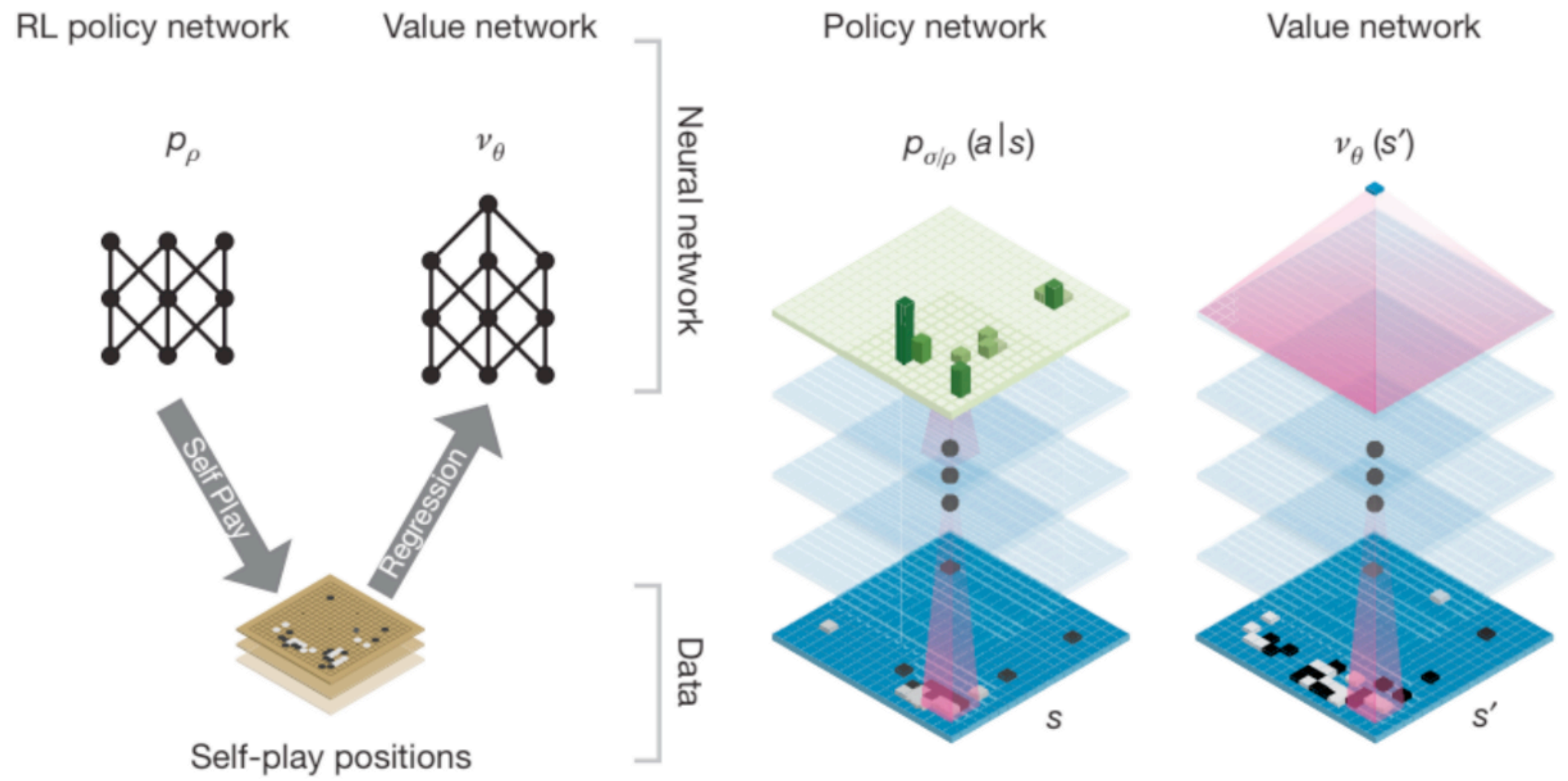
More general machine learning models

Composition of NNs with various **stochastic** or **deterministic** operations.

- Generative models
- Reinforcement learning
- …



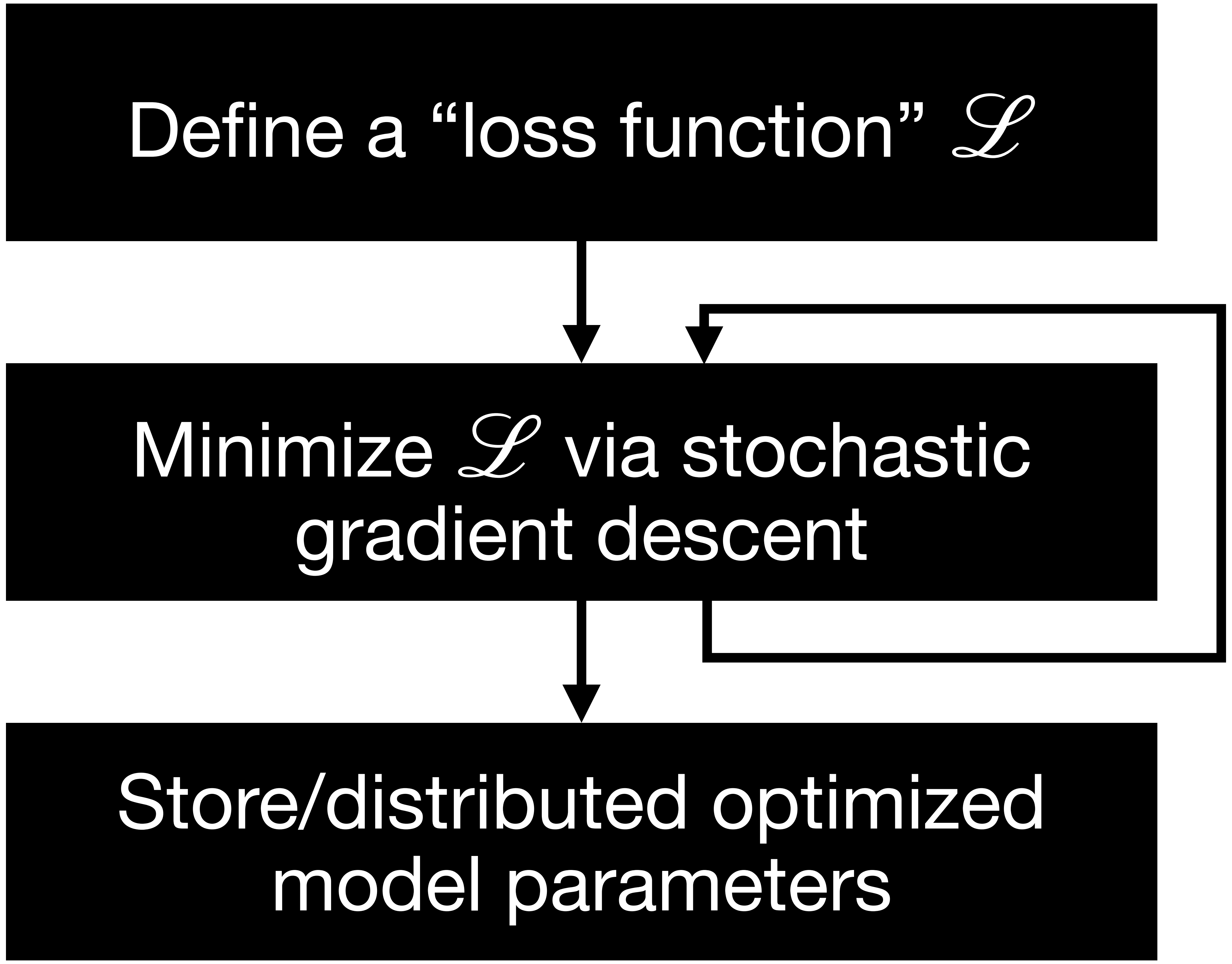
Variational Autoencoder (VAE)



AlphaGo RL agent

Silver+ (DeepMind) Nature 529, 484–489 (2016)
https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

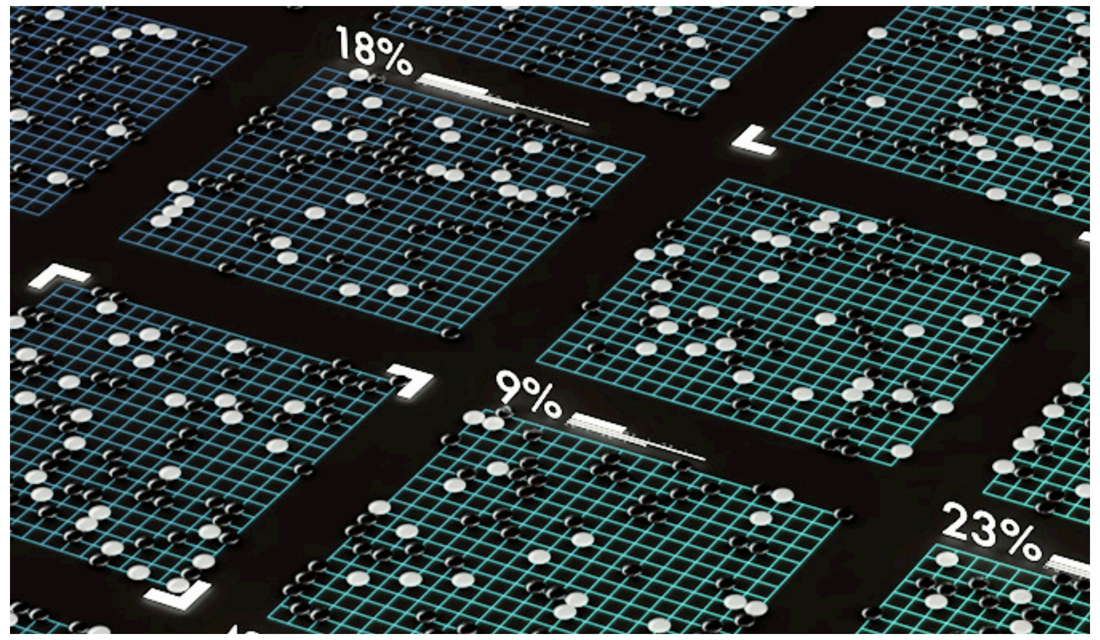
Optimizing the models



Supervised and unsupervised learning

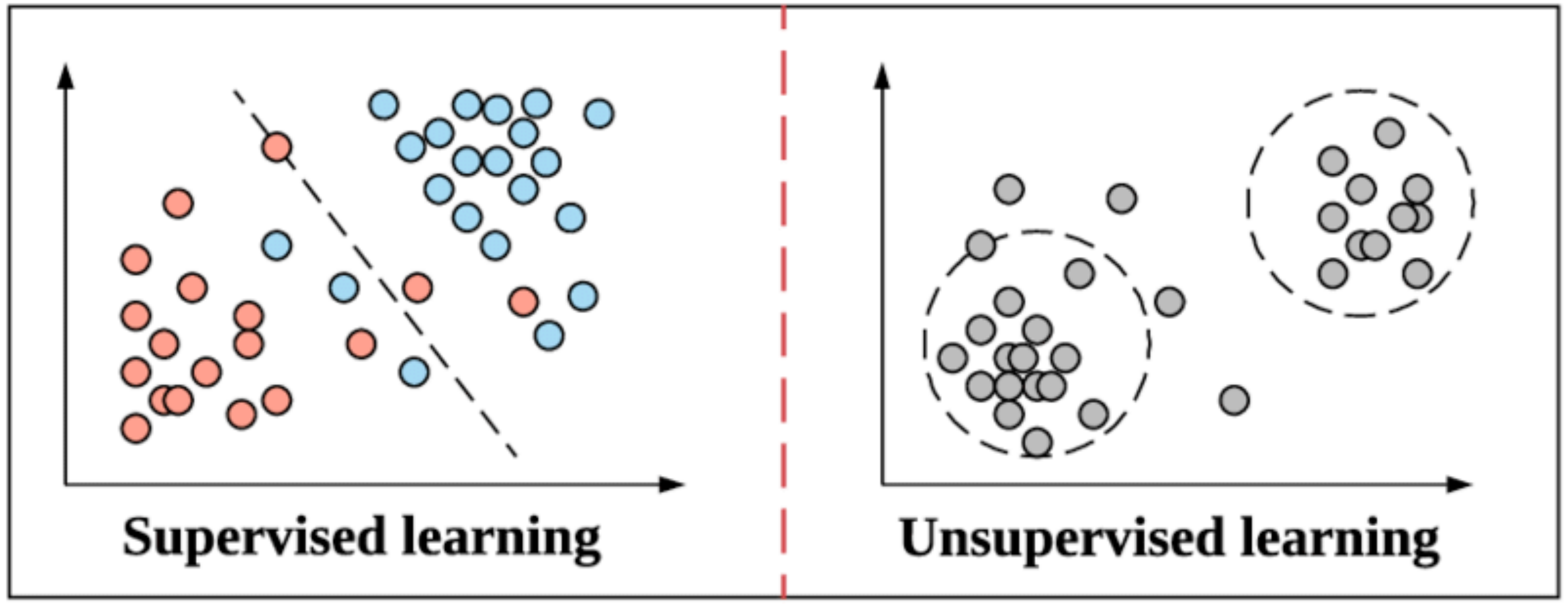
Supervised: "ground truth" training data available

- Images with human-identified labels
- "Go" game positions with heuristic strength values



Unsupervised: unlabeled training data

- Automatic clustering
- Self-training (GANs, RL self-play, …)



https://www.quantamagazine.org/is-alphago-really-such-a-big-deal-20160329/

Loss functions

A measure $\mathscr{L}(\theta)$ of how **badly** the network is performing, as a function of model parameters θ.

- Aim to find $\text{argmin}_\theta \mathscr{L}(\theta)$
- Choice of loss function depends on your objective!

Regression

$$\mathscr{L}_{\text{MSE}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\mathscr{L}_{\text{MAE}} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

...

Classification

$$\mathscr{L}_{\text{Cross-entropy}} = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

...

Generative

$$\mathscr{L}_{KL\text{fwd}} = \frac{1}{n}\sum_{i=1}^{n} \log p(x_i) - \log \hat{p}(x_i), \text{ where } x_i \sim p$$

$$\mathscr{L}_{KL\text{bwd}} = \frac{1}{n}\sum_{i=1}^{n} \log \hat{p}(x_i) - \log p(x_i), \text{ where } x_i \sim \hat{p}$$

y_i and $\hat{y}_i$ respectively true/model evaluations on ith training input

Generative case: we may learn a distribution defined either empirically OR analytically

Stochastic gradient descent

Gradient descent using **stochastic** gradient evaluations.

- Estimate true loss function by sampling "mini-batches"
- Aim to capture distribution properties, rather than population properties
- Good for generalization in standard ML

