Flow-based MCMC for Lattice Ensemble Generation

(and progress towards the inclusion of fermions)

Gurtej Kanwar

Massachusetts Institute of Technology

Based on …

… flow-based sampling for lattice QFT:

[Albergo, GK, Shanahan **PRD100 (2019) 034515**]
[Albergo, Boyda, Hackett, GK, Cranmer, Racanière, Rezende, Shanahan **2101.08176**]
[Albergo, GK, Racanière, Rezende, Urban, Boyda, Cranmer, Hackett, Shanahan **2106.05934**]
[Hackett, Hsieh, Albergo, Boyda, Chen, Chen, Cranmer, GK, Shanahan **2107.00734**]

… flows for compact vars & lattice gauge theories:

[GK, Albergo, Boyda, Cranmer, Hackett, Racanière, Rezende, Shanahan **PRL125 (2020) 121601**]
[Rezende, Papamakarios, Racanière, Albergo, GK, Shanahan, Cranmer **ICML (2020) 2002.02428**]
[Boyda, GK, Racanière, Rezende, Albergo, Cranmer, Hackett, Shanahan **PRD103 (2021) 074504**]

Quark Confinement and the Hadron Spectrum 2021 | Virtual (Aug 2-6, 2021)

Importance sampling: the workhorse of LQFT

Monte Carlo sampling for efficient estimation of (many) observables

$$\boxed{\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \, \mathcal{O}(U) \, e^{-S(U)}}$$

Euclidean averages → equilibrium properties

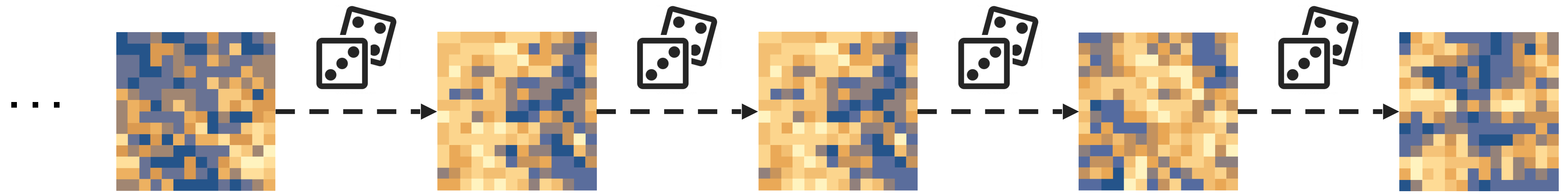
$$\langle \mathcal{O} \rangle \approx \frac{1}{n} \sum_{i=1}^{n} \mathcal{O}[U_i]$$

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

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

Markov chain Monte Carlo (MCMC)

- **Asymptotically** converges to distribution p
- However: States of the chain are “autocorrelated”
- Discard some thermalization steps, save states “thinned” to a subset with minimal correlations



Example: MCMC to generate samples of scalar field configurations

Importance sampling: the workhorse of LQFT

Monte Carlo sampling for efficient estimation of (many) observables

$$\boxed{\langle \mathcal{O} \rangle = \frac{1}{Z} \int \mathcal{D}U \, \mathcal{O}(U) \, e^{-S(U)}}$$

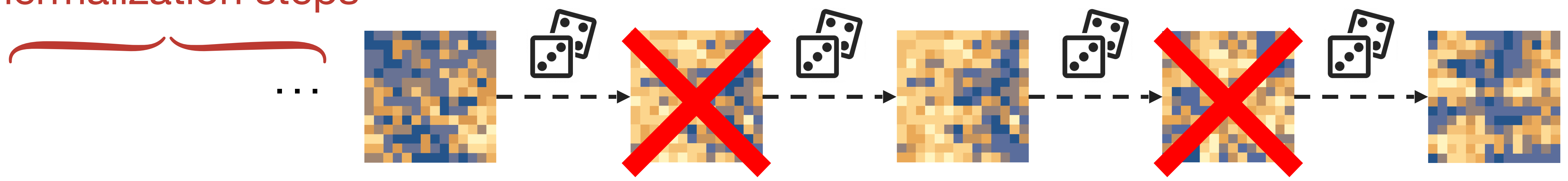
Euclidean averages → equilibrium properties

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

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

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

Markov chain Monte Carlo (MCMC)



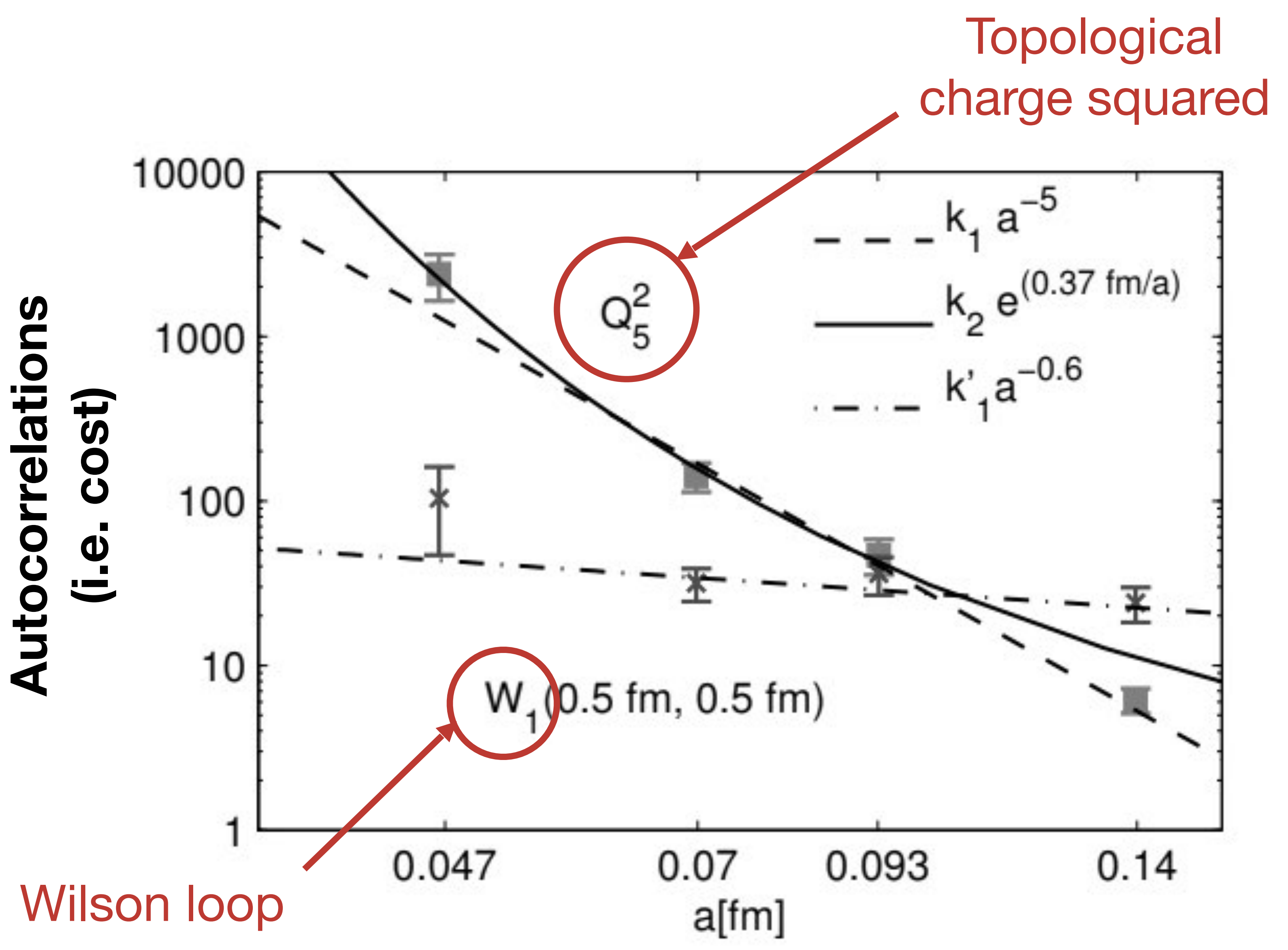
Example: MCMC to generate samples of scalar field configurations

- **Asymptotically** converges to distribution p
- However: States of the chain are “autocorrelated”
- Discard some thermalization steps, save states “thinned” to a subset with minimal correlations

Motivations for applying ML

Critical slowing down and **topological freezing** obstruct MCMC sampling near the continuum limit.

- **Problem:** Local/diffusive Markov chain updates
- Generative ML models can directly sample, may be used to propose global updates



[Schaefer et al. / ALPHA collaboration **NPB845 (2011) 93**]

Generative models provide flexible "variational ansatz" distribution $q(U)$.