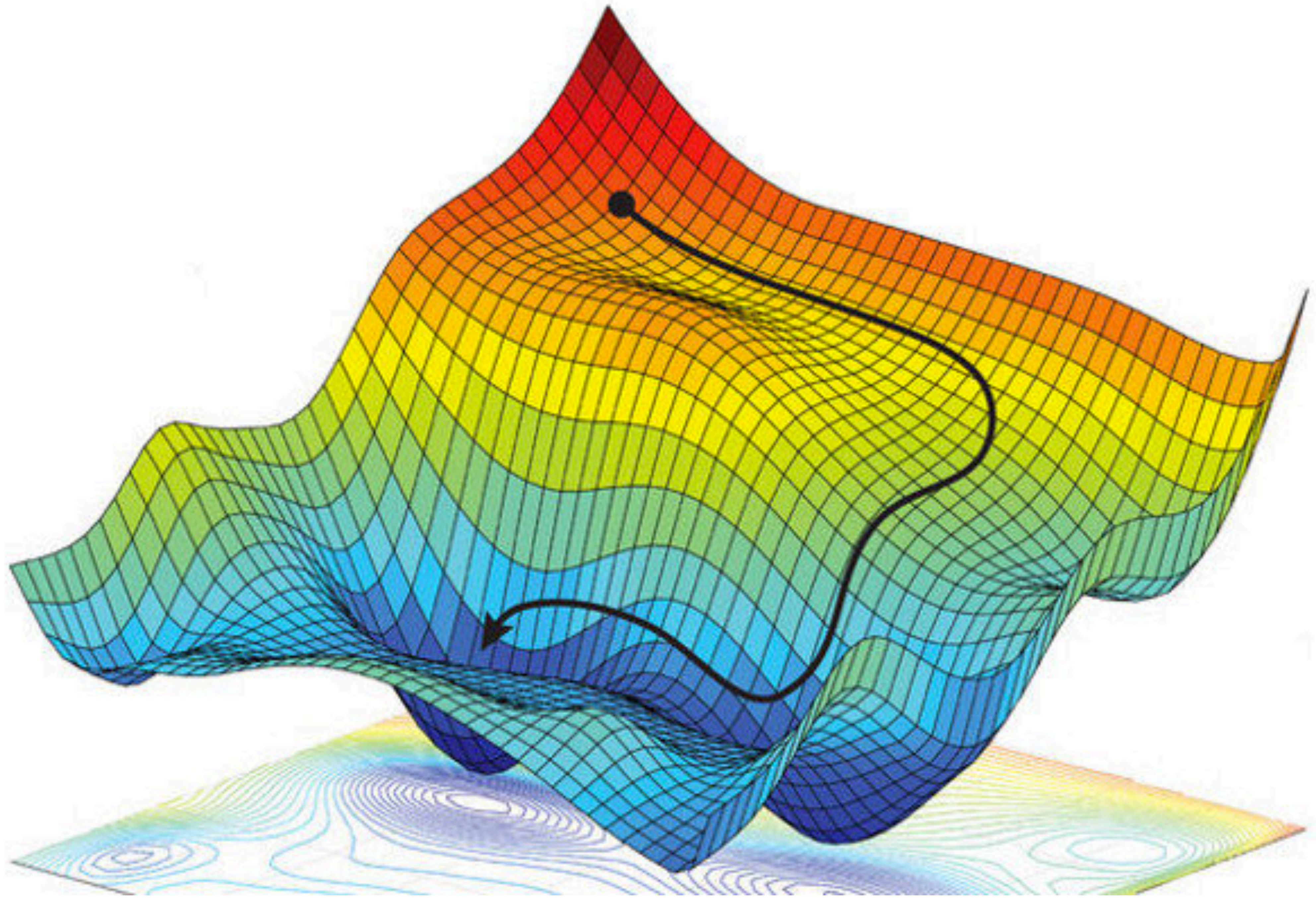


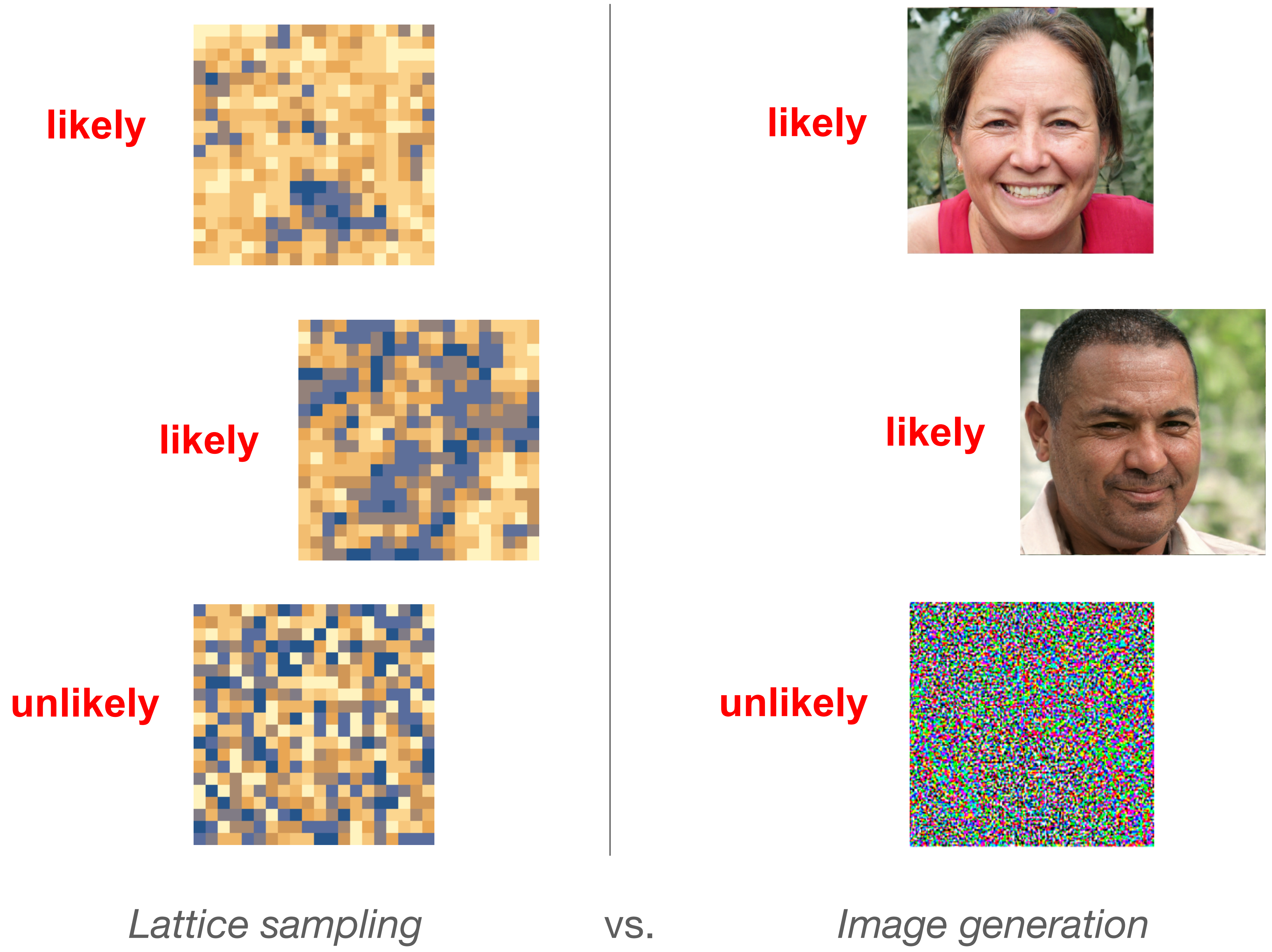
Image credit: 1805.04829

Machine learning applied to lattice gauge theory

Unique features of the lattice problem

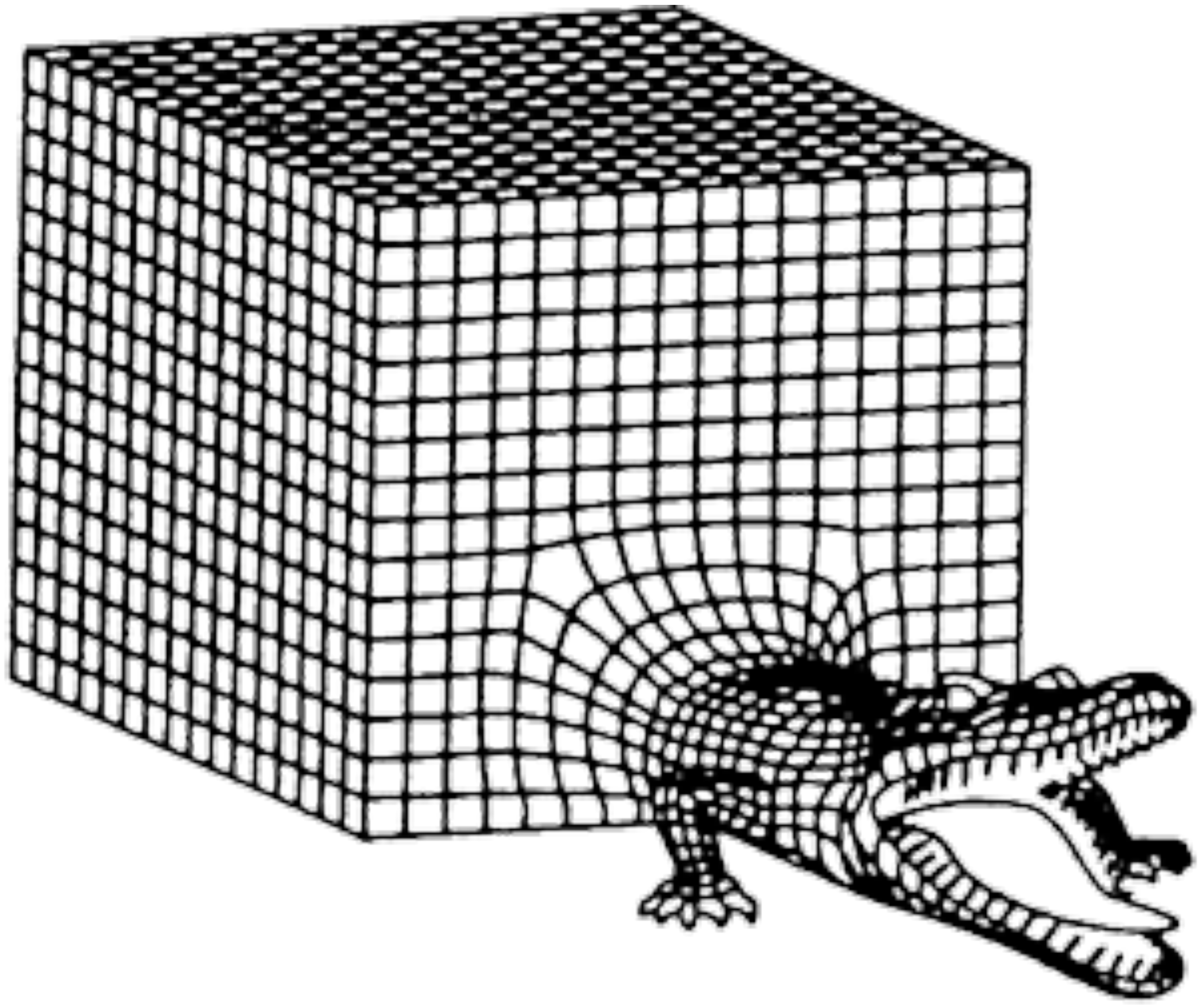
Exactness, inverted data hierarchy, and symmetries

- ⚠️ Demand **unbiased** expectation values
- ⚠️ Have an **inverted data hierarchy**
- ✓ Know **target probability** density $e^{-S(\phi)}/Z$
- ✓ Know physical **symmetries**

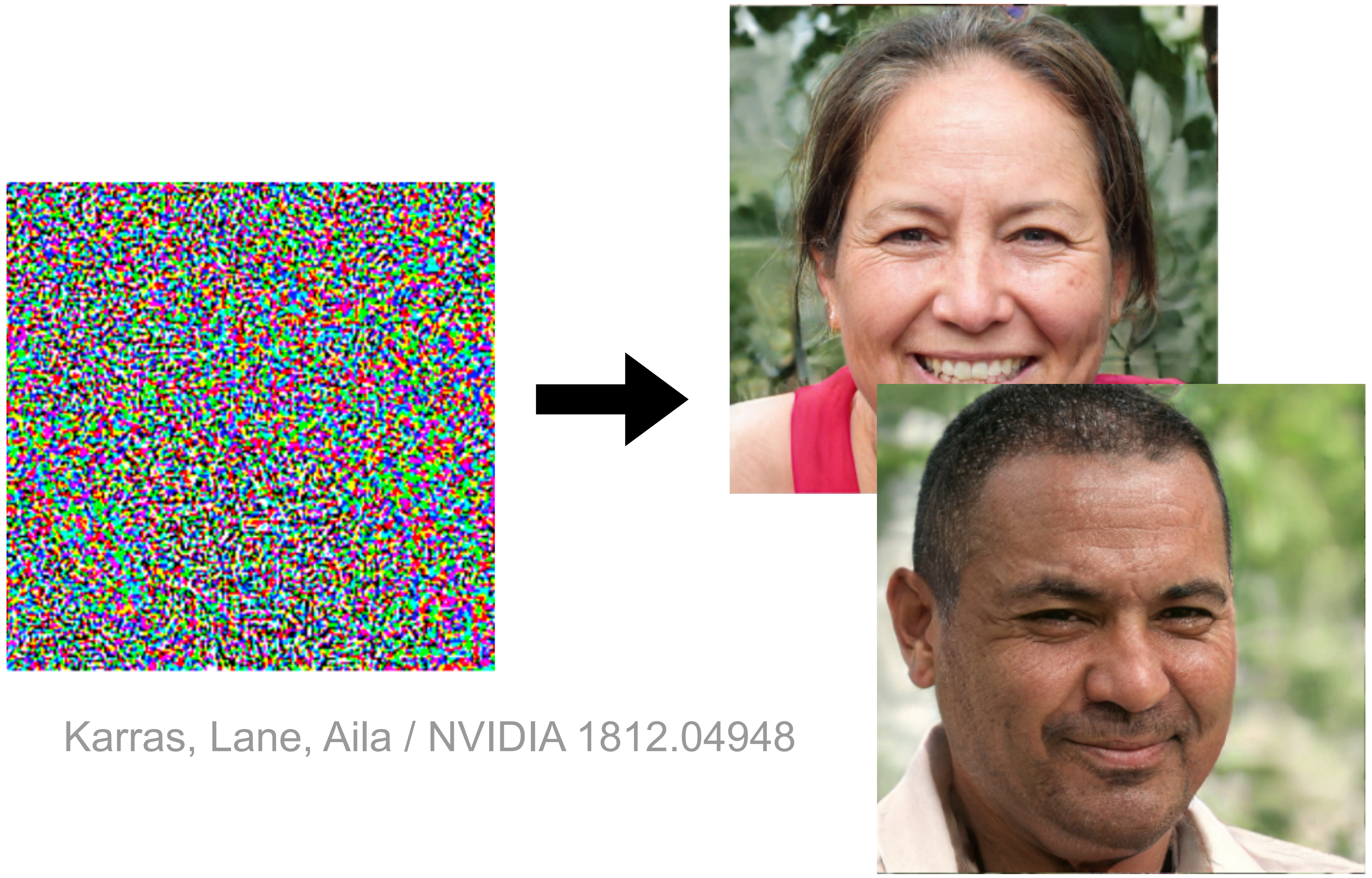


Data hierarchies

The lattice problem suffers from an **"inverted data hierarchy"**.



$\sim 10^9$ **degrees of freedom**
$\sim 10^3$ **samples**



Karras, Lane, Aila / NVIDIA 1812.04948

$\sim 10^6$ **degrees of freedom**
$\sim 10^6 - 10^8$ **samples**

Image credit: Stefan Sint

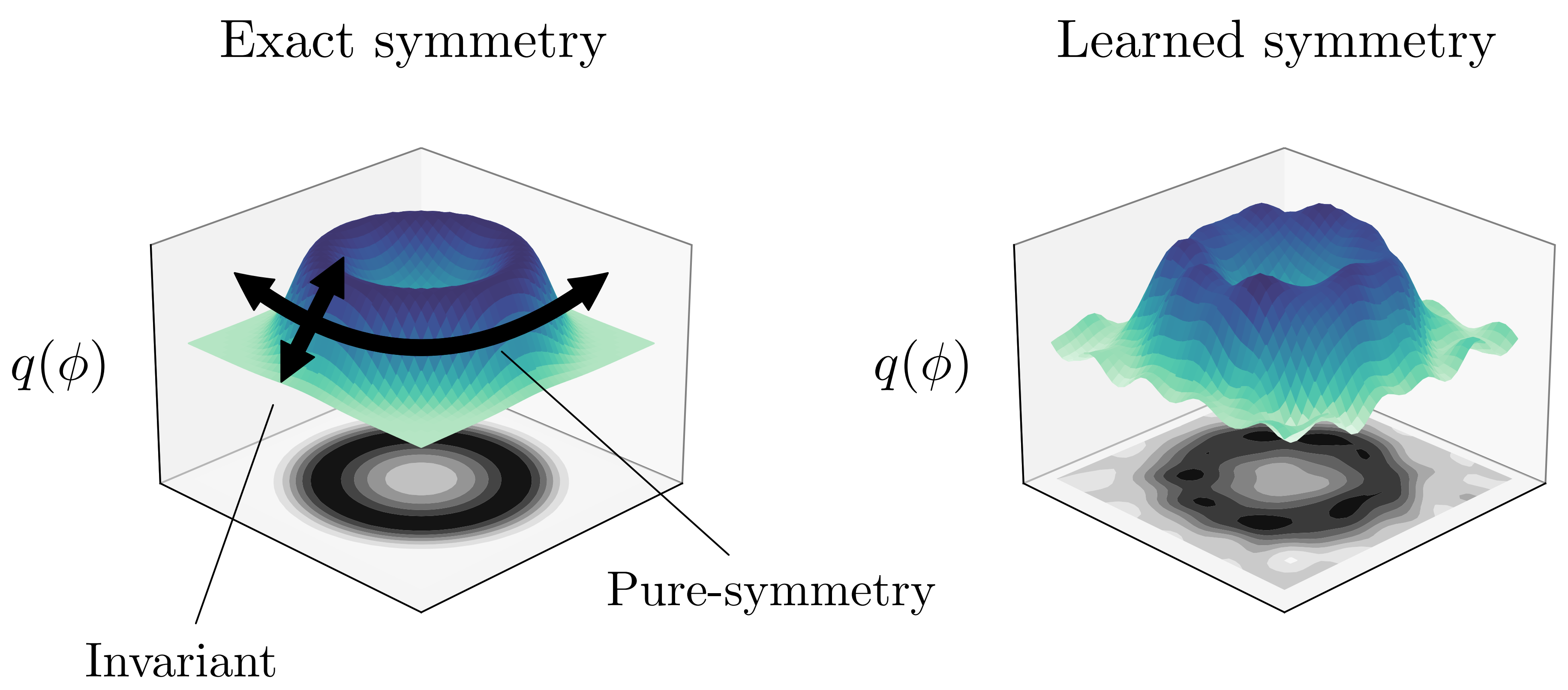
Symmetries

For **ensemble generation**, symmetries…

- Constrain the form of the Boltzmann distribution

For **observable measurements**, symmetries…

- Determine classes of valid interpolating operators



Rawat & Wang Neur. Comp. 29 (2017) 2352
LeCun+ NeurIPS 2 (1989)
Cohen & Welling 1602.07576
Dieleman+ 1602.02660
+ many others

Symmetry-enhanced ML models are being developed.

- Success in lattice contexts may start interesting discussion on "Bitter lesson" theory

*"The biggest lesson that can be read from 70 years of AI research is that **general methods** that leverage computation are ultimately **the most effective**, and by a large margin."* Rich Sutton (2019), "The Bitter Lesson"

Classifying lattice phases

Regression task which can be addressed via standard neural networks

[van Nieuwenberg+ Nature Phys. 13 (2017) 435]

[Li+ 1703.02369]

[Wetzel+ PRB96 (2017) 184410]

[Zhou+ PRD100 (2019) 011501]

[Bachtis+ PRE102 (2020) 033303]

[Bluecher+ PRD101 (2020) 094507]

[Alexandrou+ EPJB (2020) 93 226]

[Tan+ 2103.10846]

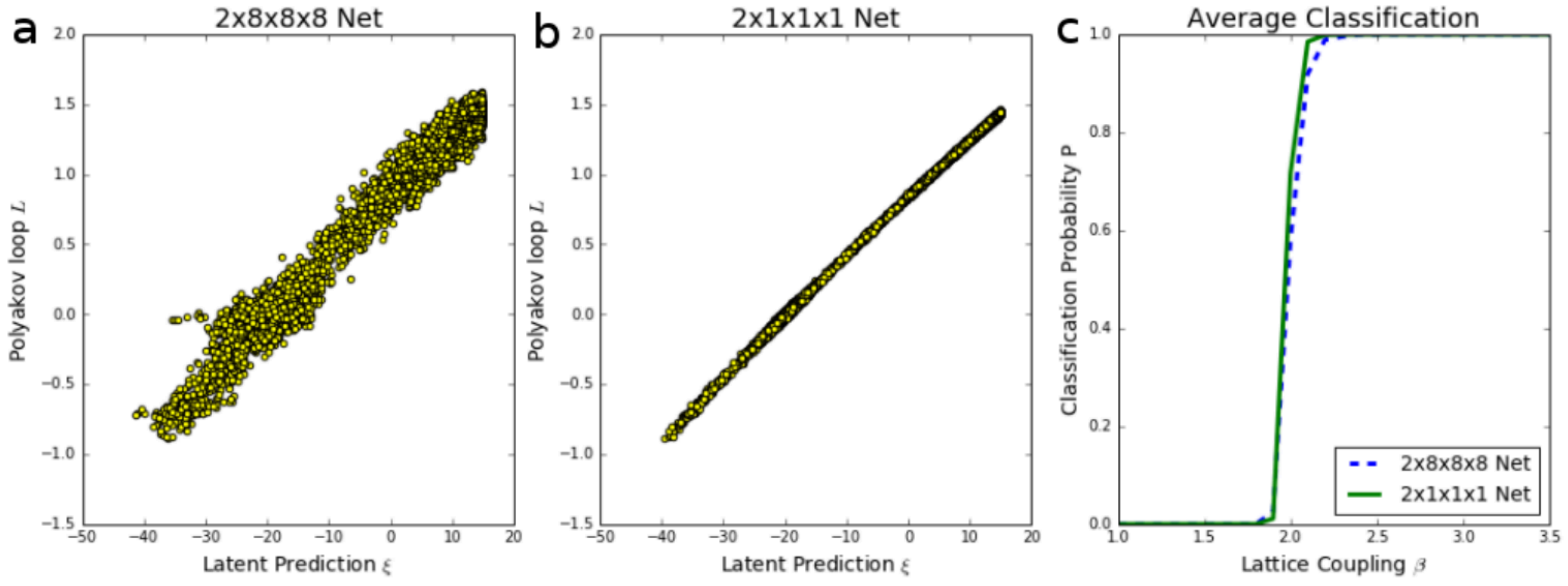
[Boyda+ PRD103 (2021) 014509]

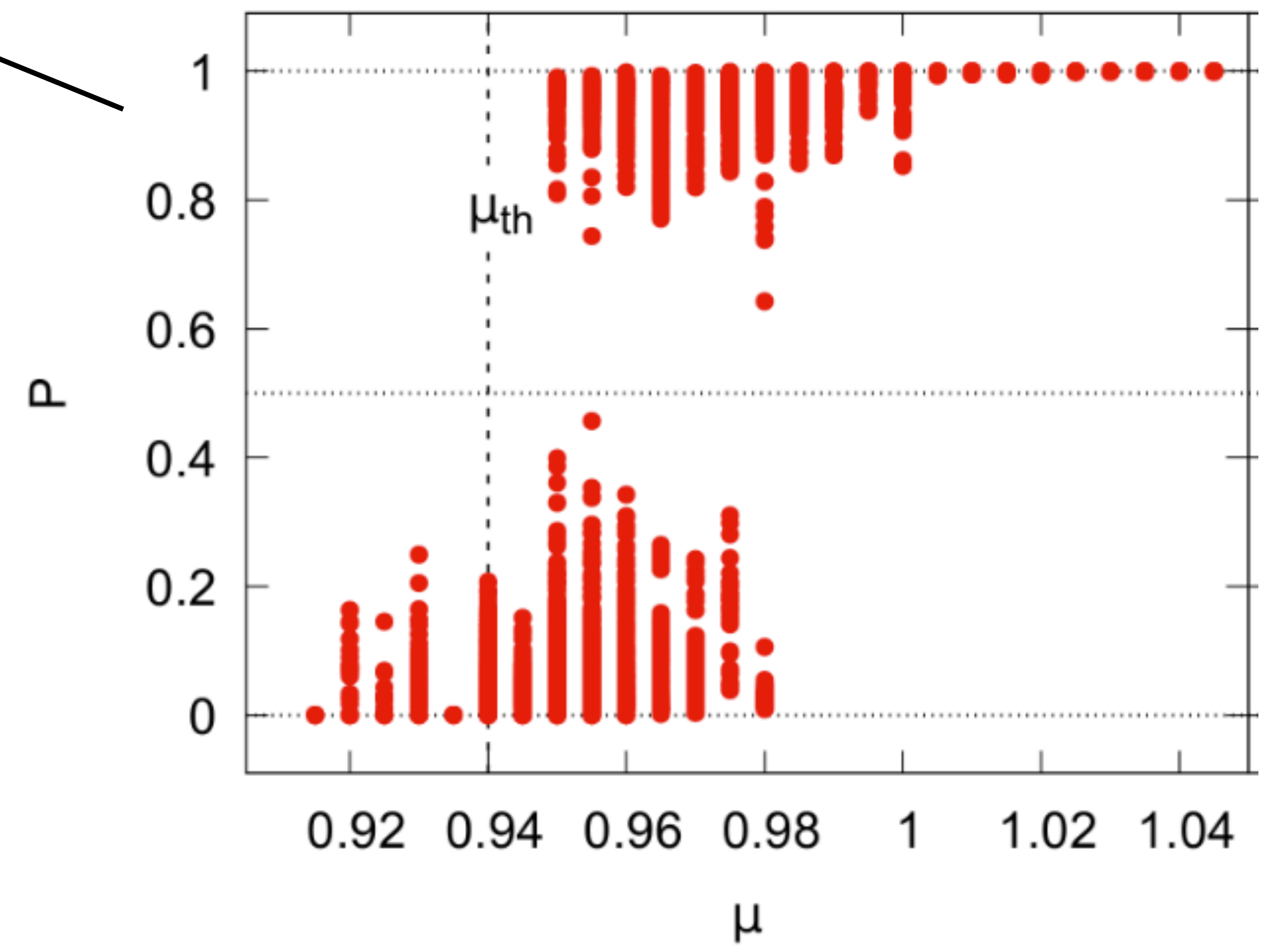
[Palermo+ PoS(LATTICE2021)030]

[Yau+ SciPost Phys. Core 5 (2022) 032]