After optimizing the model "ansatz":

$$q(U) = e^{-S_{\text{eff}}(U)} \approx p(U) = e^{-S(U) - \log Z}$$

$$\updownarrow$$

$$S_{\text{eff}}(U) \approx S(U) + \log Z$$

Efficiently sampled

Desired target

ML modeling for LQFT

[See also: N. Gerasimenuik, next talk]

Estimating thermodynamic observables:

- Flow-based models precisely estimate $\log Z$
- Asymptotic exactness $N \to \infty$

[A. Tomiya, Fri]

Improved HMC updates:

- NNs describing field transformations
- HMC updates using modified action / fields
- Exactness: Metropolis (true action)

This talk

Flow-based MCMC:

- Flows directly propose new configs
- Metropolis step (satisfying balance) for exactness

[B. Yoon, previous talk]

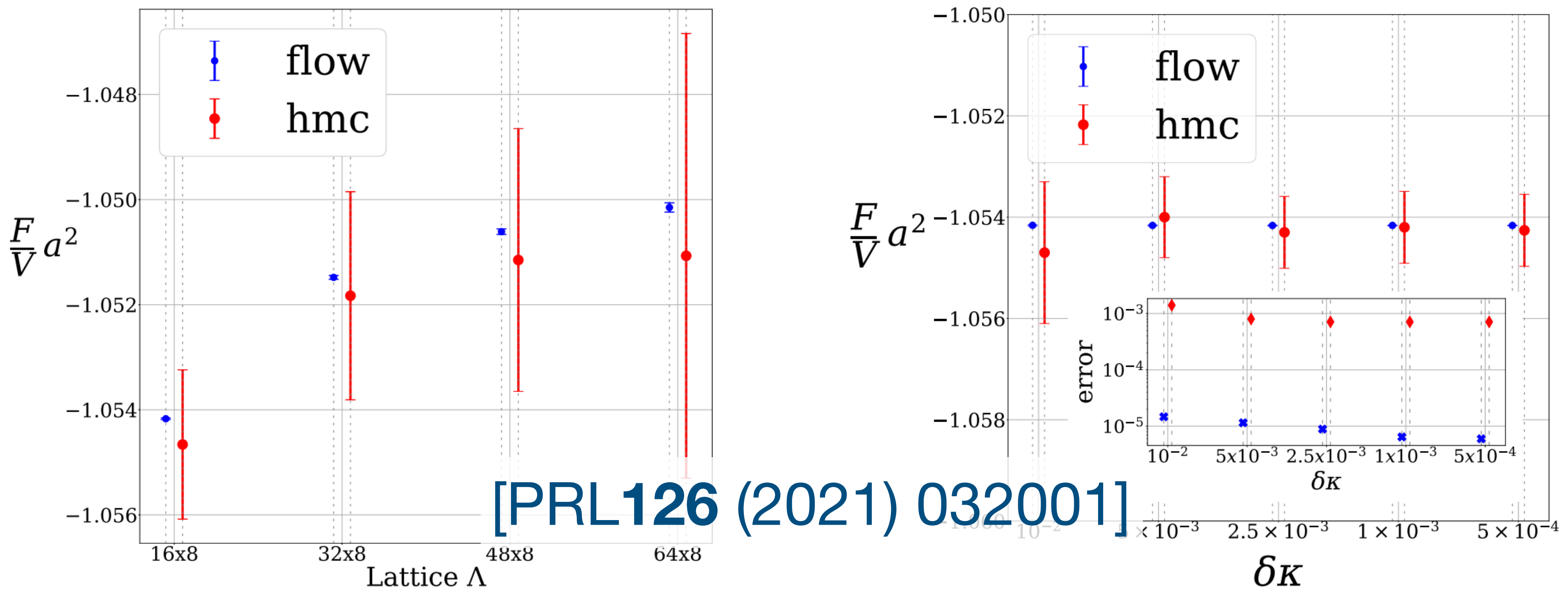
Improved MC estimators:

- ML regression (efficient approx. estimators)
- Exactness via bias correction term

ML modeling for LQFT

[See also: N. Gerasimenuik, next talk]

Estimating thermodynamic observables:



[PRL**126** (2021) 032001]

[A. Tomiya, Fri]

Improved HMC updates:

- NNs describing field transformations
- HMC updates using modified action / fields
- Exactness: Metropolis (true action)

This talk

Flow-based MCMC:

- Flows directly propose new configs
- Metropolis step (satisfying balance) for exactness

[B. Yoon, previous talk]

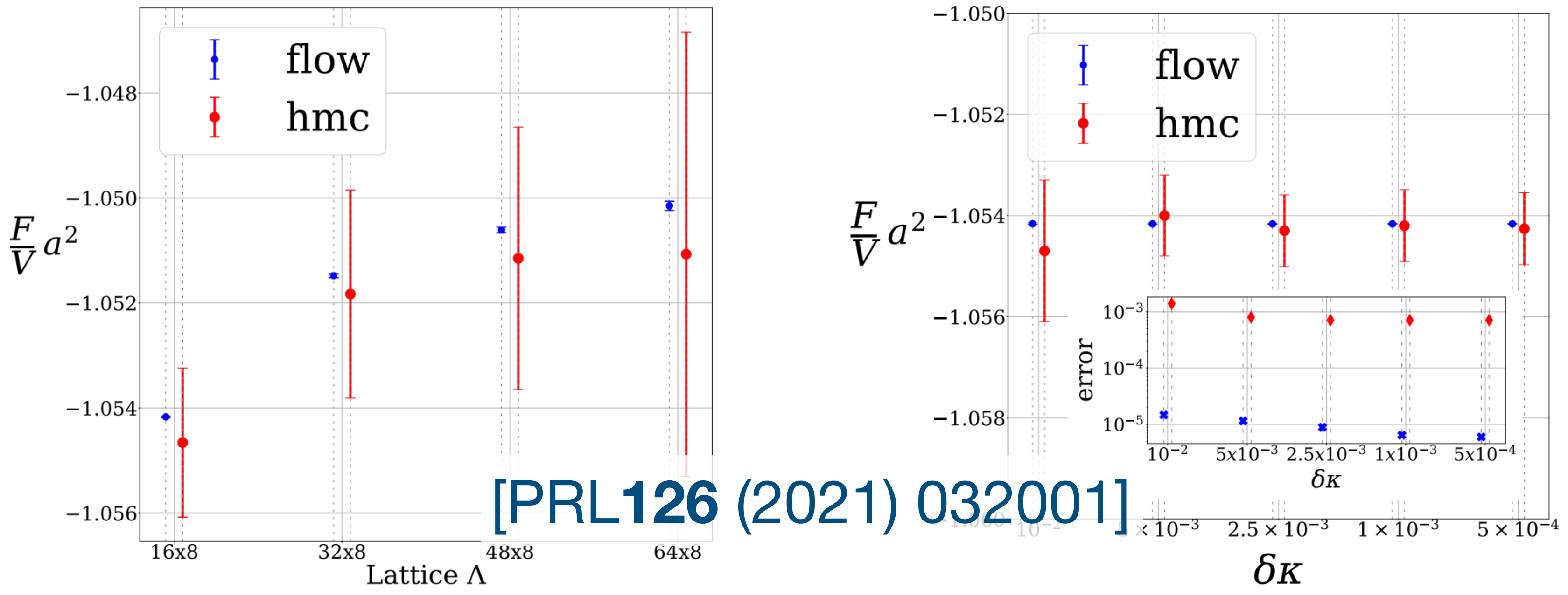
Improved MC estimators:

- ML regression (efficient approx. estimators)
- Exactness via bias correction term

ML modeling for LQFT

[See also: N. Gerasimenuik, next talk]

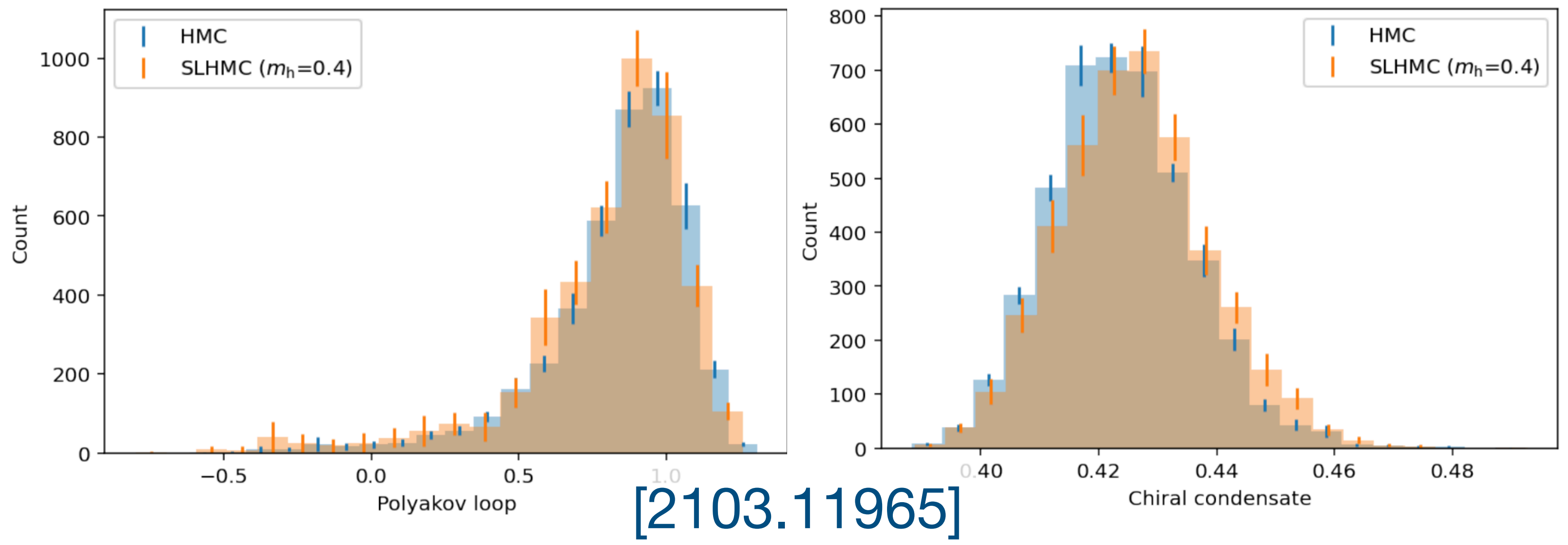
Estimating thermodynamic observables:



[PRL**126** (2021) 032001]

[A. Tomiya, Fri]

Improved HMC updates:



[2103.11965]

This talk

Flow-based MCMC:

- Flows directly propose new configs
- Metropolis step (satisfying balance) for exactness

[B. Yoon, previous talk]

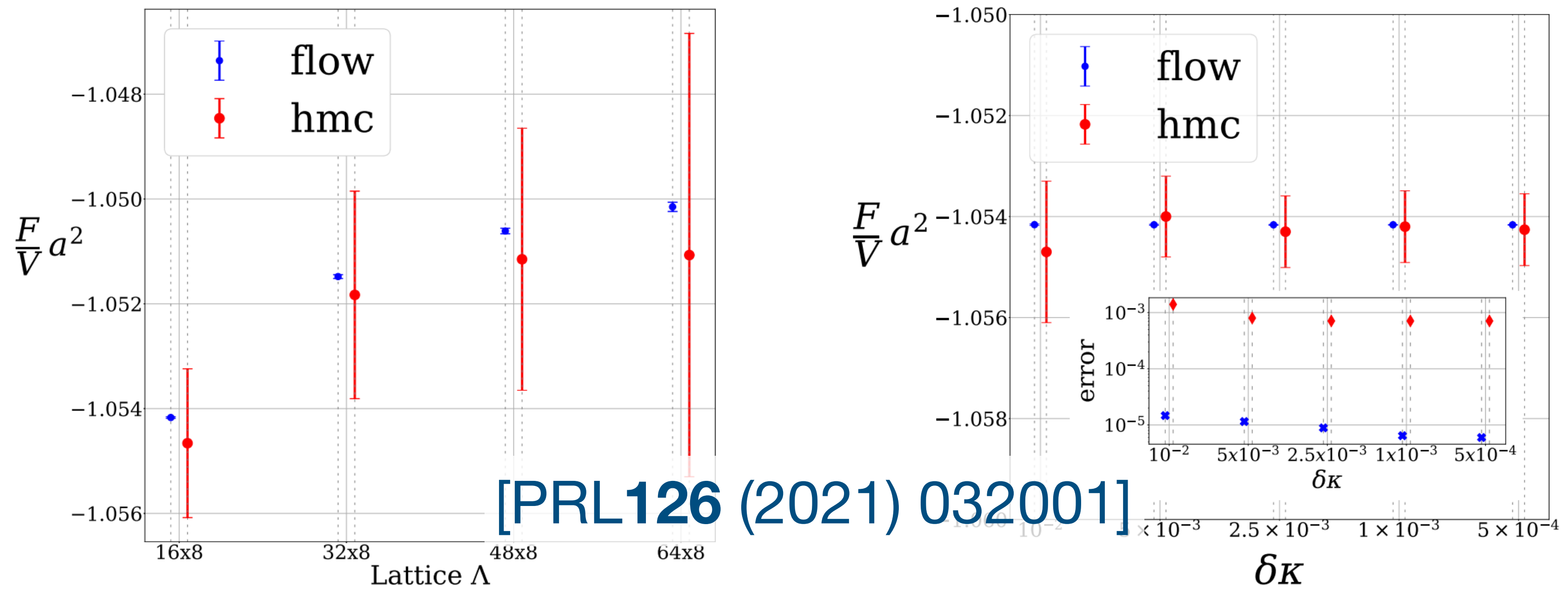
Improved MC estimators:

- ML regression (efficient approx. estimators)
- Exactness via bias correction term

ML modeling for LQFT

[See also: N. Gerasimenuik, next talk]

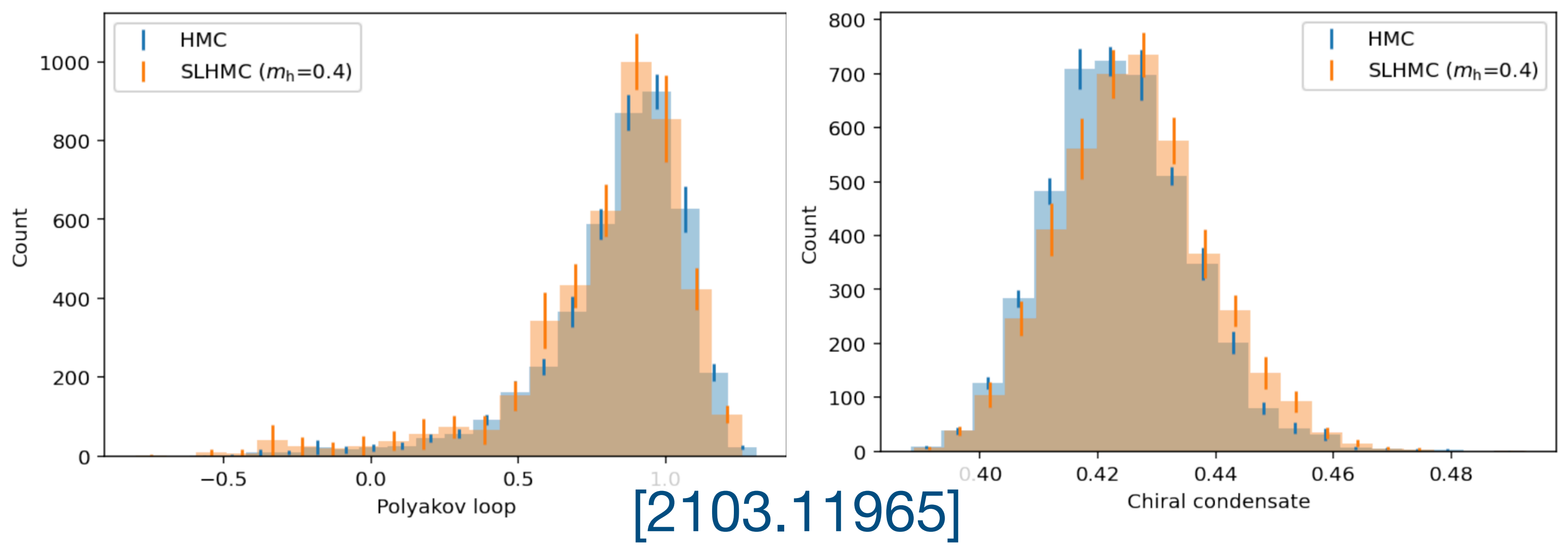
Estimating thermodynamic observables:



[PRL**126** (2021) 032001]

[A. Tomiya, Fri]

Improved HMC updates:



[2103.11965]

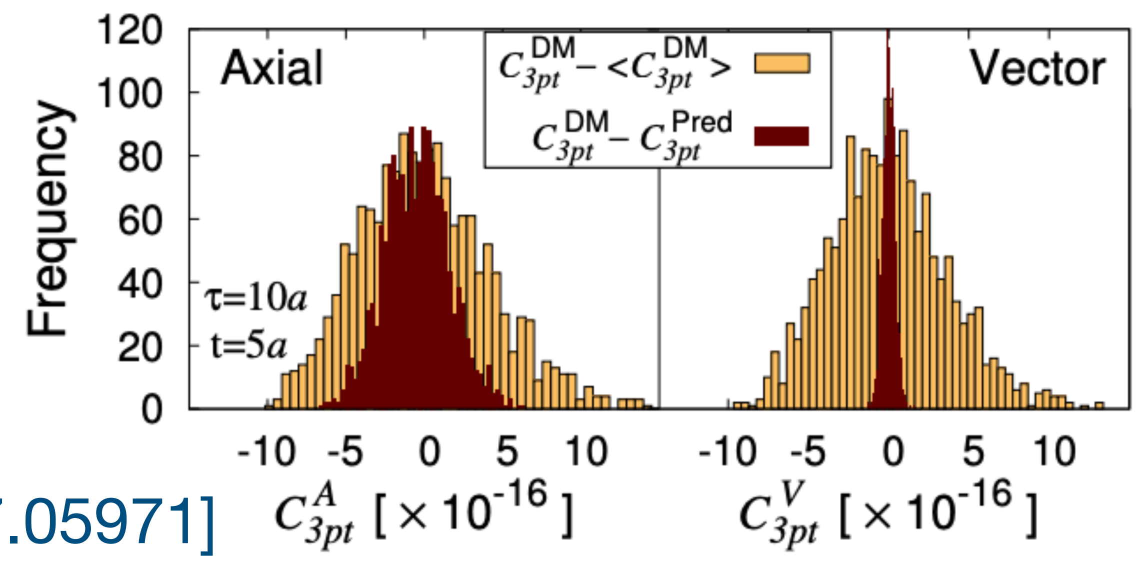
This talk

Flow-based MCMC:

- Flows directly propose new configs
- Metropolis step (satisfying balance) for exactness

[B. Yoon, previous talk]

Improved MC estimators:

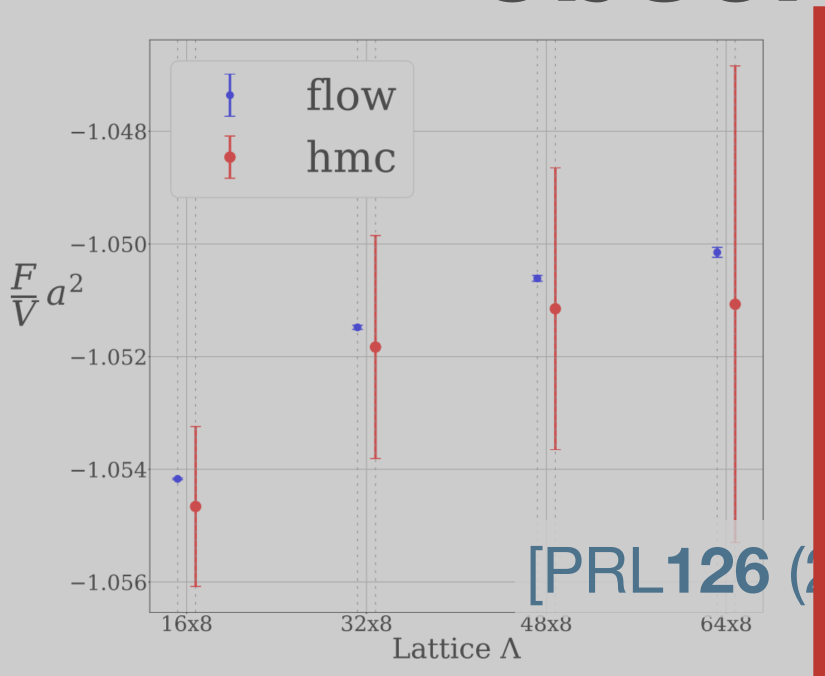


[1807.05971]

ML modeling for LQFT

[See also: N. Gerasimenuik, next talk]

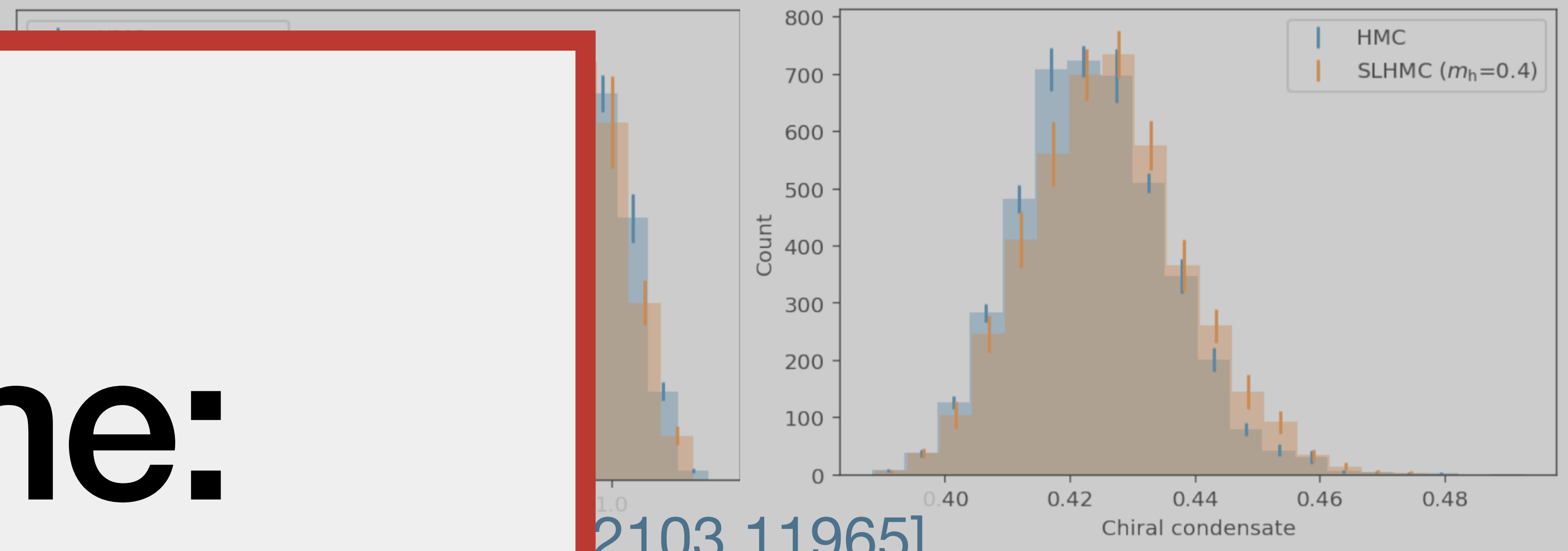
Estimating thermodynamic observables:



[PRL126 (

[A. Tomiya, Fri]

Improved HMC updates:



2103.11965]

Common theme:

Black-box ML components wrapped inside exact schemes

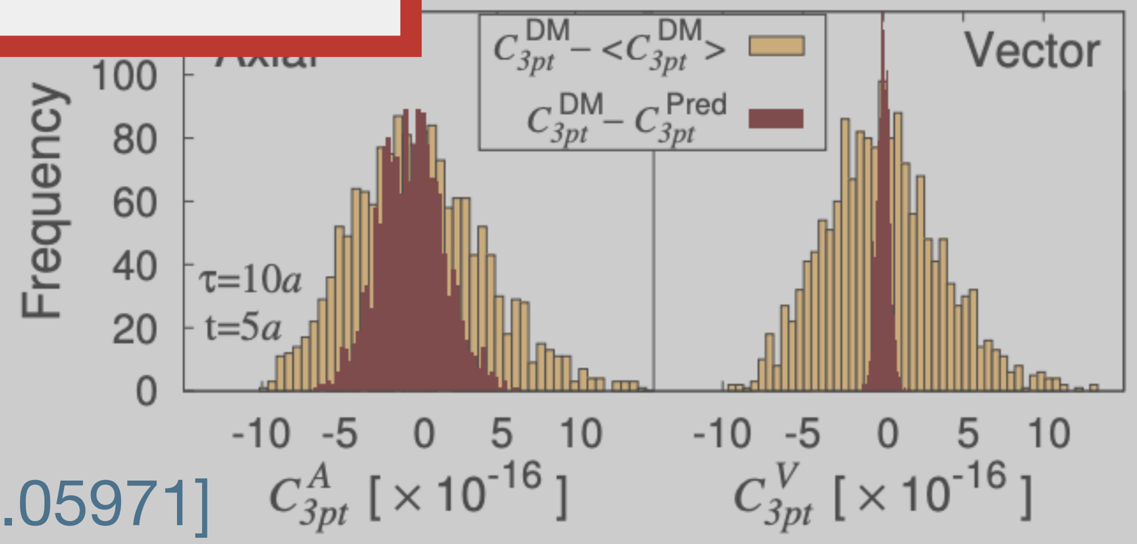
This talk

Flow-bas

- Flows directly propose new configs
- Metropolis step (satisfying balance) for exactness

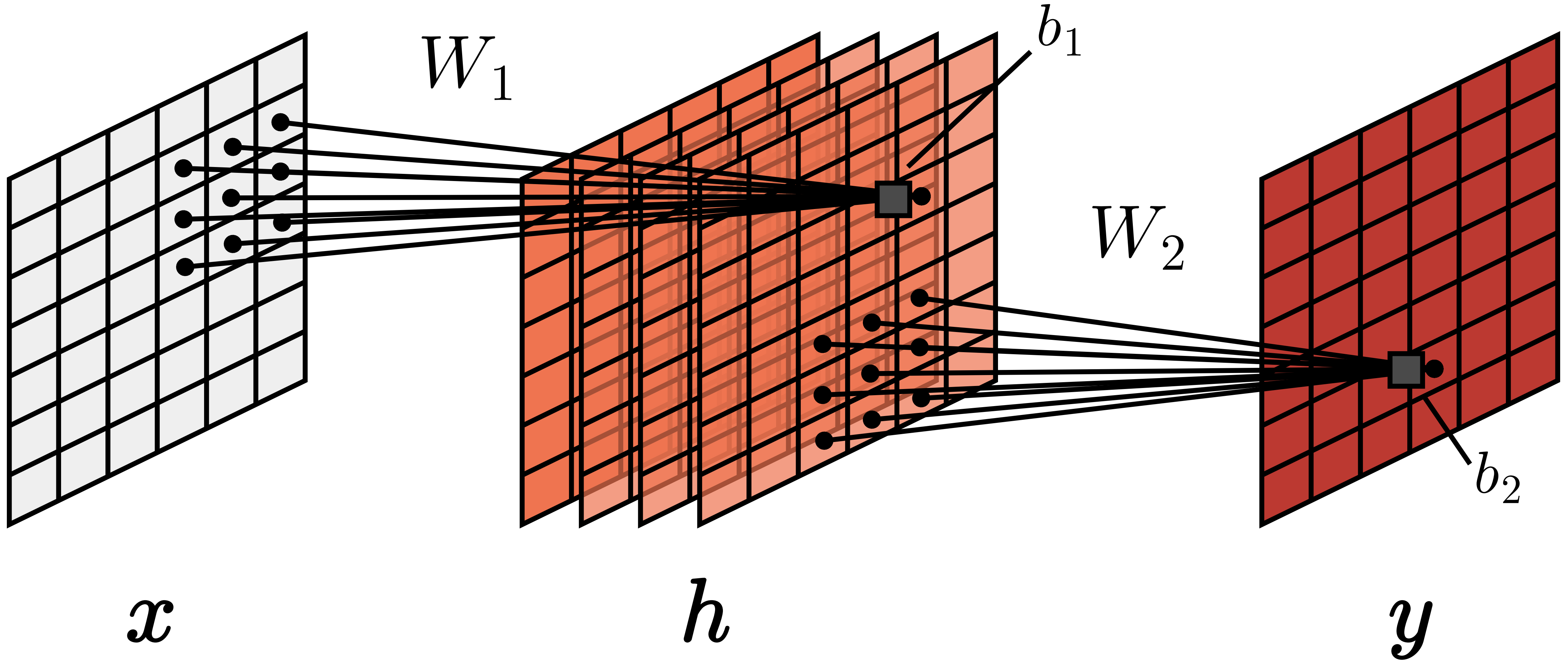
[B. Yoon, previous talk]