+ many more

SU(2) gauge theory deconfinement transition

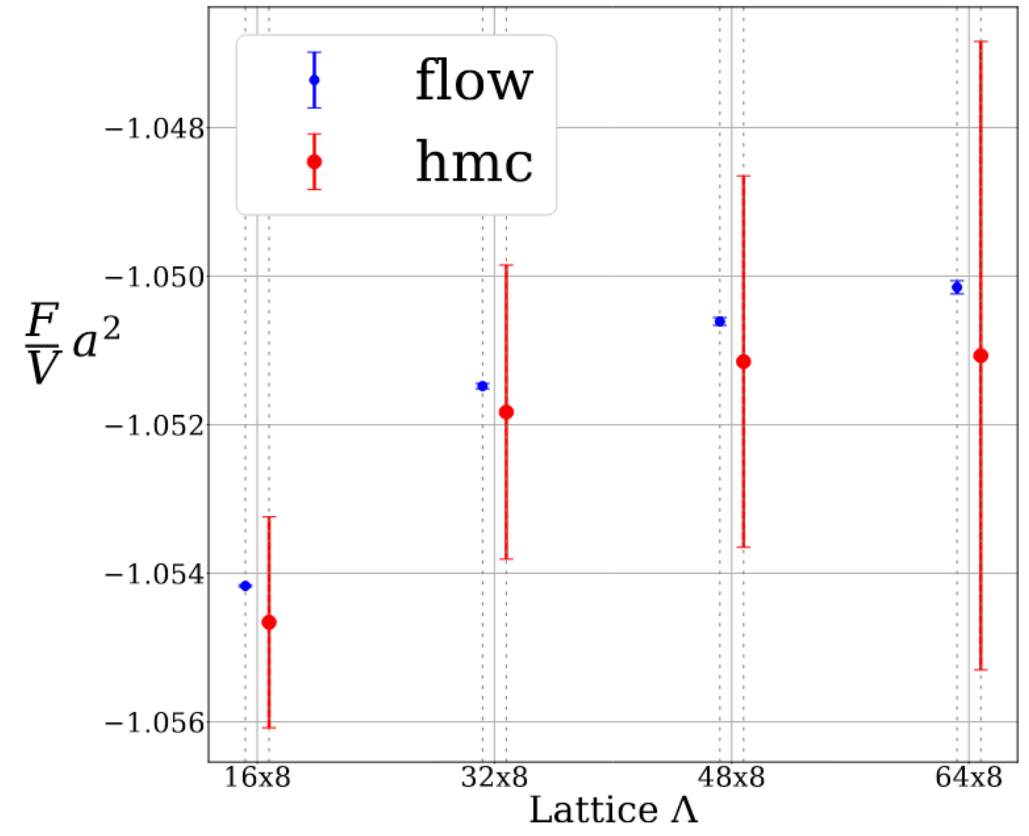




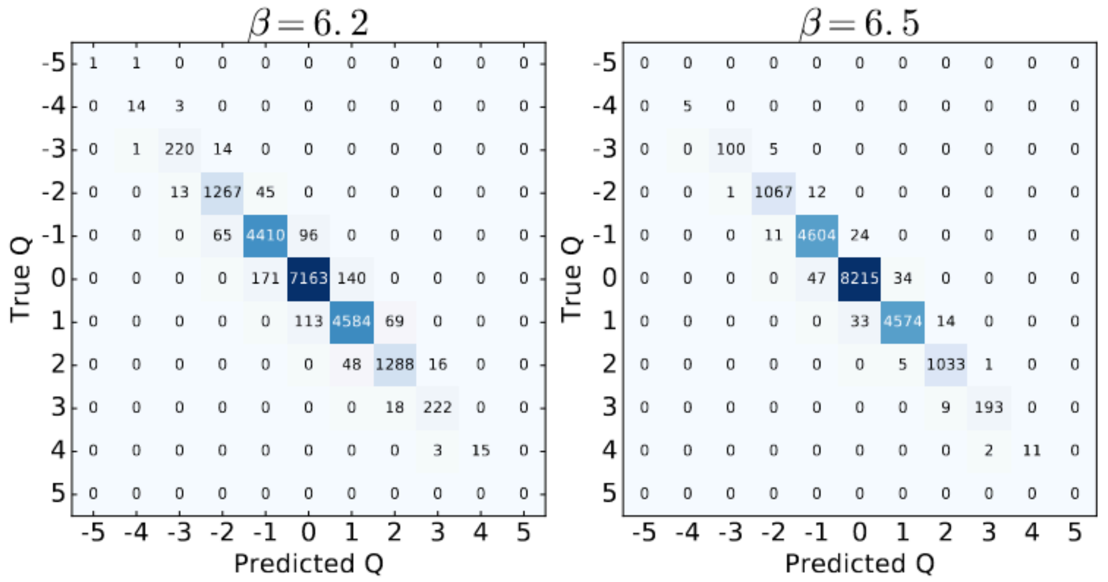
Scalar field theory transition with chemical potential

ML estimators for observables

Thermodynamic observables [Nicoli+ PRL126 (2021) 032001]

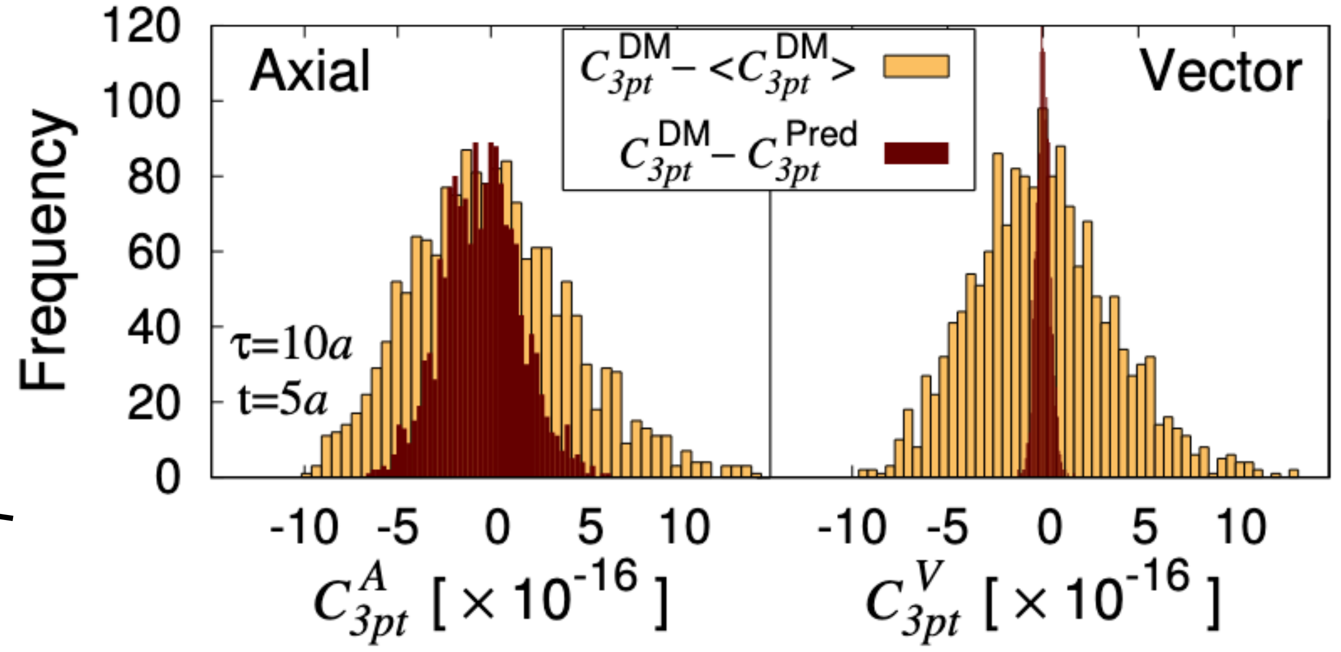


Predicting observables from raw lattices

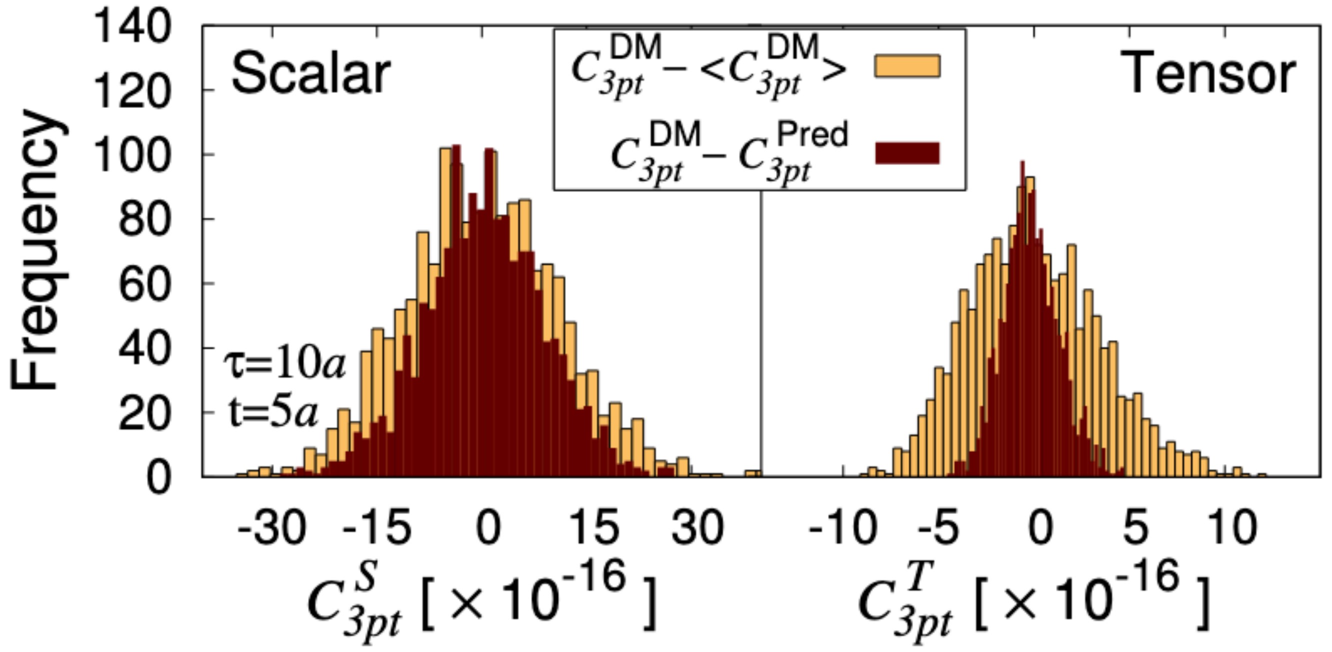


[Matsumoto+ 1909.06238]
[Bulusu+ PRD104 (2021) 074504]
[Favoni+ PRL128 (2022) 032003]

Cross-observable estimates [Yoon+ PRD100 (2019) 014504] [Zhang+ PRD101 (2020) 034516]



Learned contour deformations [Alexandru+ PRD96 (2017) 094505] [Detmold, GK+ PRD102 (2020) 014514] + many more



Preconditioners for matrix inversion
[Lehner and Wettig PRD108 (2023) 034503]

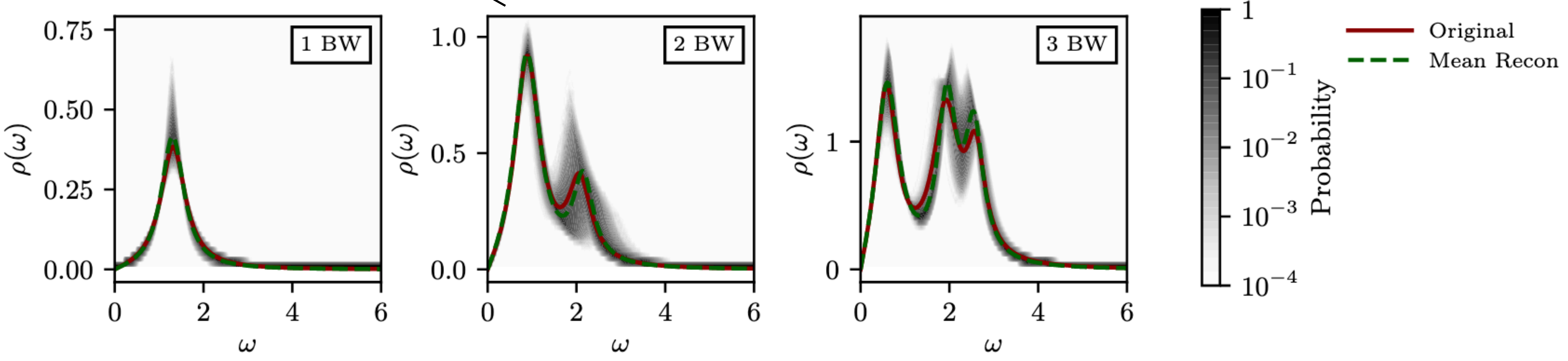
Spectral function reconstruction

Euclidean-time Green's functions → spectral densities $\rho(\omega)$
(i.e. inverse Källén–Lehmann)

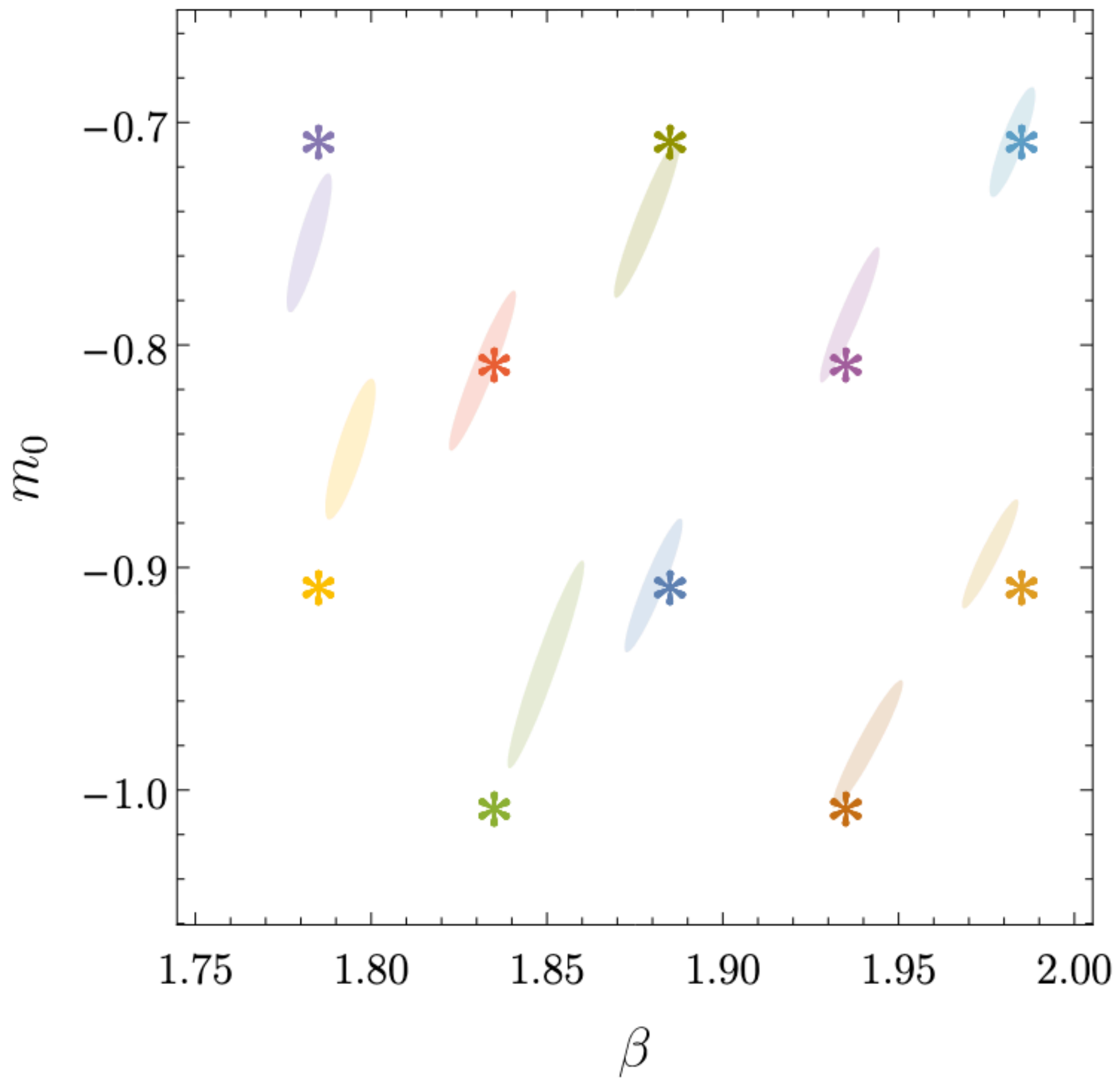
Neural-network parameterization of $\rho(\omega)$

[Offler+ 1912.12900]
[Kades+ PRD102 (2020) 096001]
[Chen+ 2110.13521]

[Wang+ PRD106 (2022) L051502]
[Shi+ CPC282 (2023) 108547]
[Horak+ PRD105 (2022) 036014]



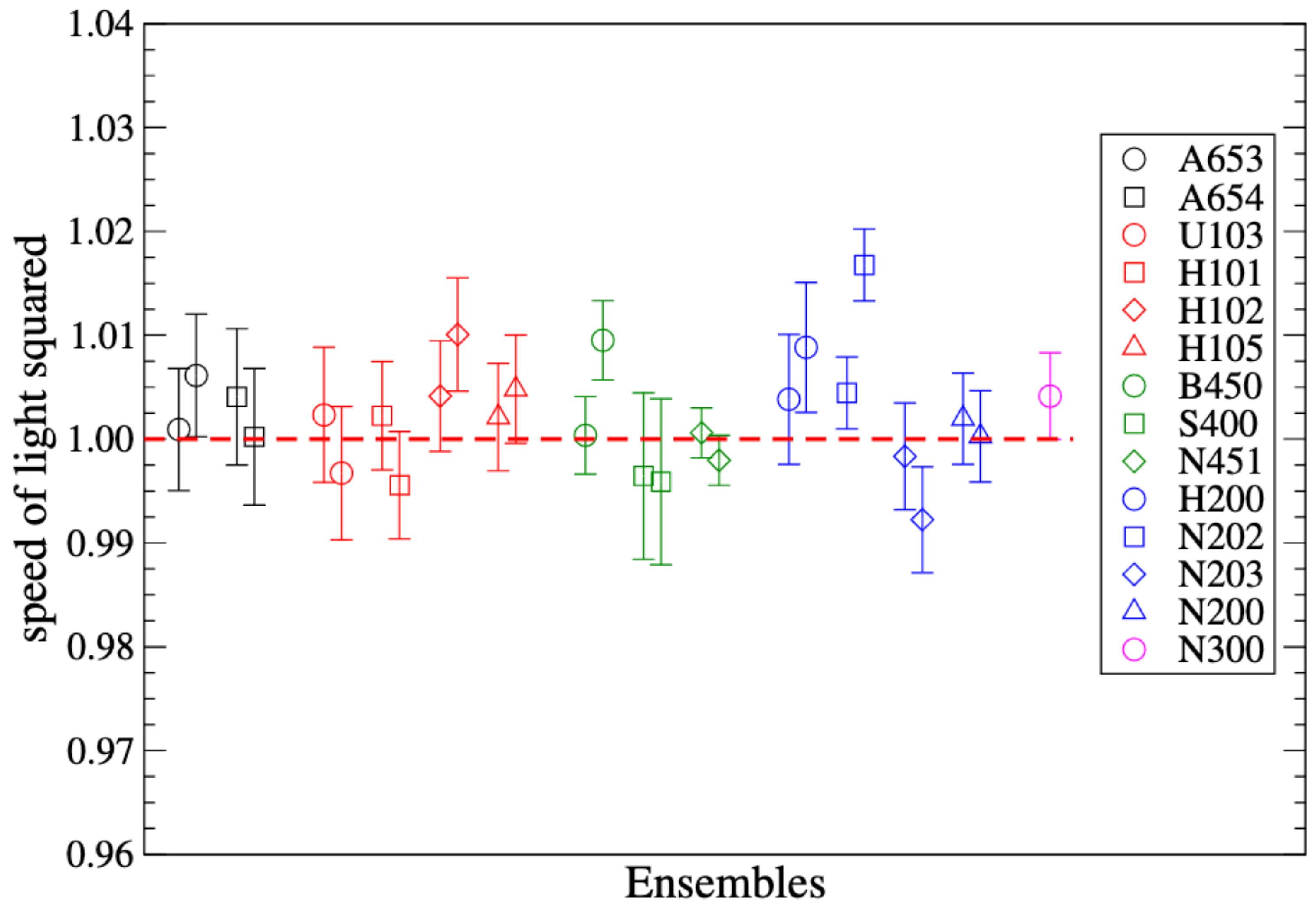
Action parameter regression



[Shanahan+ PRD97 (2018) 094506]

- Regression from configs to action
- Gauge-symmetry important to include in networks!