MC estimators:



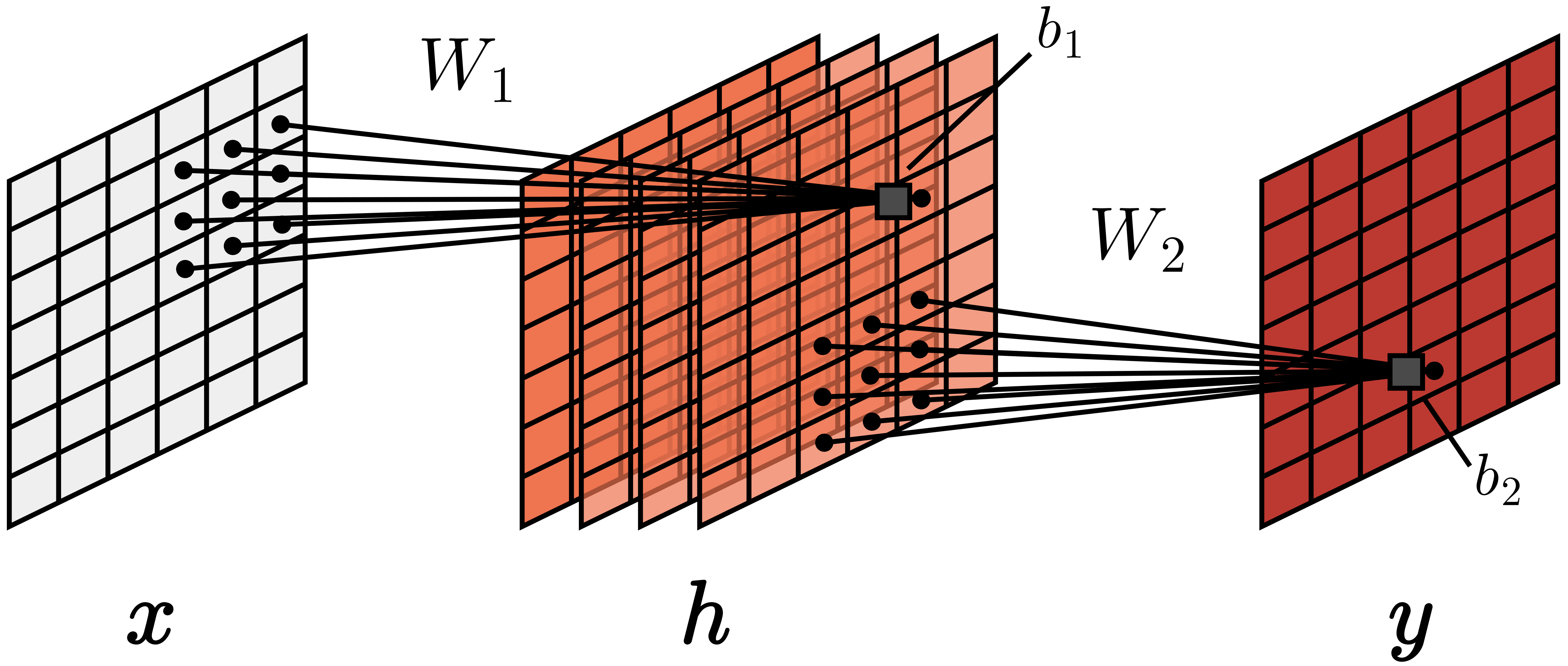
[1807.05971]

Flow-based sampling: Overview



Flow-based sampling: Overview

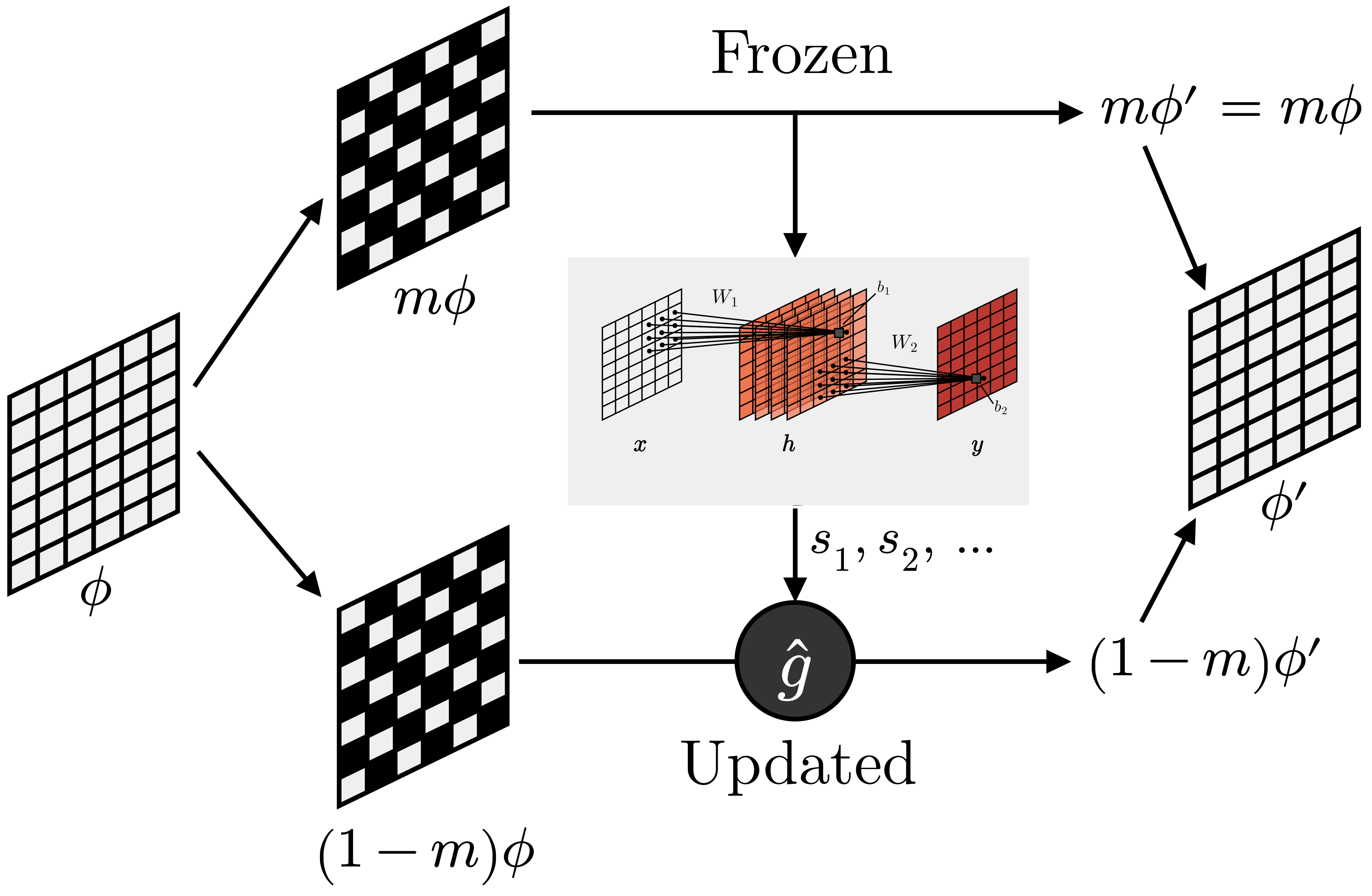
(Convolutional) neural networks: Black-box (local) function approximators



Flow-based sampling: Overview

(Convolutional) neural networks: Black-box (local) function approximators

Coupling layers: Invertible transformations, tractable Jacobian

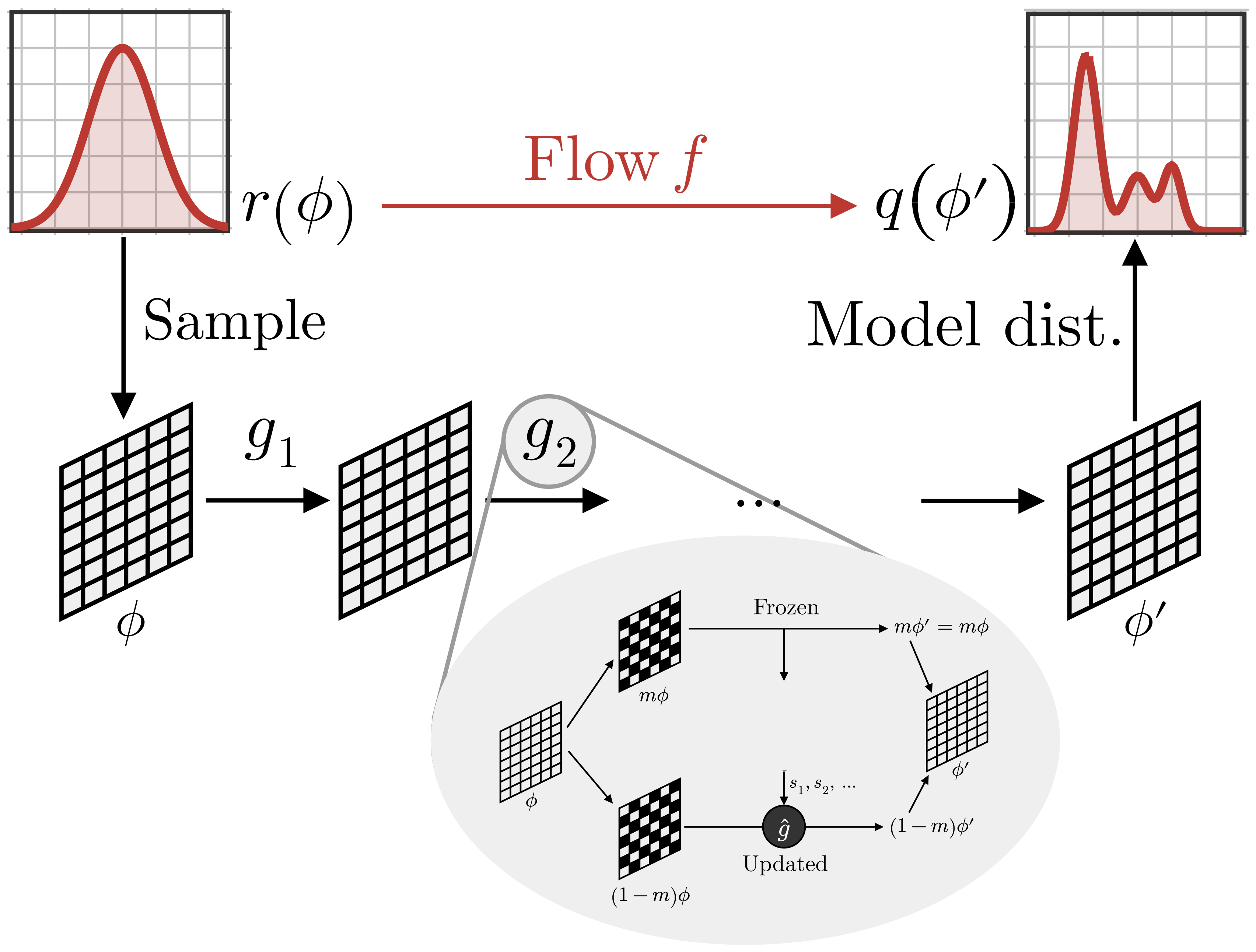


Flow-based sampling: Overview

(Convolutional) neural networks: Black-box (local) function approximators

Coupling layers: Invertible transformations, tractable Jacobian

Flow-based model: Transform prior density to **computable** and **sample-able** output model density



$$\boxed{q(\phi') = r(\phi) \left| \det_{ij} \frac{\partial [f(\phi)]_i}{\partial \phi_j} \right|^{-1}}$$

Flow-based sampling: Overview

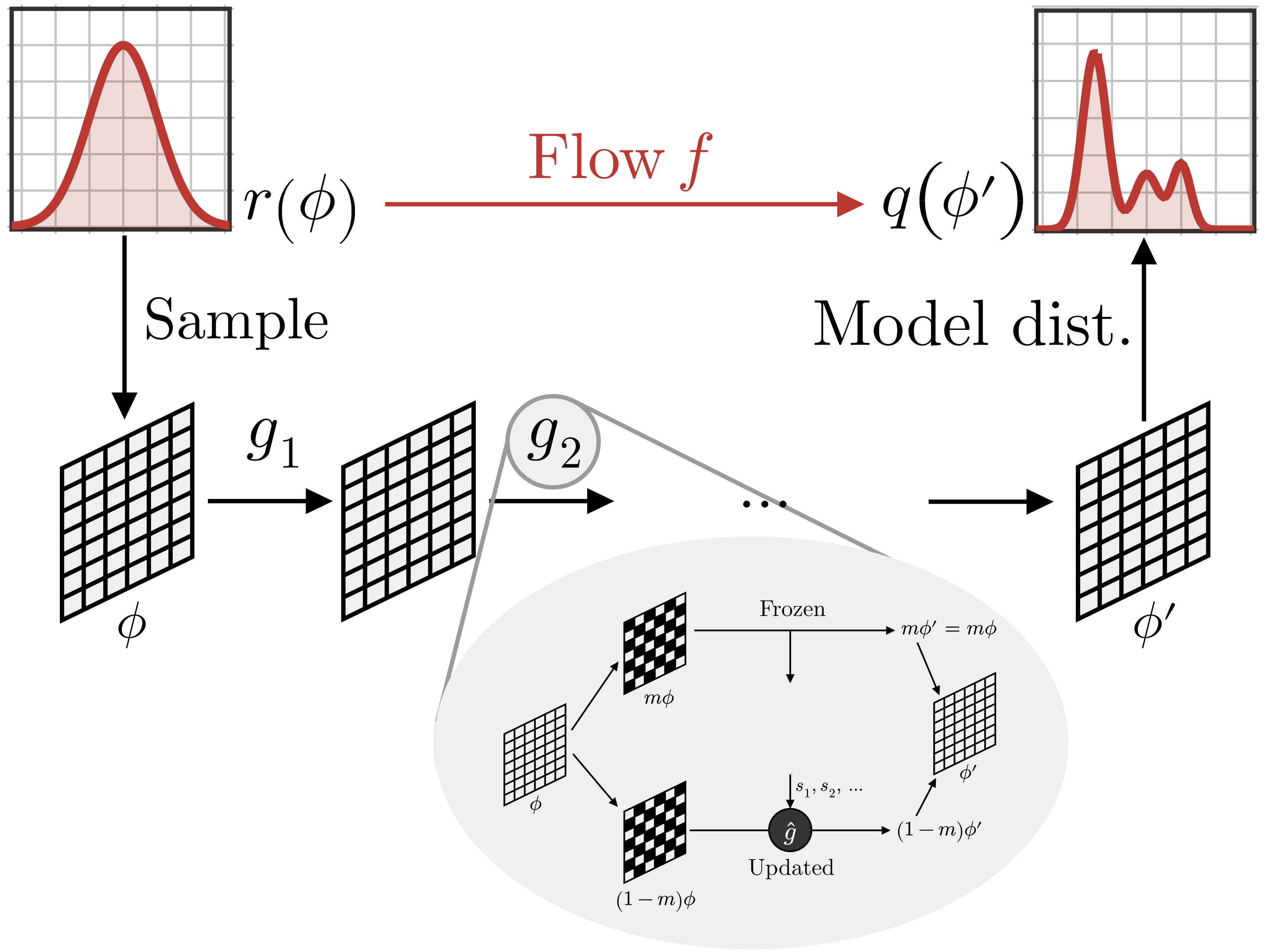
(Convolutional) neural networks: Black-box (local) function approximators

Coupling layers: Invertible transformations, tractable Jacobian

Flow-based model: Transform prior density to **computable** and **sample-able** output model density

Training:

- Measure KL divergence
- Apply gradient-based opt



$$q(\phi') = r(\phi) \left| \det_{ij} \frac{\partial [f(\phi)]_i}{\partial \phi_j} \right|^{-1}$$

Flow-based sampling: Overview

(Convolutional) neural networks: Black-box (local) function approximators

Coupling layers: Invertible transformations, tractable Jacobian

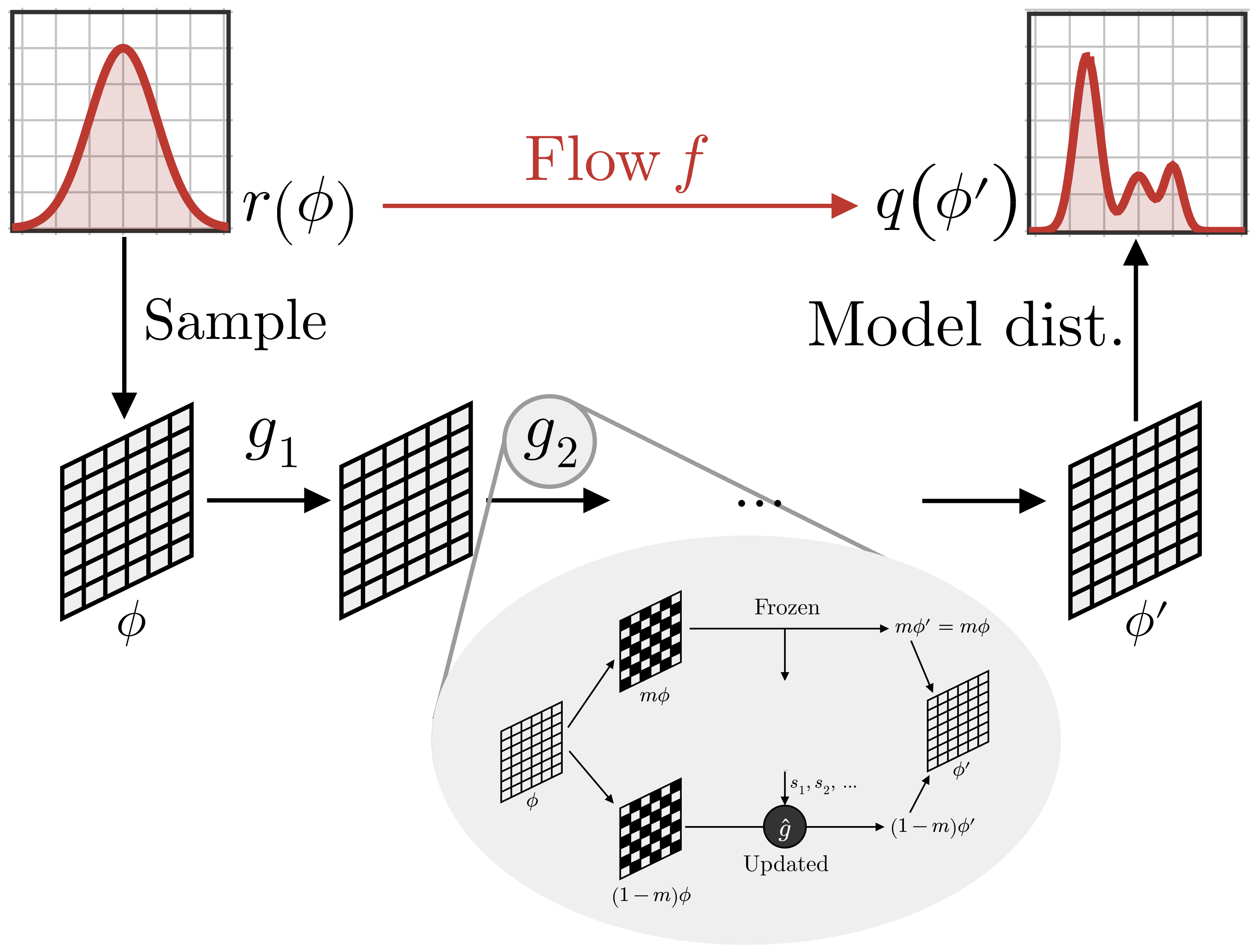
Flow-based model: Transform prior density to **computable** and **sample-able** output model density