- Fully-connected network to predict (measured masses) → (action parameters)
- Speed-of-light tuning on anisotropic lattices
  [Hudspith and Mohler PRD106 (2022) 034508]



- Learned action approximating RG fixed point
  [Holland+ 2311.17816]

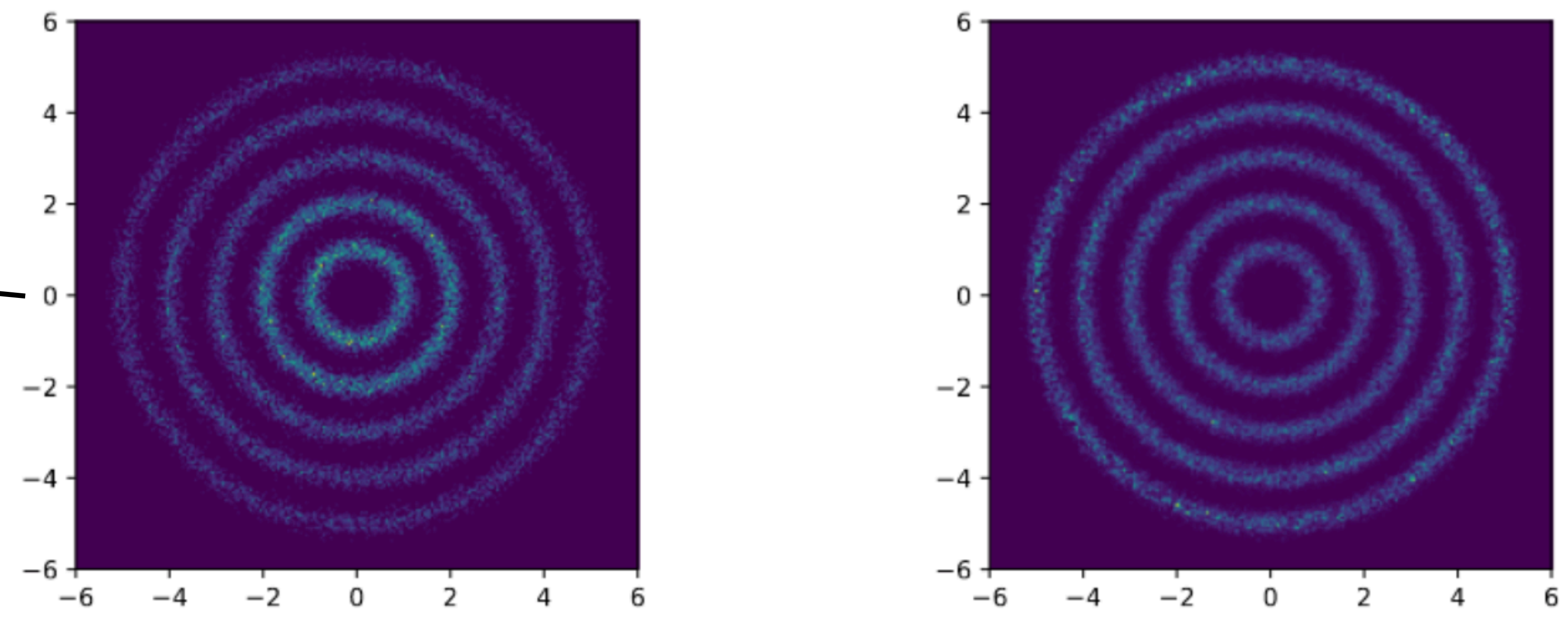
Ensemble generation

Finding improved Markov chain Monte Carlo updates

[Wang PRE96 (2017) 051301]
[Huang and Wang PRB95 (2017) 035105]
[Song+ NeurIPS (2017) 1706.07561]
[Tanaka and Tomiya 1712.03893]
[Foreman+ ICLR (2021) 2105.03418]
[Albandea+ 2302.08408]

“Self-learning Monte Carlo”

[Liu+ PRB95 (2017) 241104] + many more



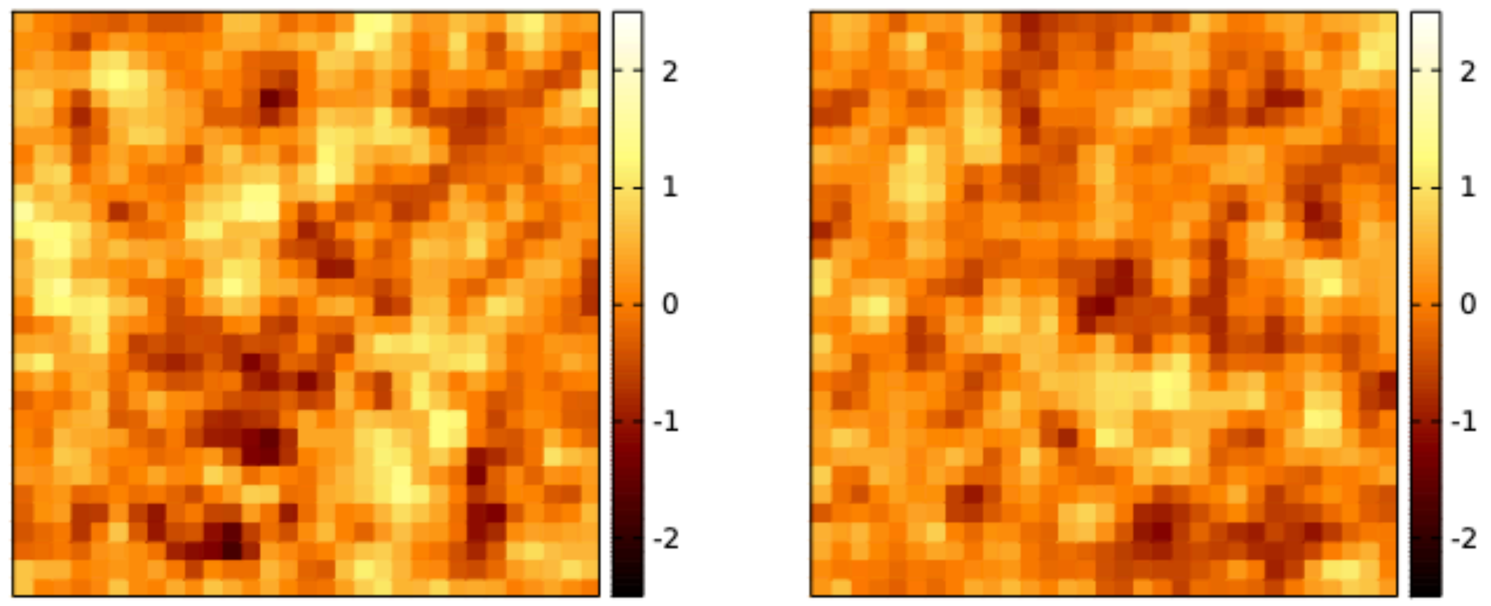
(c) HMC   (d) A-NICE-MC

Directly sampling configurations

[Köhler+ 1910.00753]
[Pawlowski and Urban MLST1 (2020) 045011]
[Carrasquilla+ Nature Mach. Int. 1 (2019) 155]

“Flow-based sampling”

[Albergo, GK, Shanahan PRD100 (2019) 034515]
[Nicoli+ PRL126 (2021) 032001]
[Gerdes+ 2207.00283] + many more



(a) HMC   (b) GAN-overrelaxation

Review: Cranmer, GK+ Nat. Rev. Phys. 5 (2023) 526

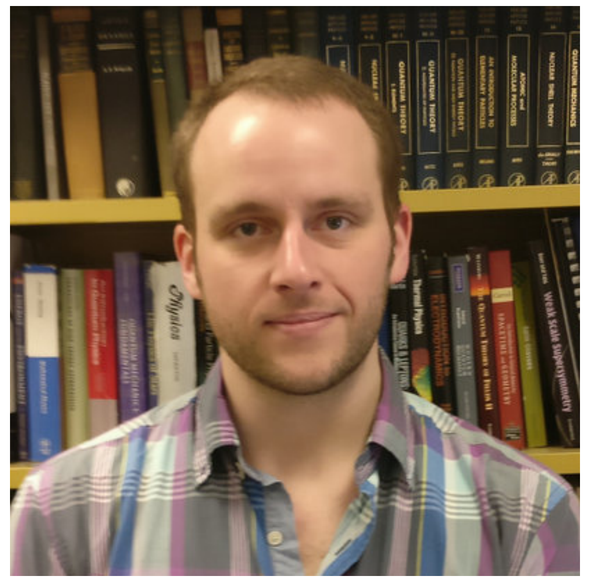
Case study:
flow-based sampling

Massachusetts Institute of Technology

The NSF Institute for Artificial Intelligence and Fundamental Interactions


Phiala Shanahan


Denis Boyda


Dan Hackett


Fernando Romero-López


Julian Urban


Ryan Abbott

NEW YORK UNIVERSITY


Michael Albergo

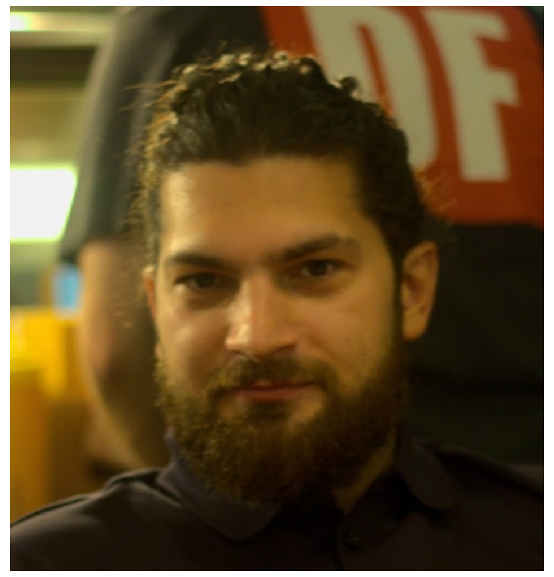
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON


Kyle Cranmer

DeepMind


Sébastien Racanière


Danilo Rezende


Aleksander Botev


Alexander Matthews


Ali Razavi

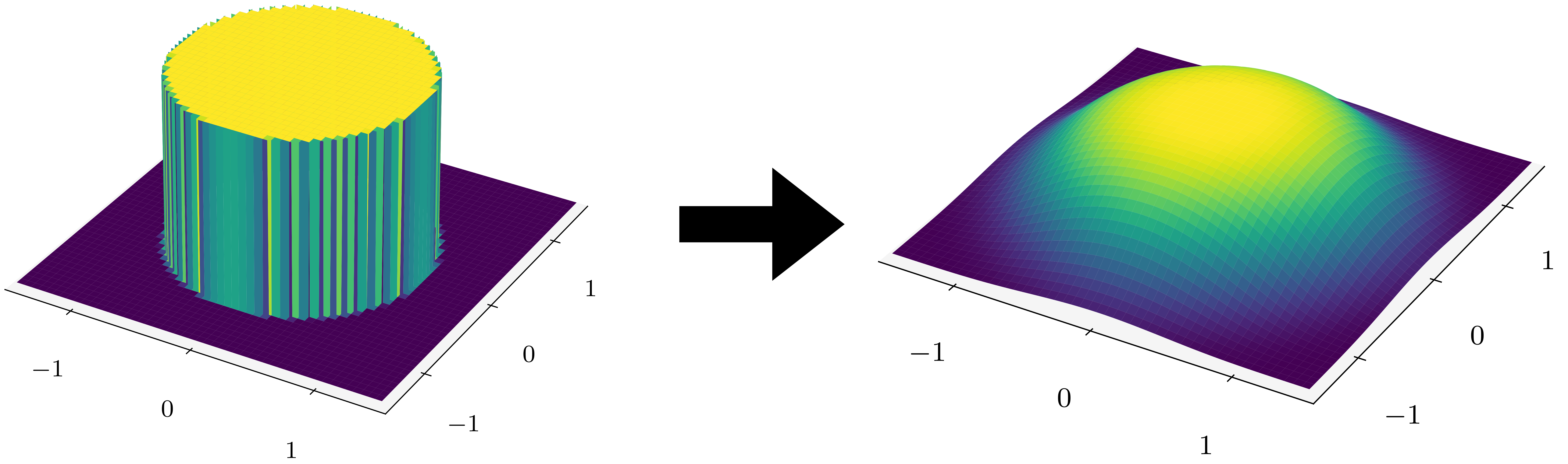
A taste of flow-based sampling

AKA a “normalizing flow”

Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145
Lüscher CMP293 (2010) 899

Box-Muller transform (Marsaglia polar form)

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



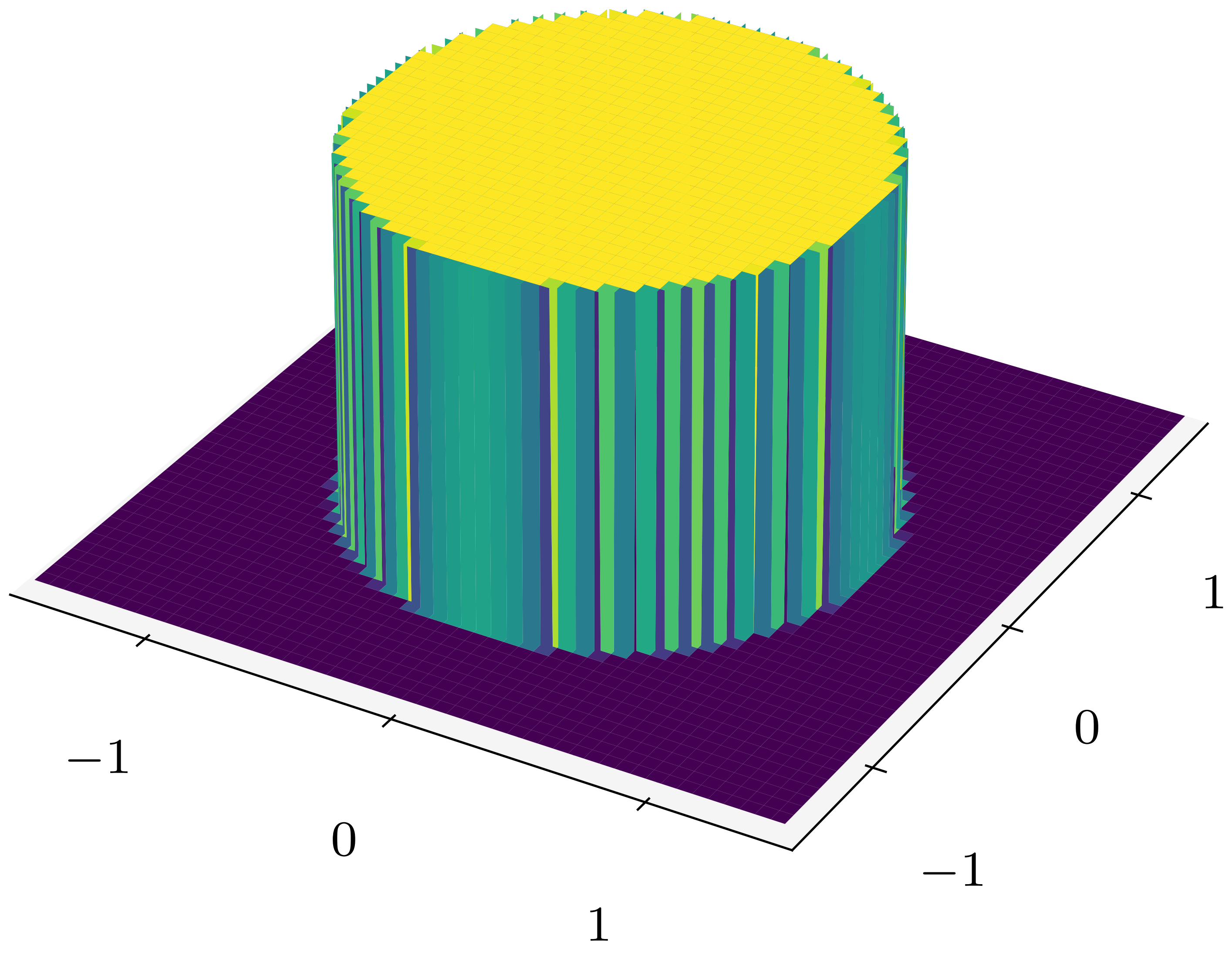
A taste of flow-based sampling

AKA a "normalizing flow"

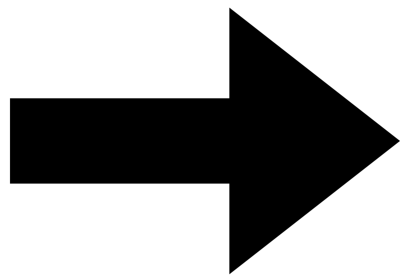
Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145
Lüscher CMP293 (2010) 899

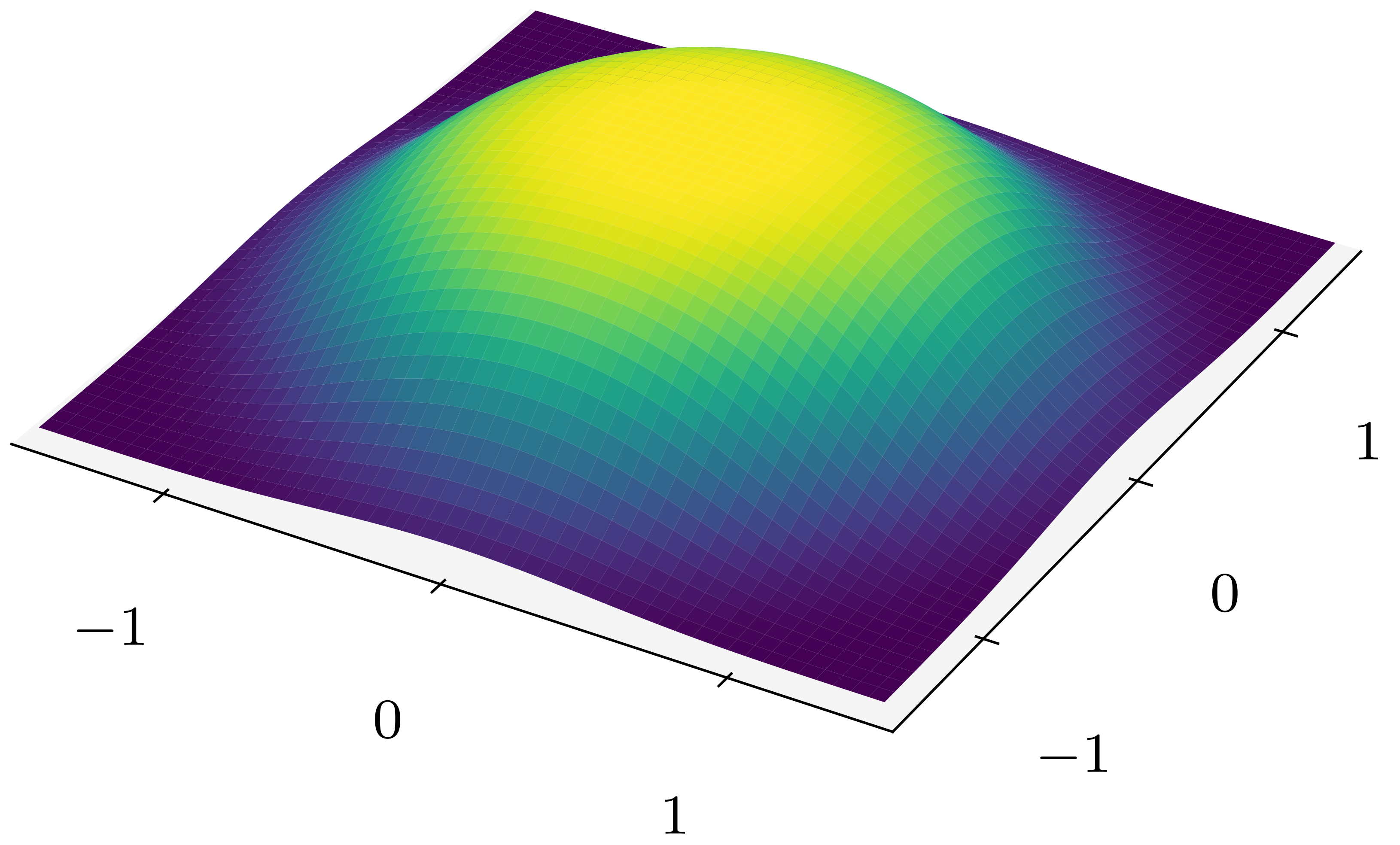
Box-Muller transform (Marsaglia polar form)

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



Flow f





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

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

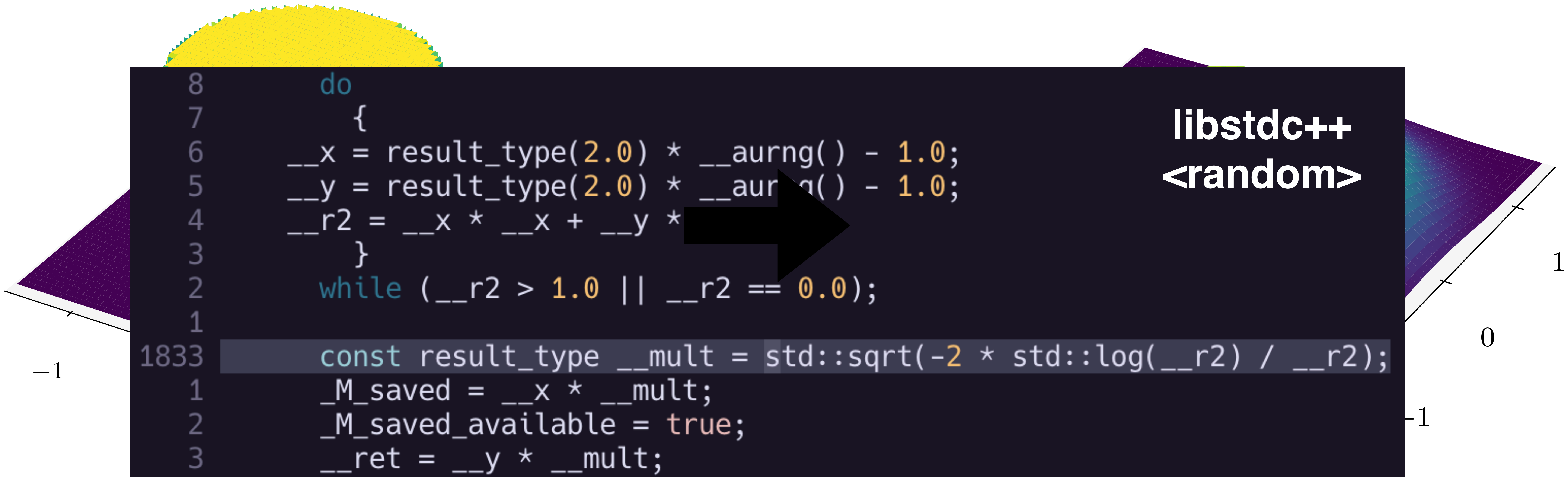
A taste of flow-based sampling

AKA a “normalizing flow”

Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145
Lüscher CMP293 (2010) 899

Box-Muller transform (Marsaglia polar form)

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



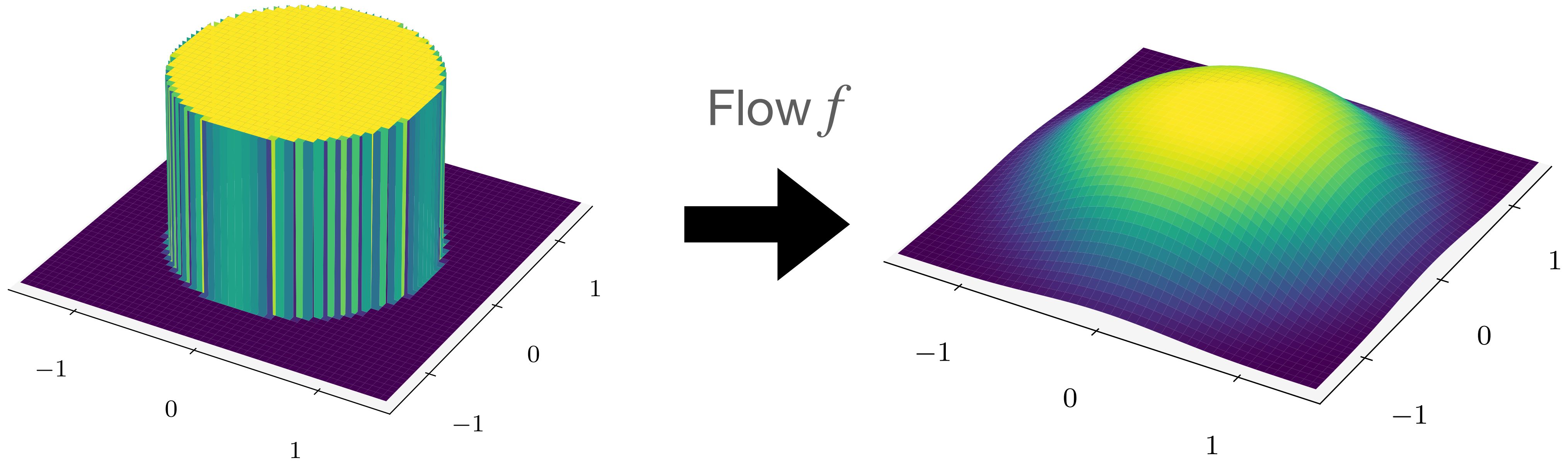
(Simple) Prior density:
$r(x, y)$

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

Normalizing flows

General idea:

Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145
Lüscher CMP293 (2010) 899

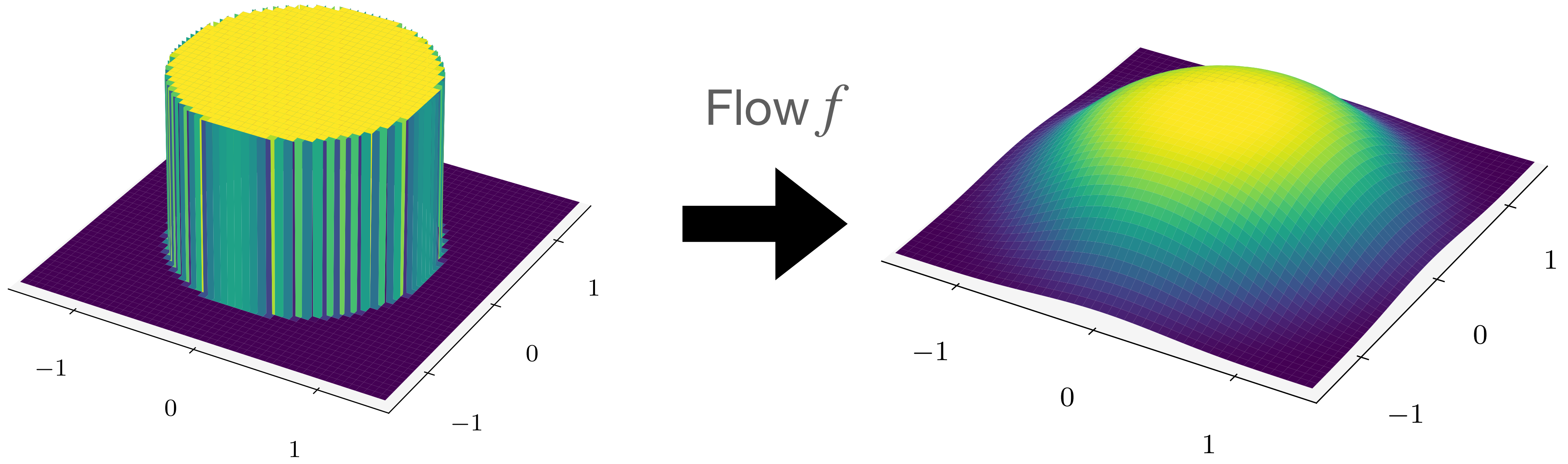


Simple prior distribution $r(\xi)$ ⟶ Desired model distribution $q(\phi)$

Normalizing flows

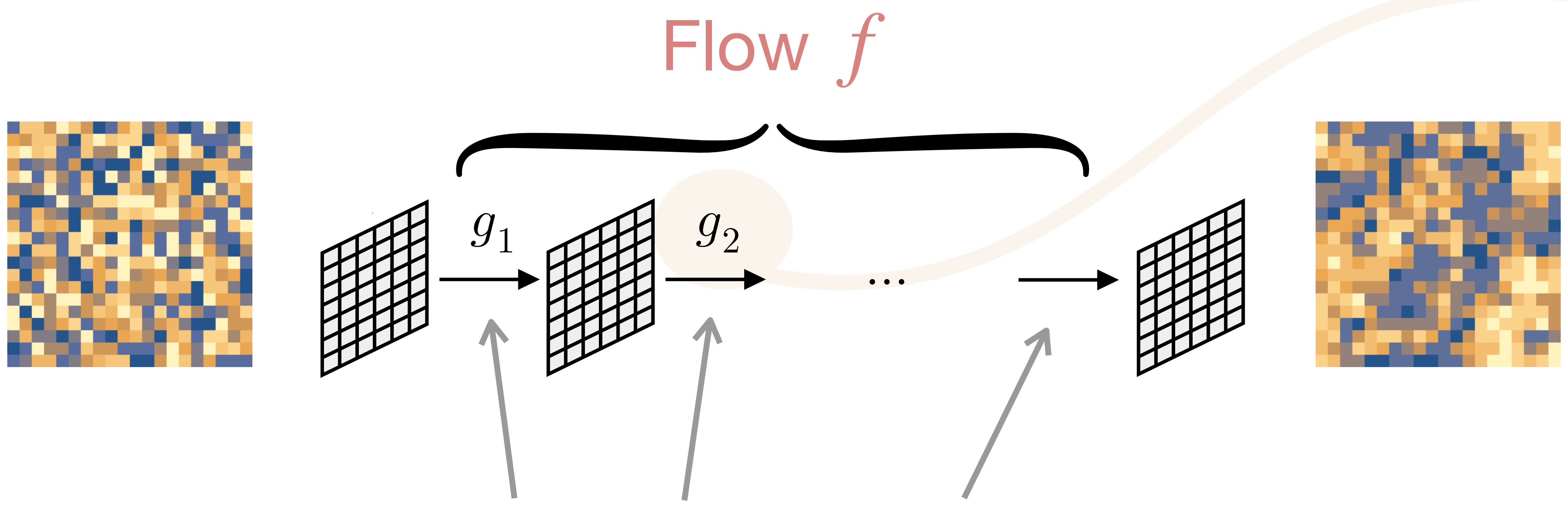
General idea:

Tabak & Vanden-Eijnden CMS8 (2010) 217
Tabak & Turner CPA66 (2013) 145
Lüscher CMP293 (2010) 899



Simple prior distribution $r(\xi)$ → Desired model distribution $q(\phi)$

With **machine learning**:



Each layer is a **diffeomorphism** with **tractable Jacobian**.

“RealNVP” coupling layer g_i

Dinh, Sohl-Dickstein, Bengio 1605.08803

Could mitigate critical slowing down by training models to directly sample configs at various lattice spacings

Albergo, GK, Shanahan PRD100 (2019) 034515

Self-training scheme

Optimization must be designed for inverted data hierarchy in the lattice problem.

Albergo, GK, Shanahan PRD100 (2019) 034515

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

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

2. Measure using samples ϕ_i **from the model**

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

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]

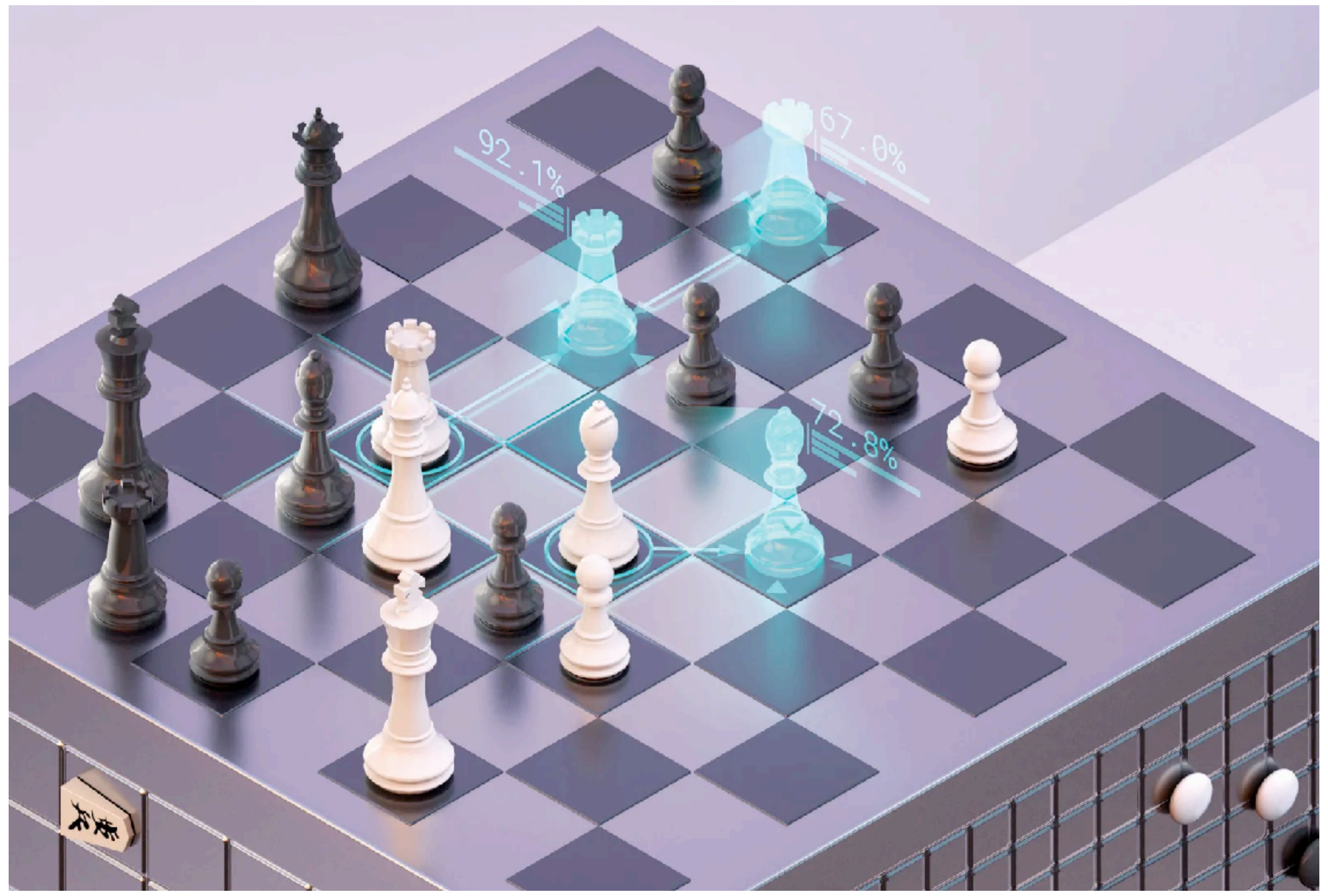
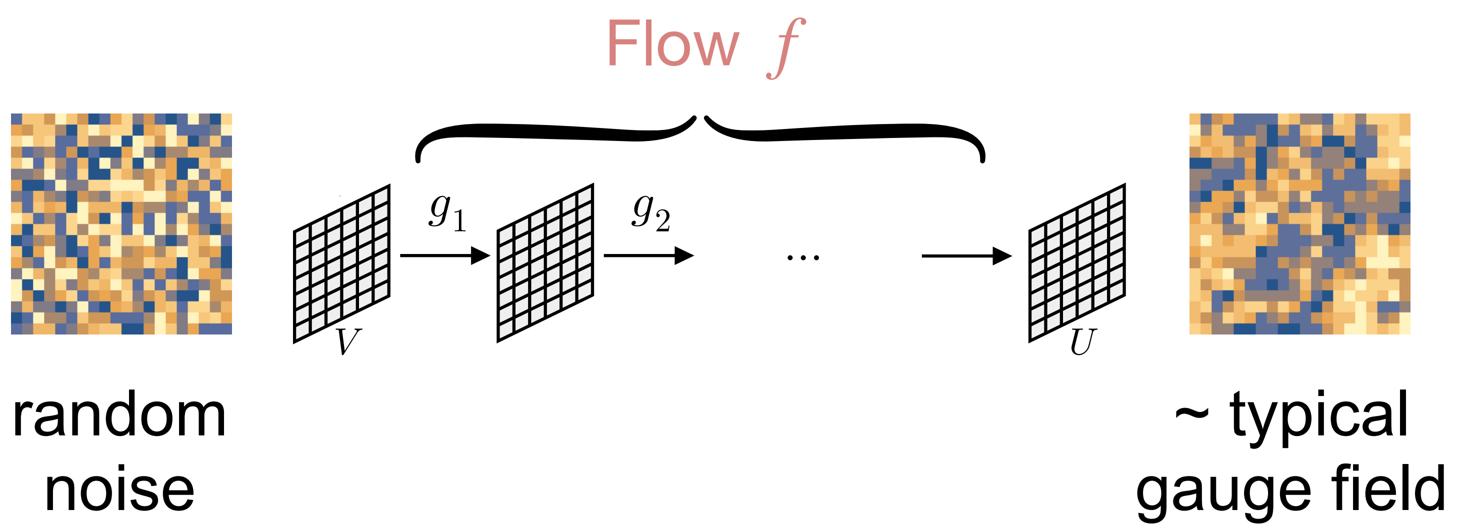
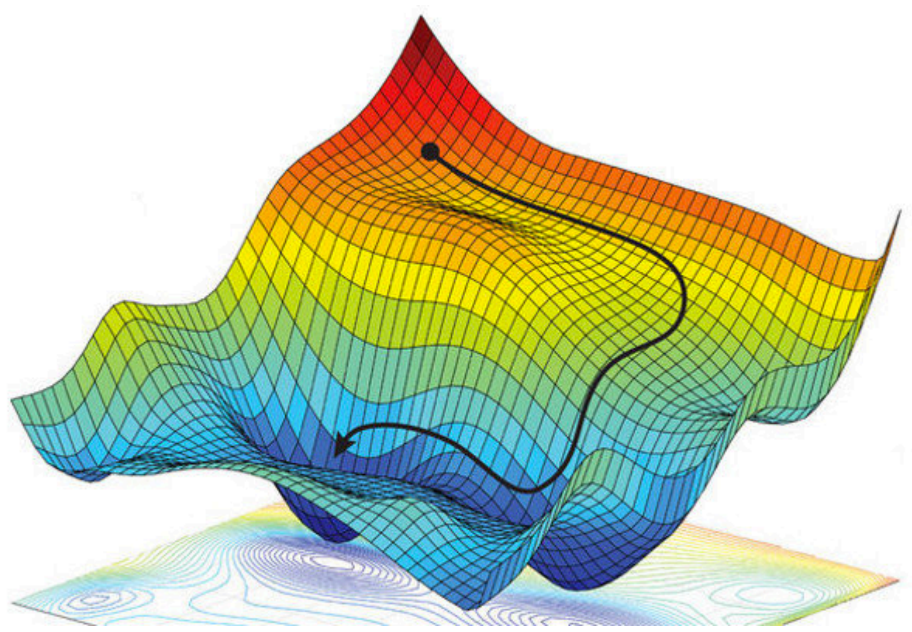
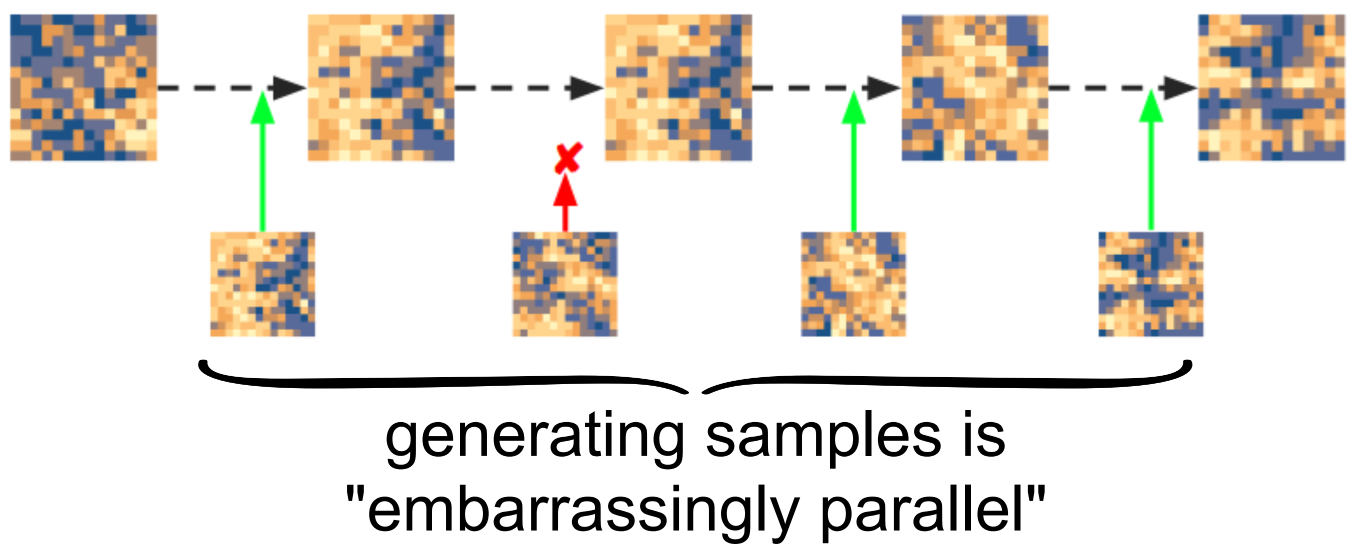