Training:

- Measure KL divergence
- Apply gradient-based opt

Exactness:

- Use $q(\phi')$ and $p(\phi')$to correct approximation



$$q(\phi') = r(\phi)\left|\det_{ij}\frac{\partial[f(\phi)]_i}{\partial\phi_j}\right|^{-1}$$

Coupling layers

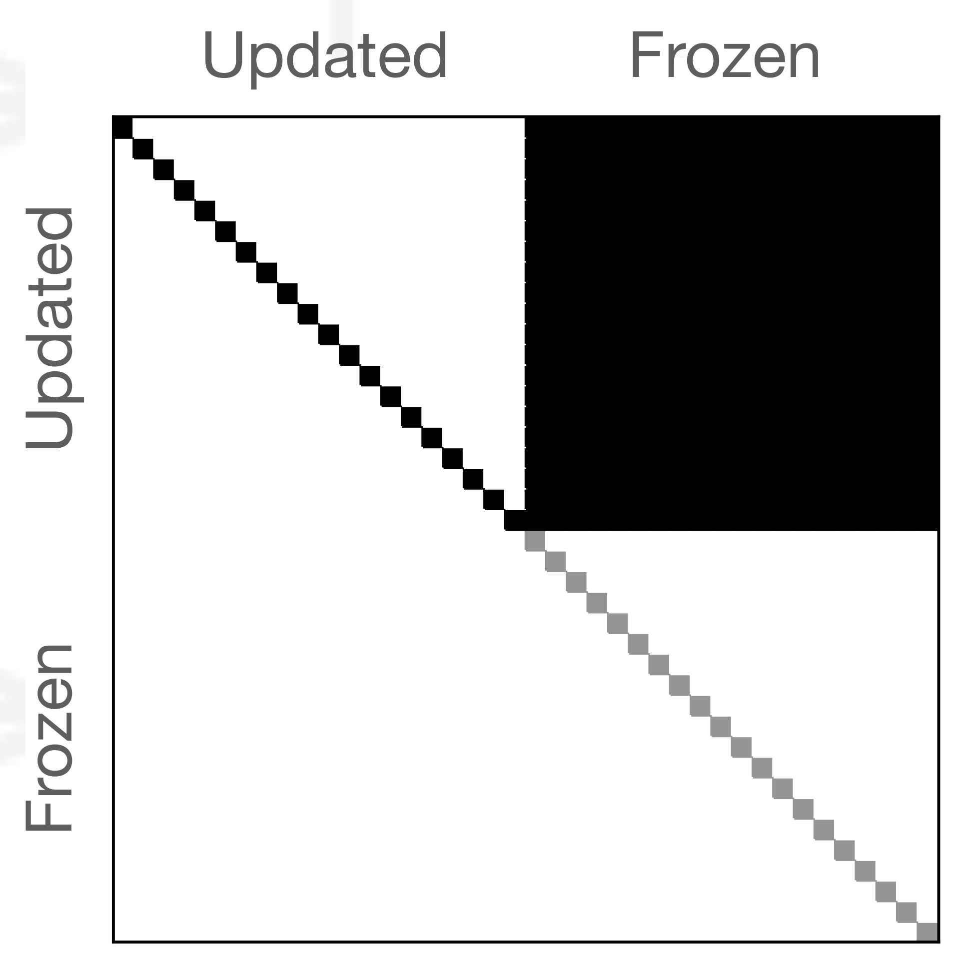
Similar to leapfrog integrator

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 LDJ from diag elts)

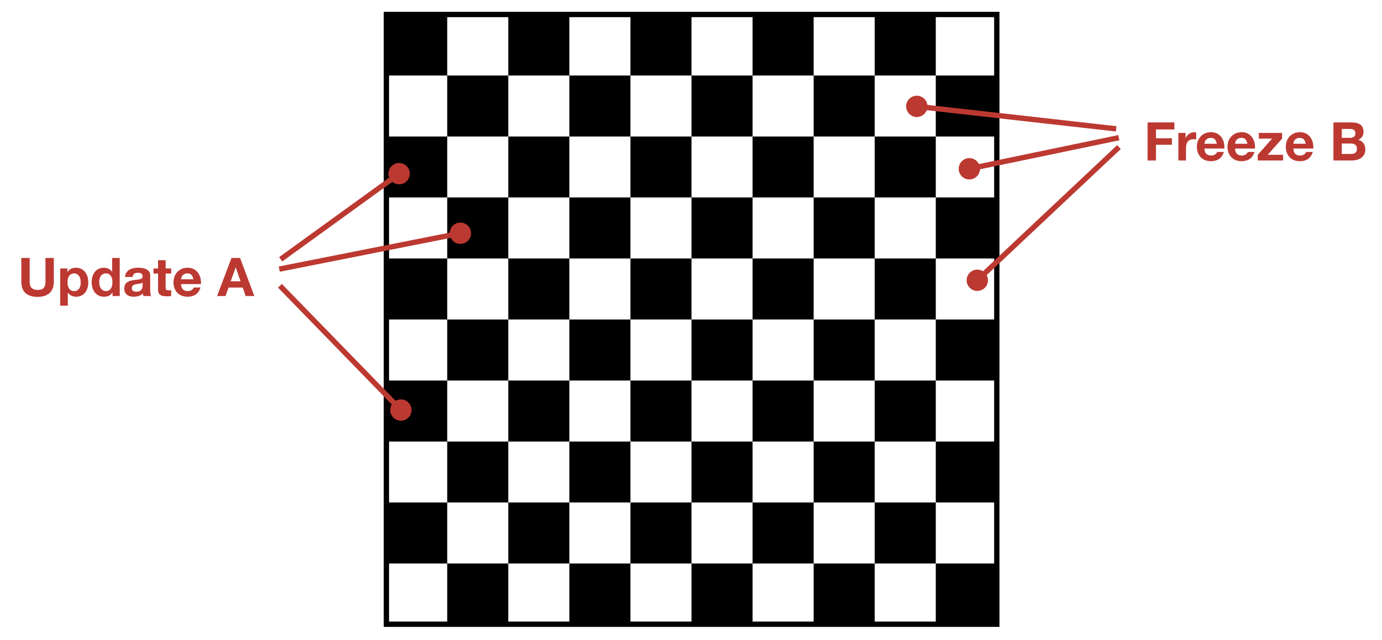
$$\begin{array}{ccc} & \text{“Updated” } i\ (m_i = 0) & \text{“Frozen” } i\ (m_i = 1) \\ \dfrac{\partial[g(V)]_i}{\partial V_j} = & \multicolumn{2}{c}{\left(\begin{array}{ccc|ccc} \frac{\partial[g(V)]_1}{\partial V_1} & & & & & \\ & \frac{\partial[g(V)]_2}{\partial V_2} & & & \text{(nonzero)} & \\ & & \ddots & & & \\ \hline & & & 1 & & \\ & 0 & & & 1 & \\ & & & & & \ddots \end{array}\right)} \end{array}$$