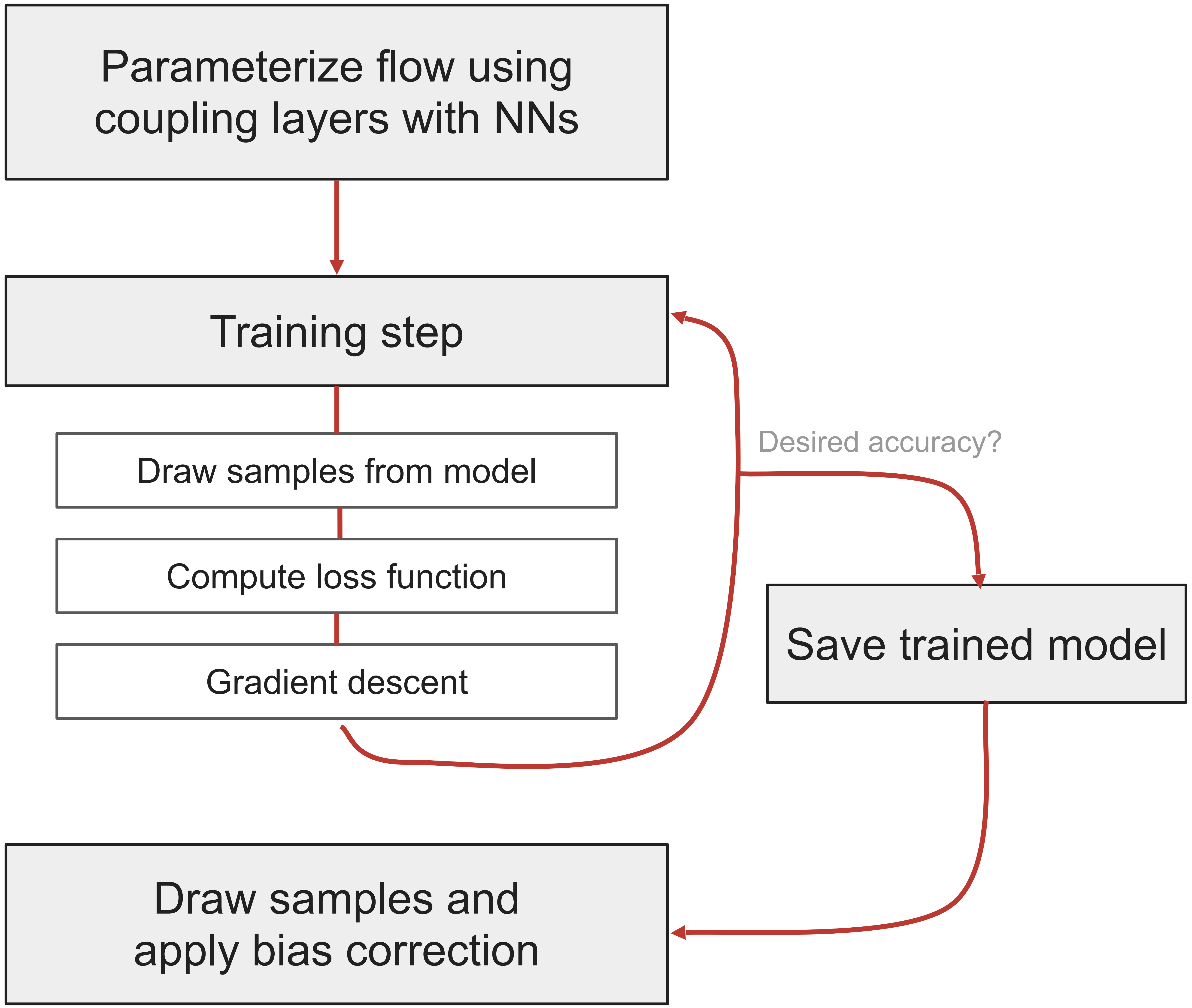
Image credit: DeepMind

Birds-eye view







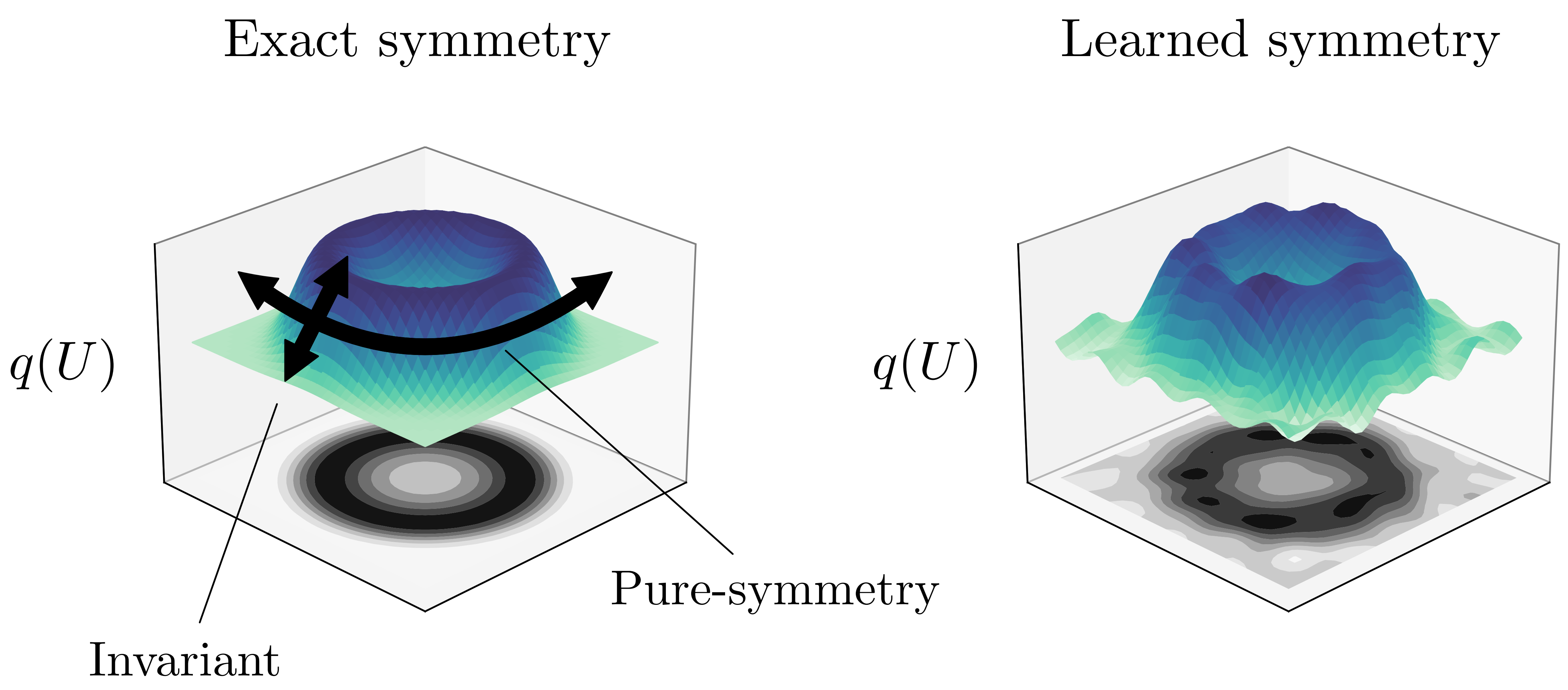


Lattice gauge theory & Symmetries

Lattice gauge theory actions (typically) **satisfy several symmetries**:

1. (Discrete) translational symmetries
2. Hypercubic symmetries
3. Gauge symmetries

Symmetries **factor** distribution into uniform component along symmetry direction, and non-uniform component along invariant direction. Schematically:

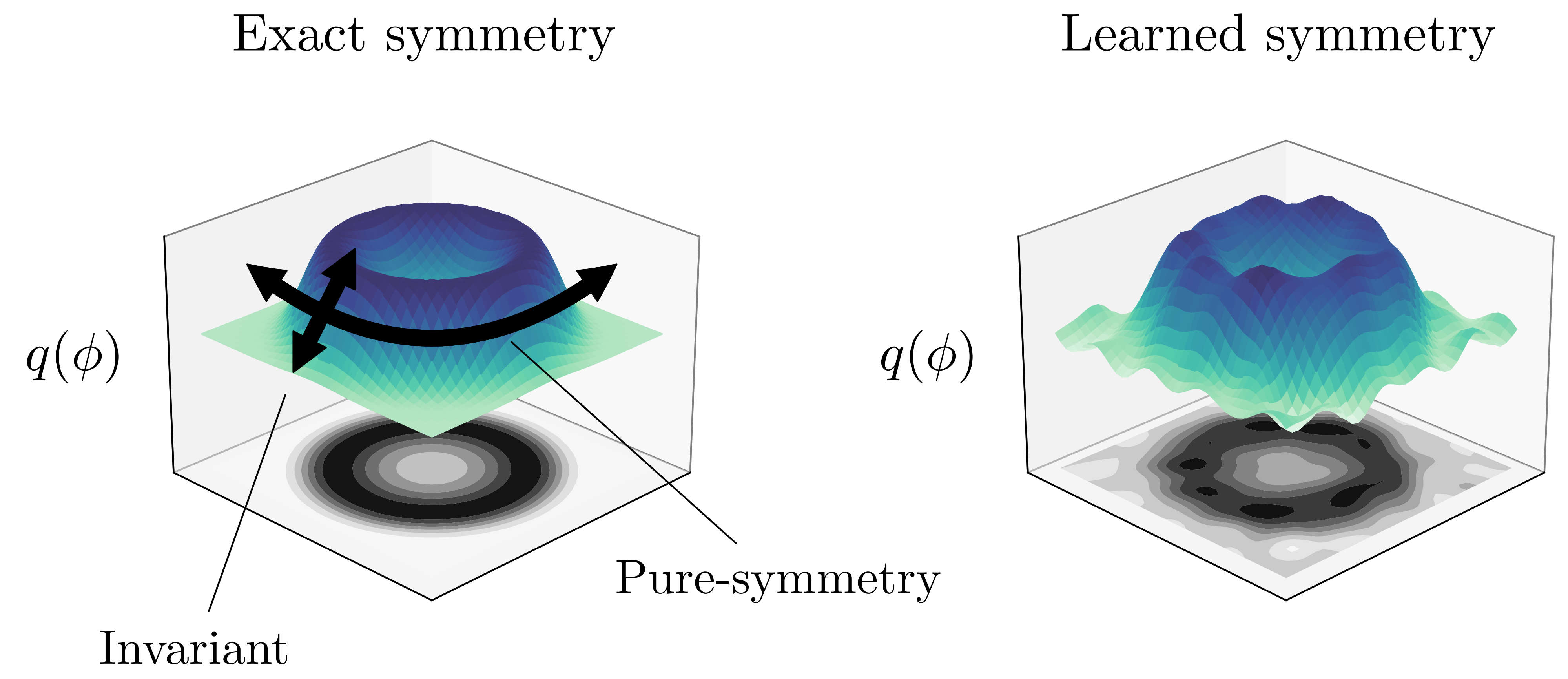


Symmetries in flows

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

Symmetries…

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



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

Cohen, Welling 1602.07576

$$r(t \cdot U) = r(U) \qquad f(t \cdot U) = t \cdot f(U)$$

Gauge symmetry

Distribution should be symmetric under $(\Omega \cdot U)_\mu(x) = \Omega(x) U_\mu(x) \Omega^\dagger(x+\hat{\mu})$ for all gauge-group-valued fields $\Omega(x)$.

GK, Albergo, … PRL125 (2020) 121601

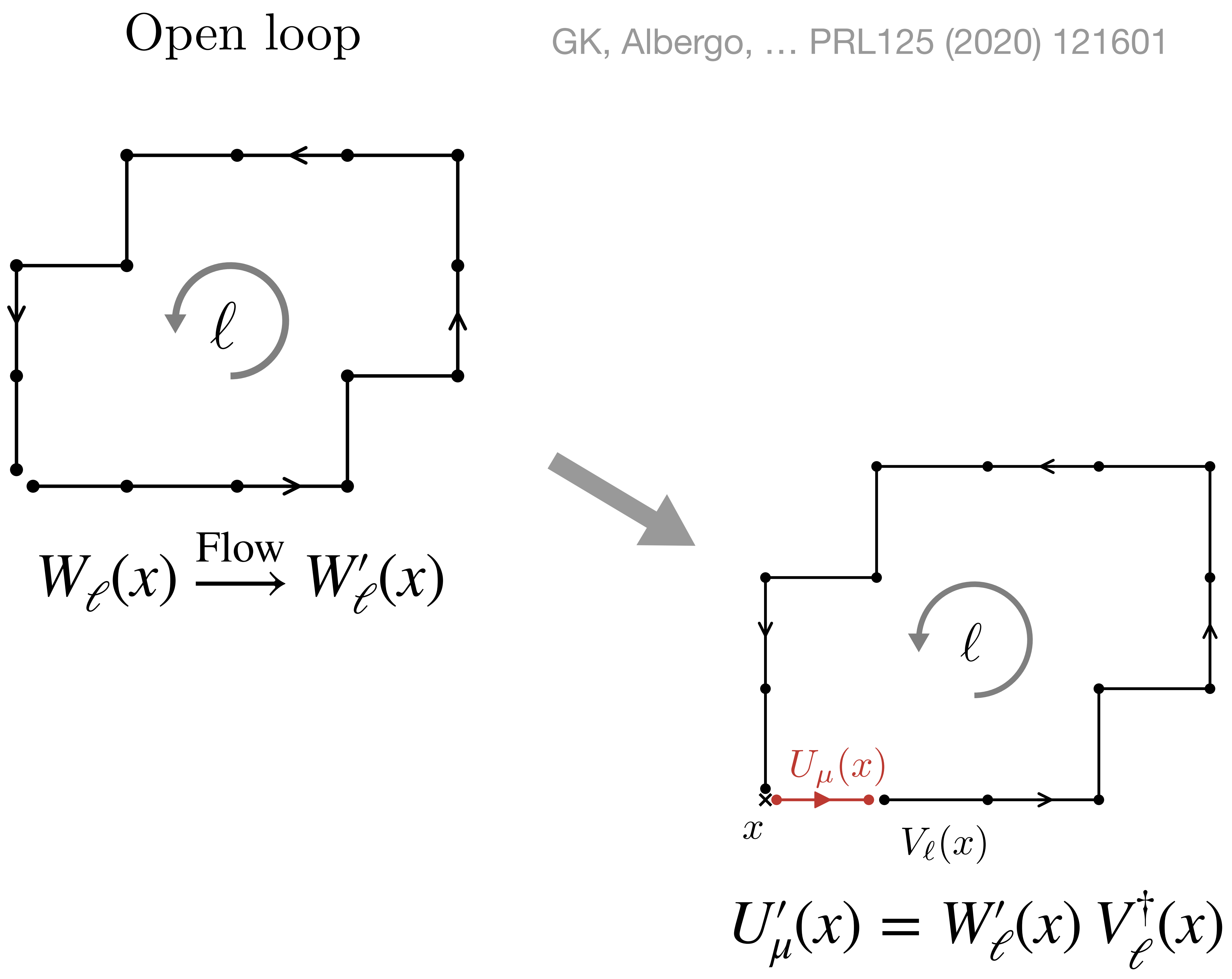
Gauge-invariant prior:

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

Gauge-equivariant flow:

Coupling layers act on
(untraced) Wilson loops.

Loop transformation
easier to satisfy.



Gauge symmetry

Distribution should be symmetric under $(\Omega \cdot U)_\mu(x) = \Omega(x) U_\mu(x) \Omega^\dagger(x + \hat{\mu})$ for all gauge-group-valued fields $\Omega(x)$.

Gauge-invariant prior:

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

Gauge-equivariant flow:

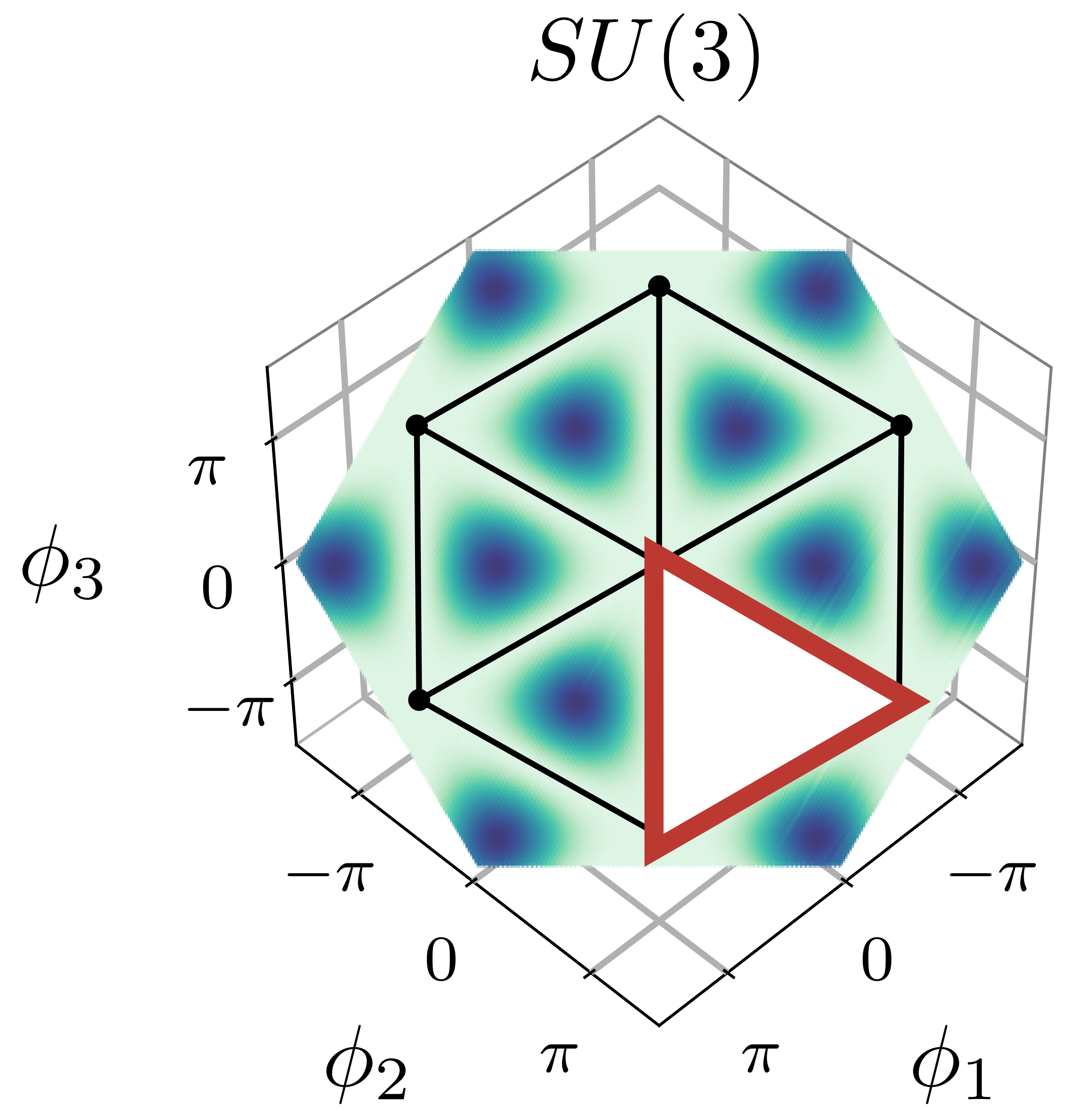
Coupling layers act on (untraced) Wilson loops.

Loop transformation easier to satisfy.

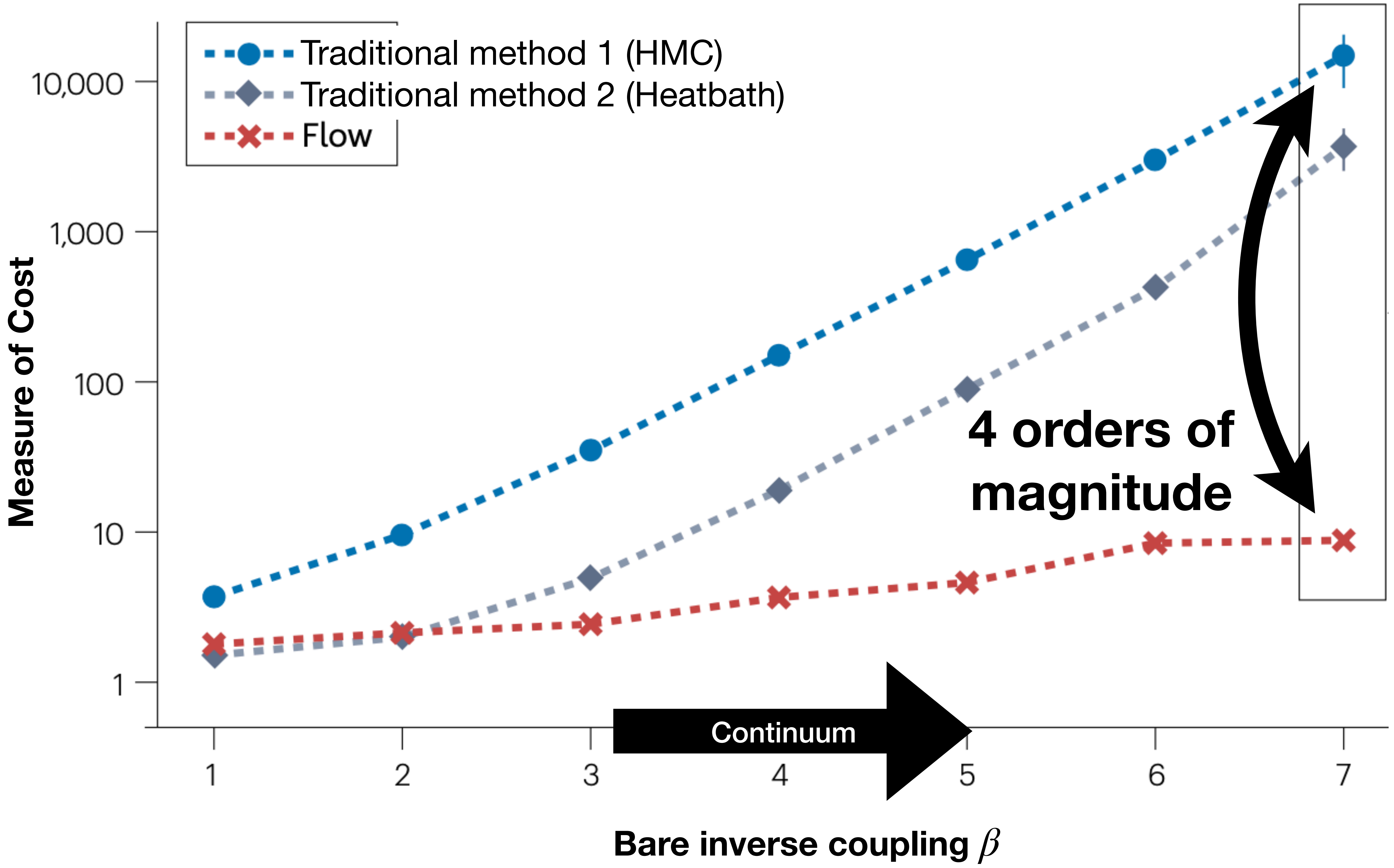
Custom flows designed for $U(1)$ and $SU(N)$ gauge manifolds

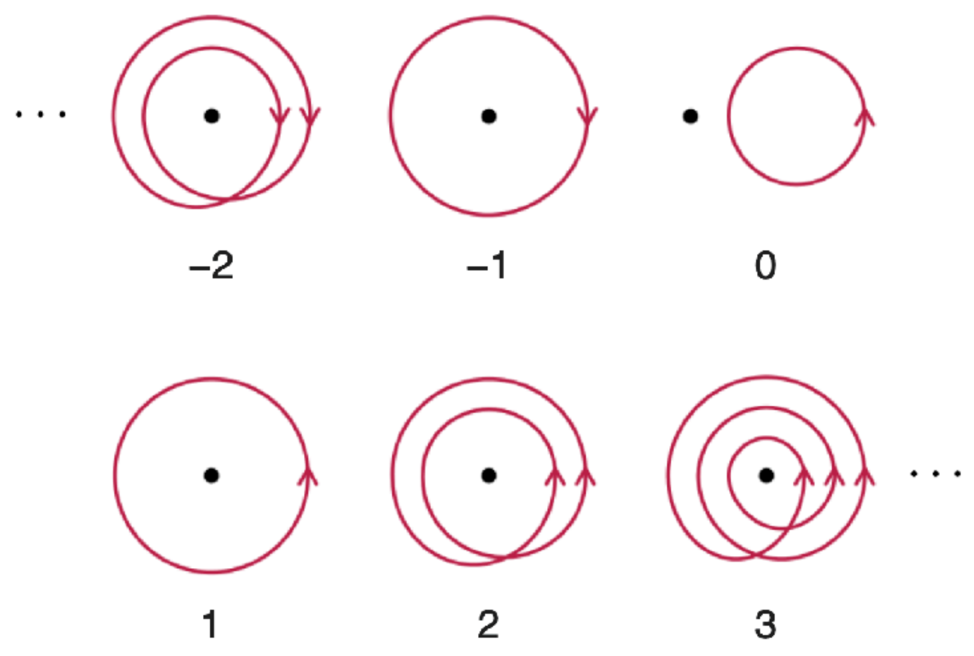
GK, Albergo, … PRL125 (2020) 121601

Boyda, GK, … PRD103 (2021) 074504
Rezende, …, GK, … PMLR119 (2020) 8083



Topological freezing solved for a U(1) gauge theory



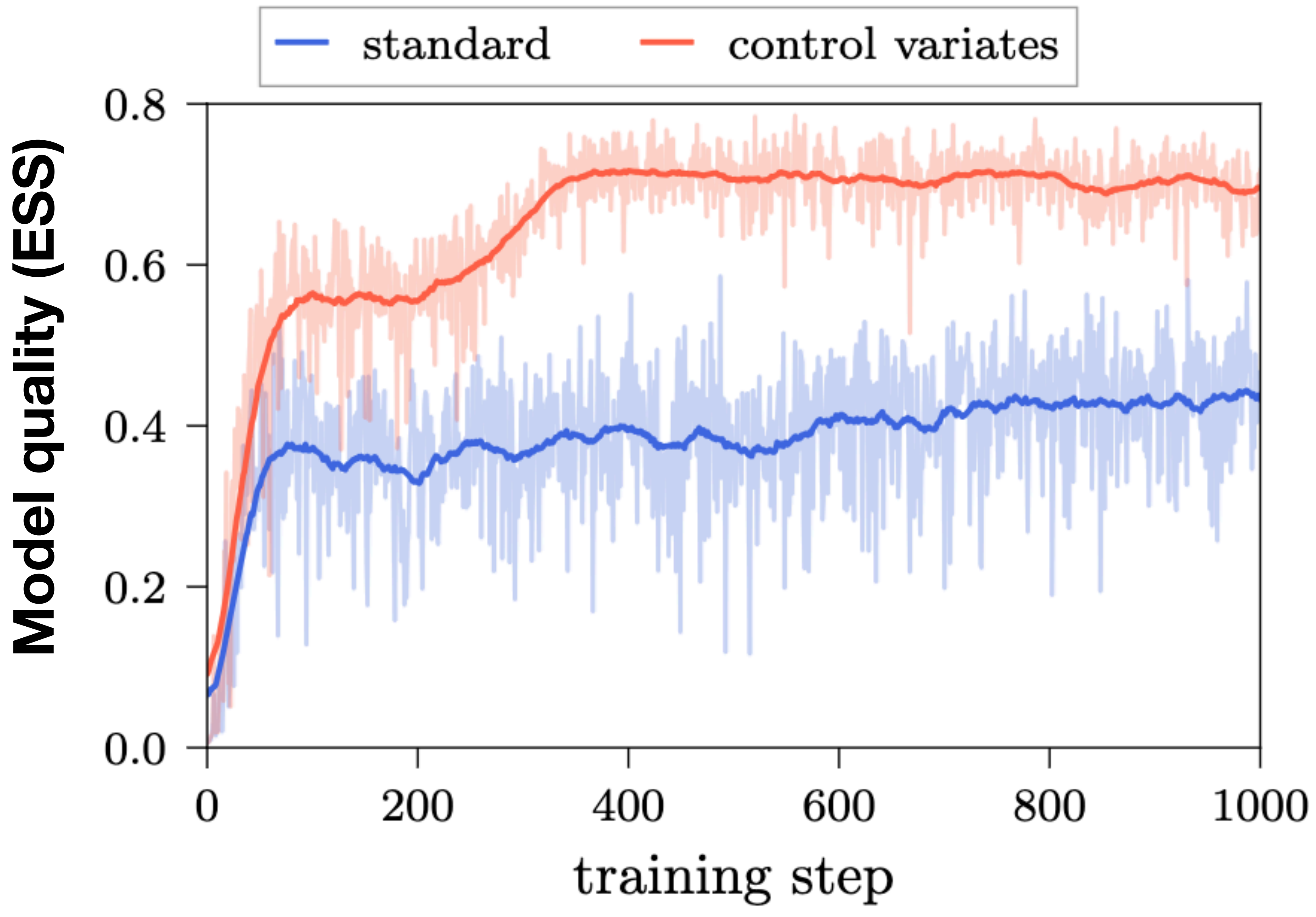


Cranmer, GK, Racanière, Rezende, Shanahan
Nature Reviews Physics 5 (2023) 526

Recent developments

- Better training procedures
  - Minimize gradient noise with control variates or path gradients

    Vaitl, Nicoli, Nakajima, Kessel (2022) 2207.08219



Abbott+ (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

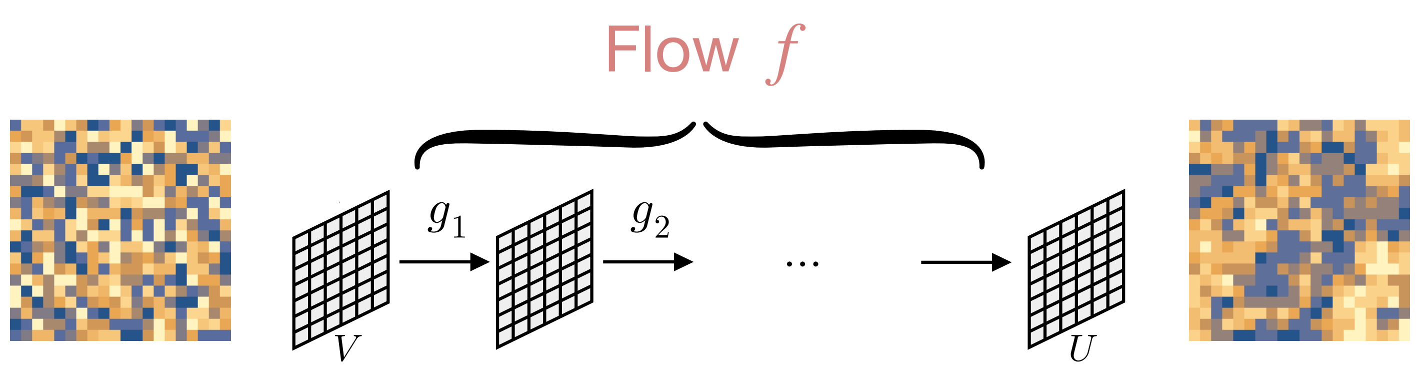
    Bacchio, Kessel, Schaefer, Vaitl PRD107 (2023) L051504

Conclusions

Machine learning methods show promise

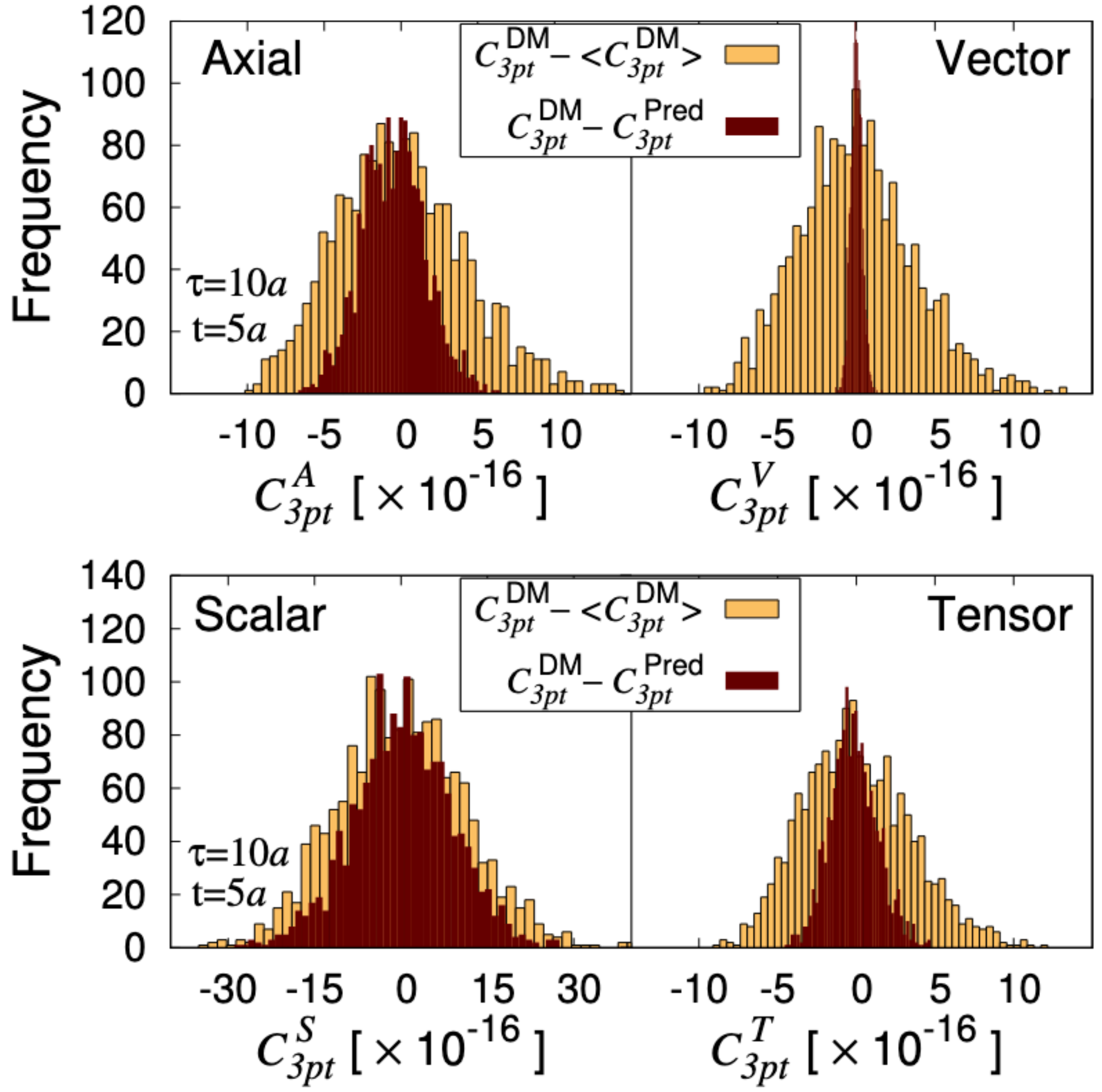
1. Ensemble generation

- Early success with flow-based generative models



2. Defining observables

- Order parameters, interpolating operators



3. Measuring observables

- Improved estimators, learned contour deformations

4. Analysis

- Challenging inverse problems, e.g. spectral functions

Albergo, GK, Shanahan PRD100 (2019) 034515
Yoon+ PRD100 (2019) 014504

Some general lessons

Exactness can often be encoded in physical applications

1. Analytical knowledge
2. Neural nets **inside** larger models

"Transfer learning" can be very useful

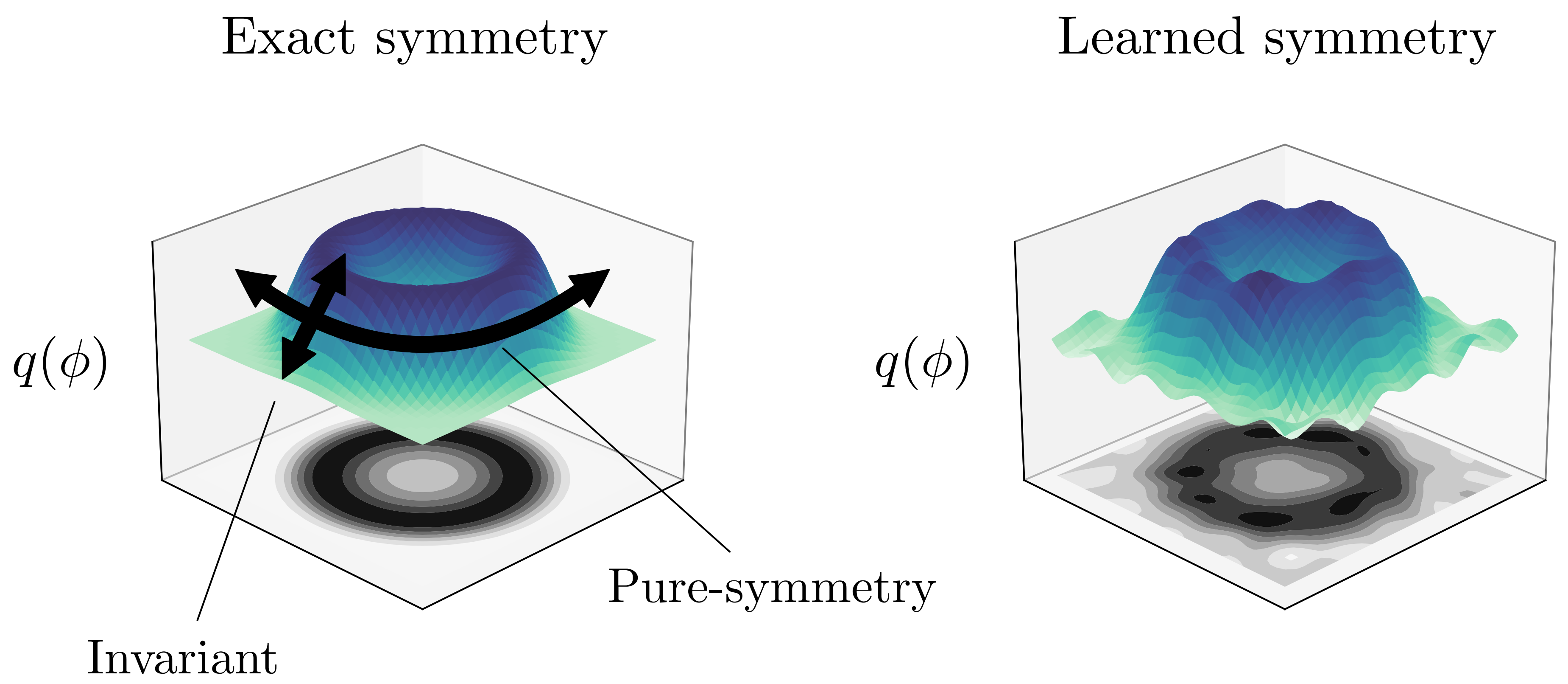
- Begin training from a related model
- Transfer between theories or tasks

Specialized models often necessary

- Symmetries found to improve efficiency

  GK, Albergo+ PRL125 (2020) 121601
  Albergo, GK+ PRD104 (2021) 114507
  Boyda, GK+ PRD103 (2021) 074504

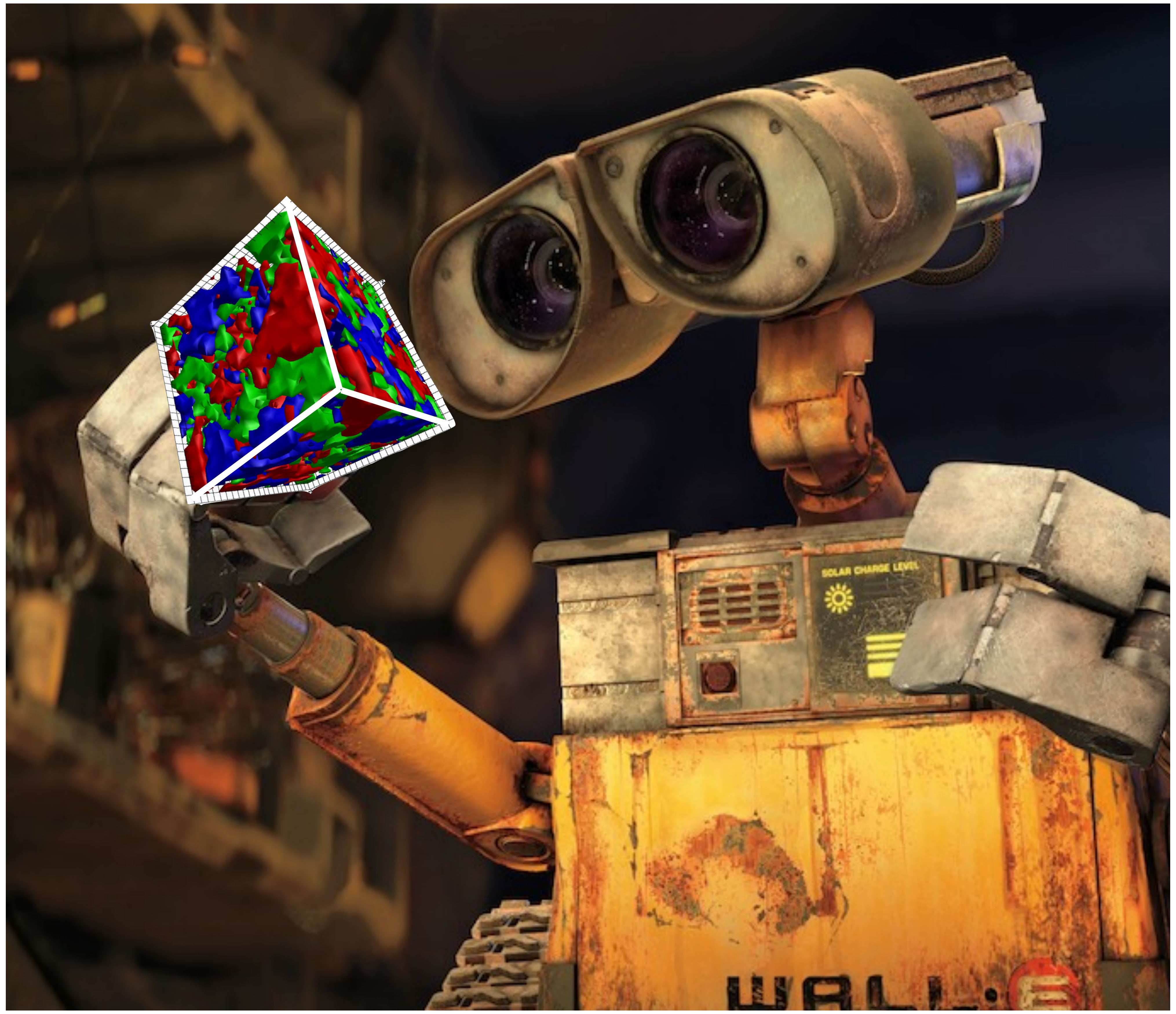
- Needed to handle structure of gauge manifold



Open questions

- Can we learn something intelligible from the trained models?
- Can generative approaches besides normalizing flows be made exact?
- Have we found a counter-example to the "Bitter lesson" or should we accept the conclusions of this theory?
- Can we exploit shared components of models between theories or applications? (Works very well for ChatGPT!)

Thank you!



Backup slides

Exactness

Samples from **model** are from biased distribution $q(U) \neq p(U)$, but…

For each U_i drawn from the model, we know $q(U_i)$ and $p(U_i)$

Flow-based models provide this.

Known in terms of the lattice action.

→

Exact bias correction possible
(e.g. "flow-based MCMC" or reweighting)

$$\langle \mathcal{O} \rangle_p = \frac{\langle \mathcal{O}(U)\, p(U)/q(U) \rangle_q}{\langle p(U)/q(U) \rangle_q}$$

Note: Efficiency of bias correction depends on how close q and p are.

RealNVP for scalar fields

Scalar field $\phi(x) \in \mathbb{R} \approx$ grayscale image

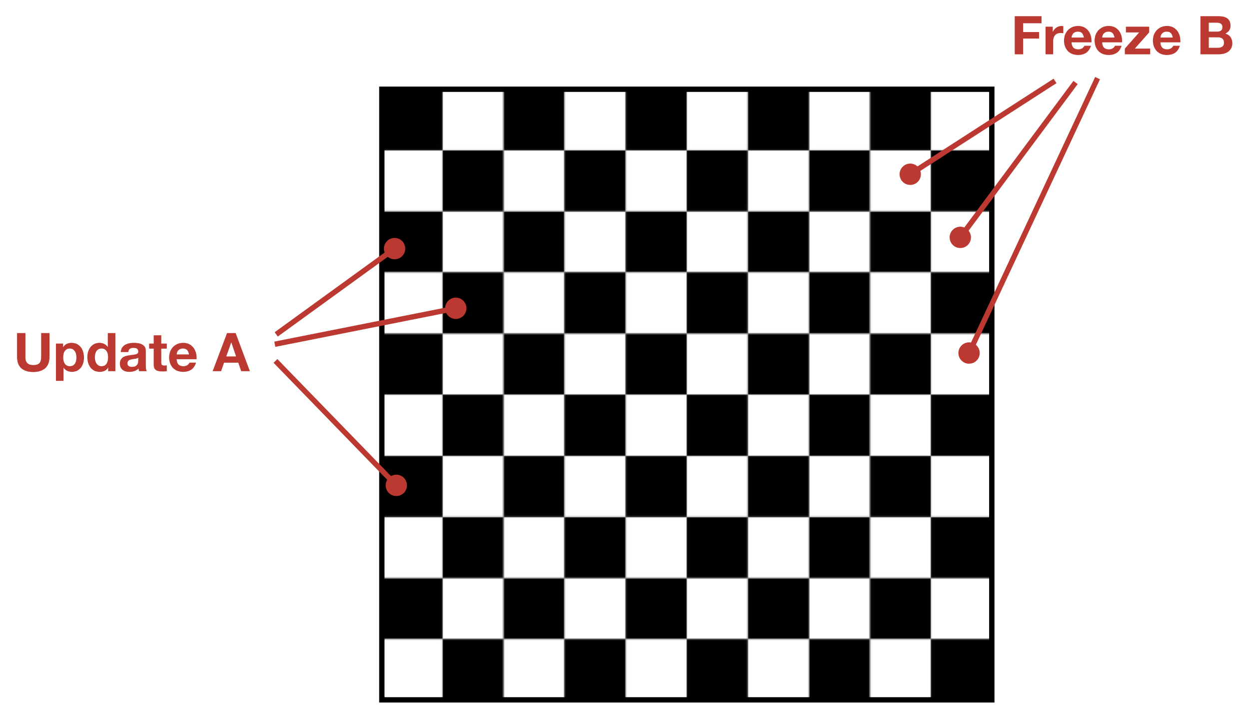
Tractable Jacobian

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

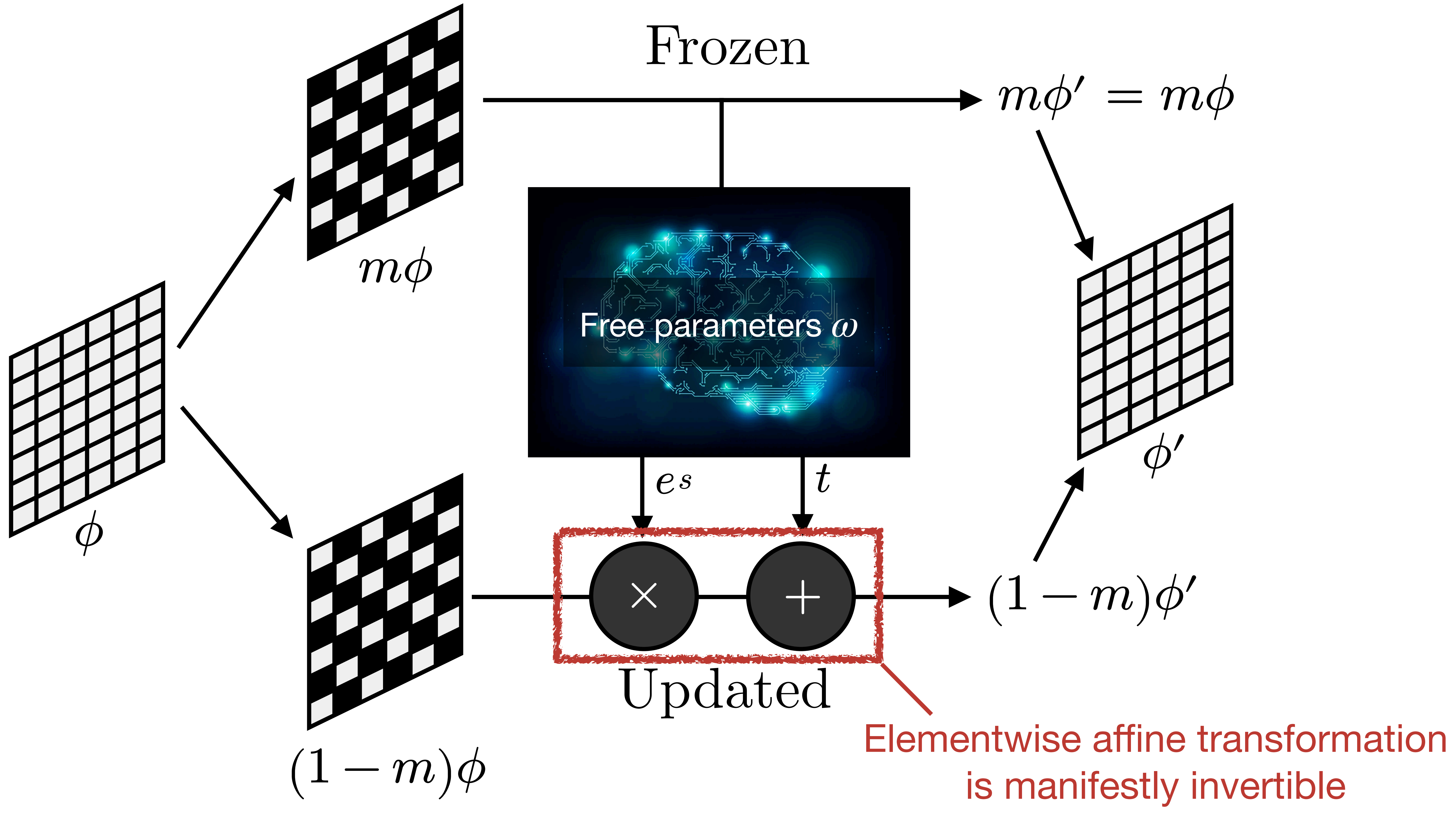
$$\implies \ln \det J = \sum_i s_i$$

Real NVP coupling layer:

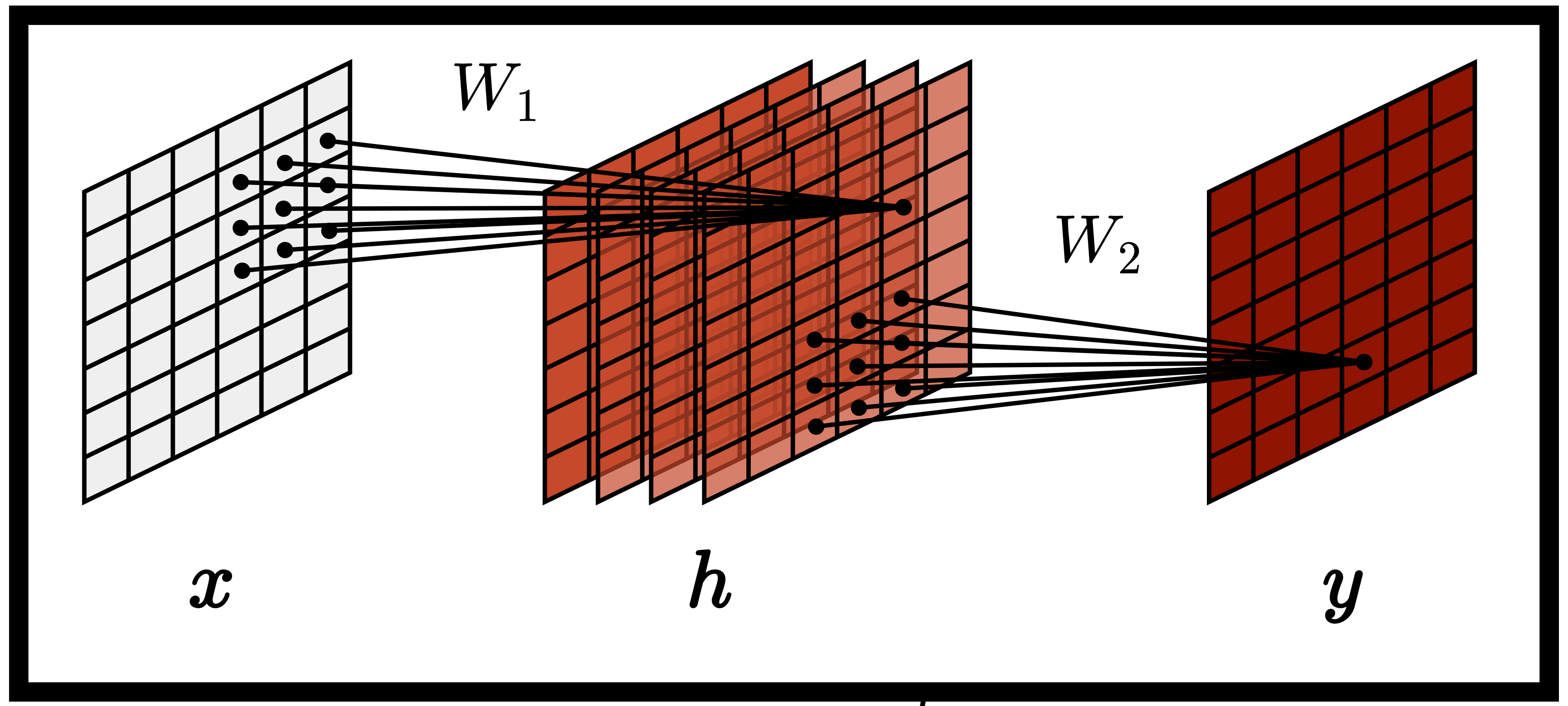
[Dinh, Sohl-Dickstein, Bengio **1605.08803**]



Checkerboard masking pattern m



Translational symmetry



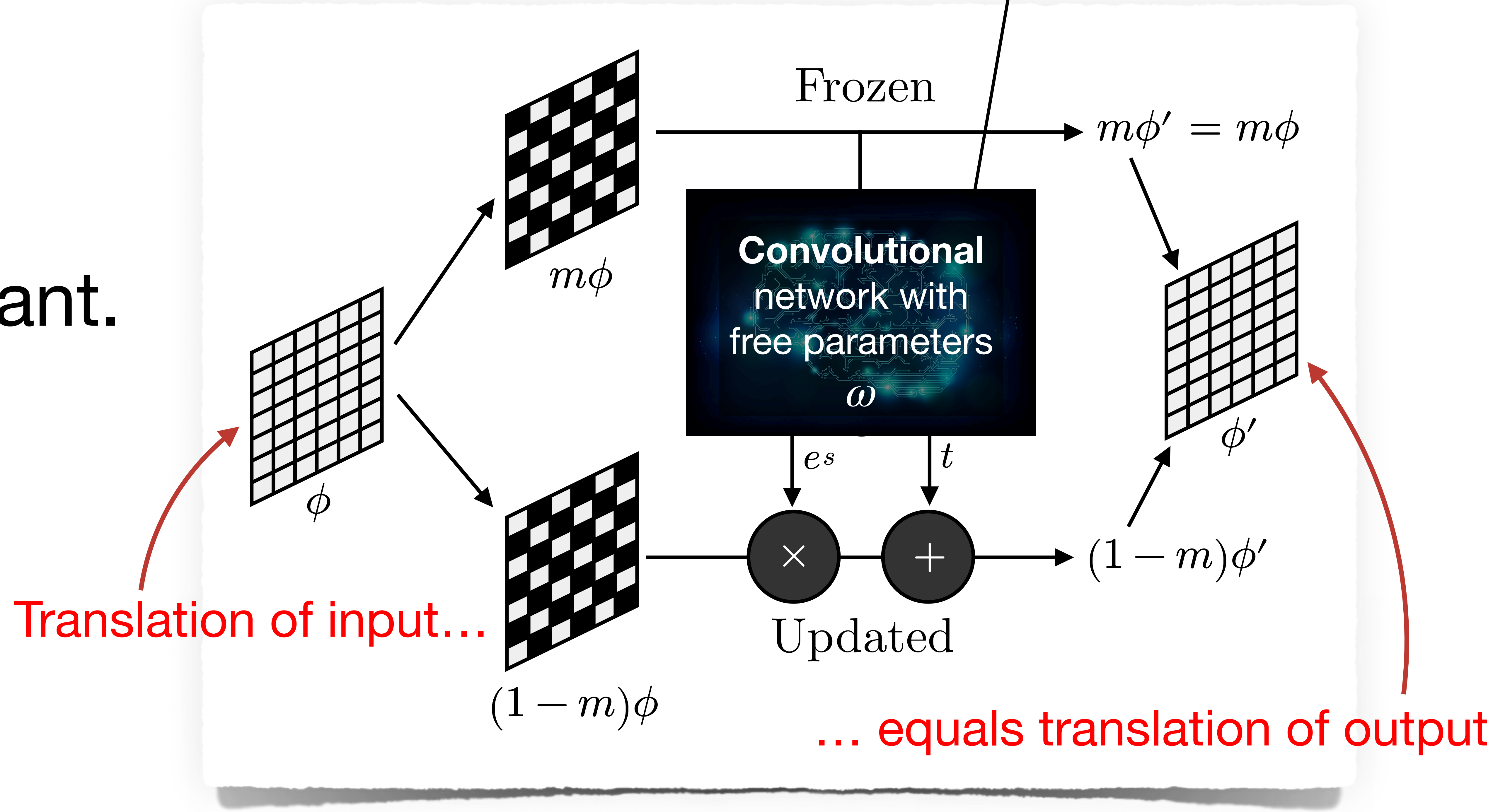
1. Use **Convolutional Neural Nets (CNNs).**

   - Output values (e.g. $e^{s(x)}$ and $t(x)$) for each site are local functions of frozen DoFs

   - CNNs are equivariant under translations

2. Make masking pattern (mostly) invariant.

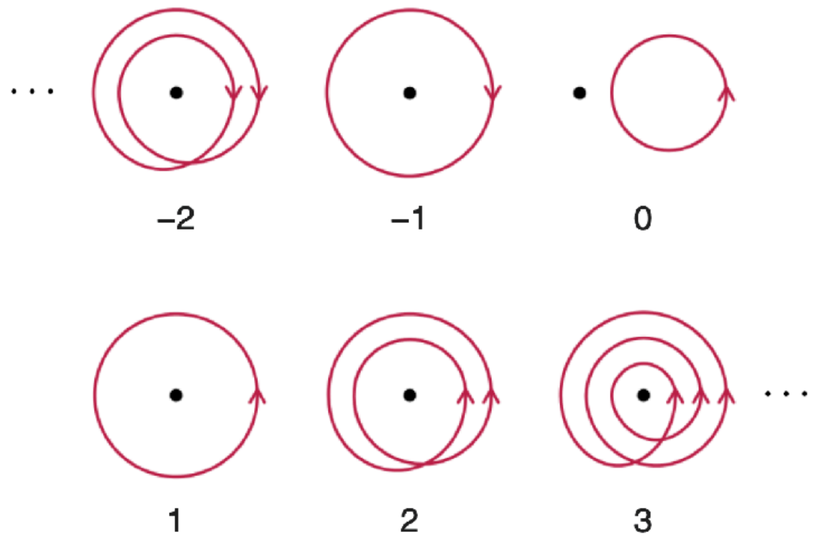
   - E.g. checkerboard

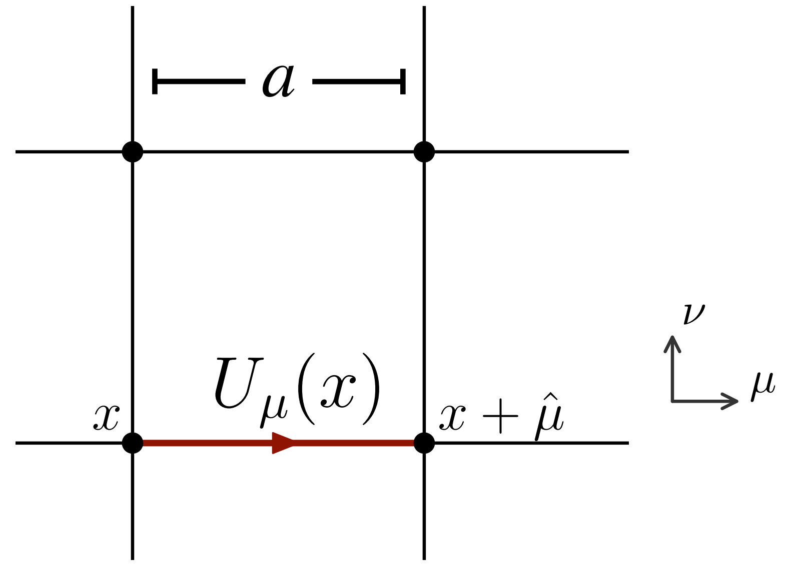


U(1) gauge theory in 1+1D

There is exact lattice topology in 2D.

$$Q = \frac{1}{2\pi} \sum_x \arg(P_{01}(x))$$



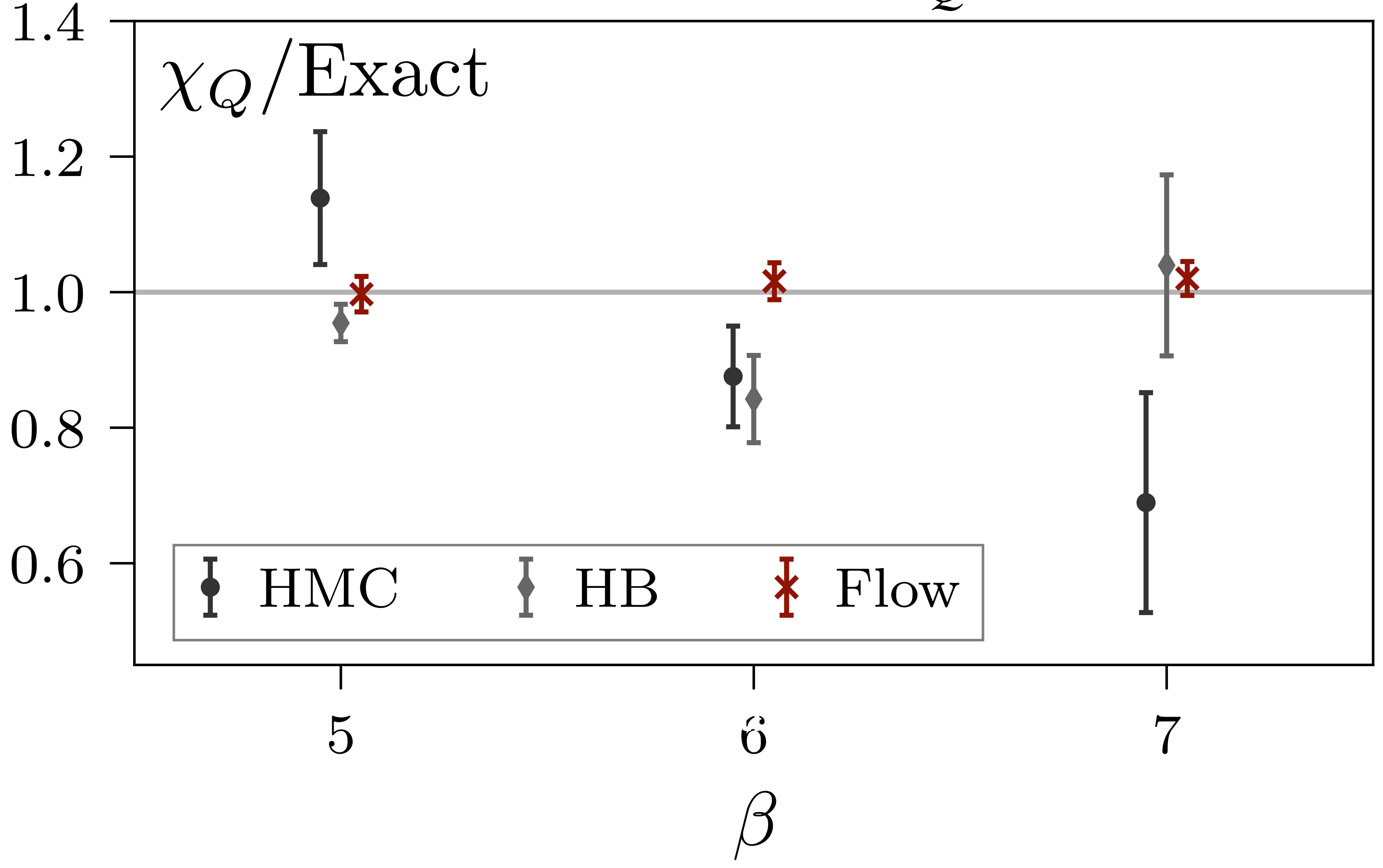


$$S(U) = -\beta \sum_x \sum_{\mu<\nu} \mathrm{Re}\, P_{\mu\nu}(x)$$

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

- Topological freezing towards continuum limit ($\beta \to \infty$)
- Compared **flow** vs **analytical, HMC,** and **heat bath** on 16×16 lattices for bare inverse coupling $\beta \in \{1,\ldots,7\}$
- One flow-based model trained for each β

Topological susceptibility $\chi_Q = \langle Q^2/V \rangle$



[GK, Albergo, Boyda, Cranmer, Hackett, Racanière, Rezende, Shanahan PRL125 (2020) 121601]