Schematically



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

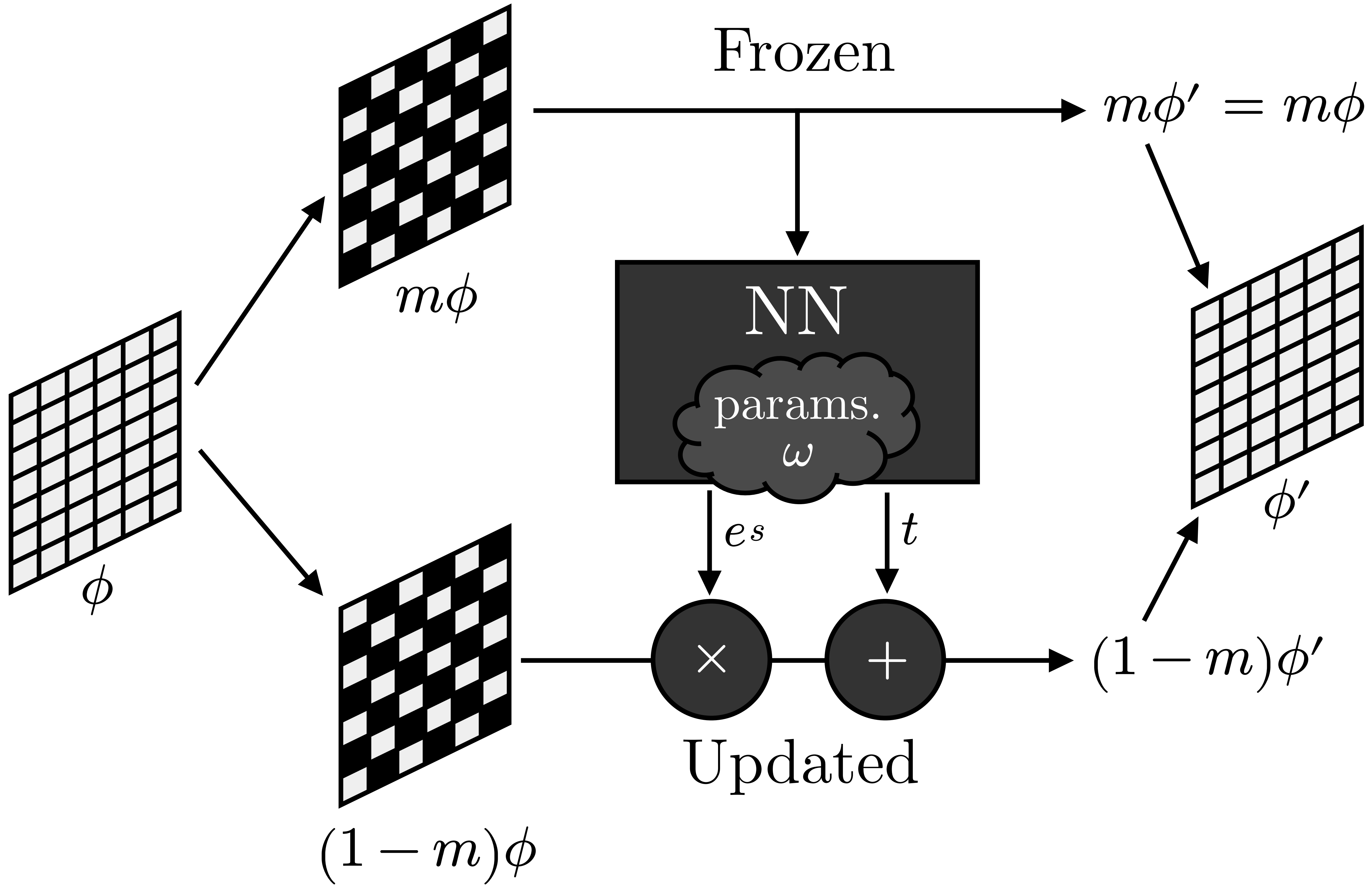
Ex: RNVP for scalar fields



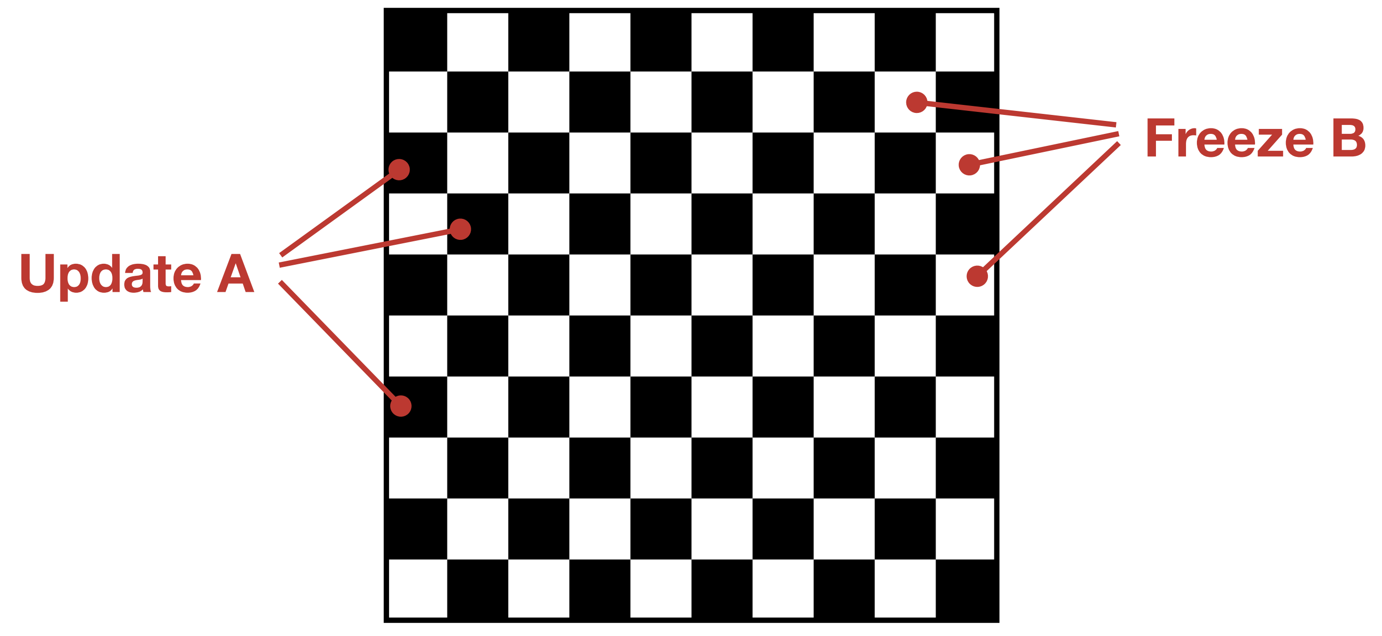
Checkerboard masking pattern m

Real scalar field $\phi(x) \in \mathbb{R} \;\approx\;$ grayscale image

Real NVP coupling layer: [Dinh, Sohl-Dickstein, Bengio **1605.08803**]



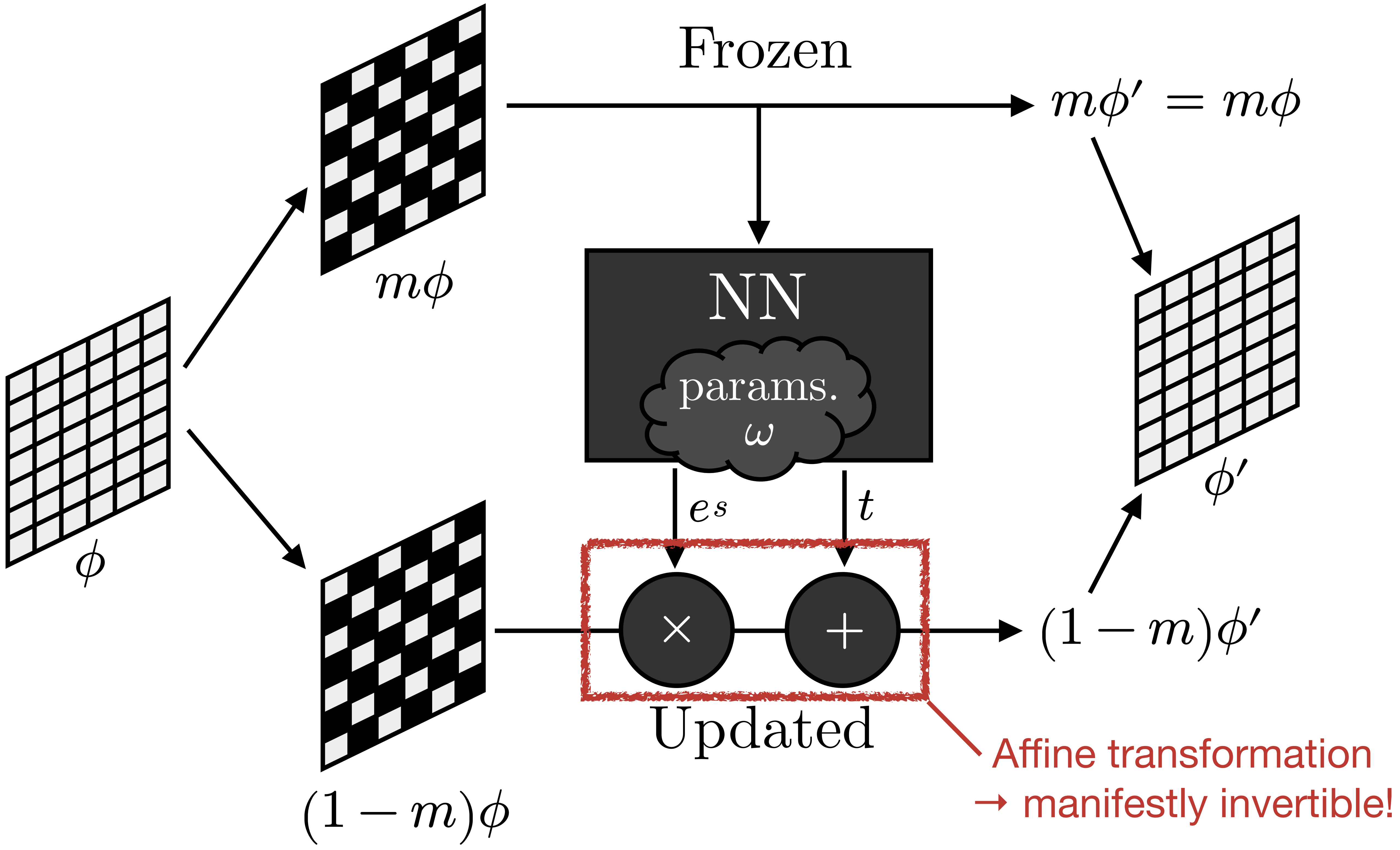
Ex: RNVP for scalar fields



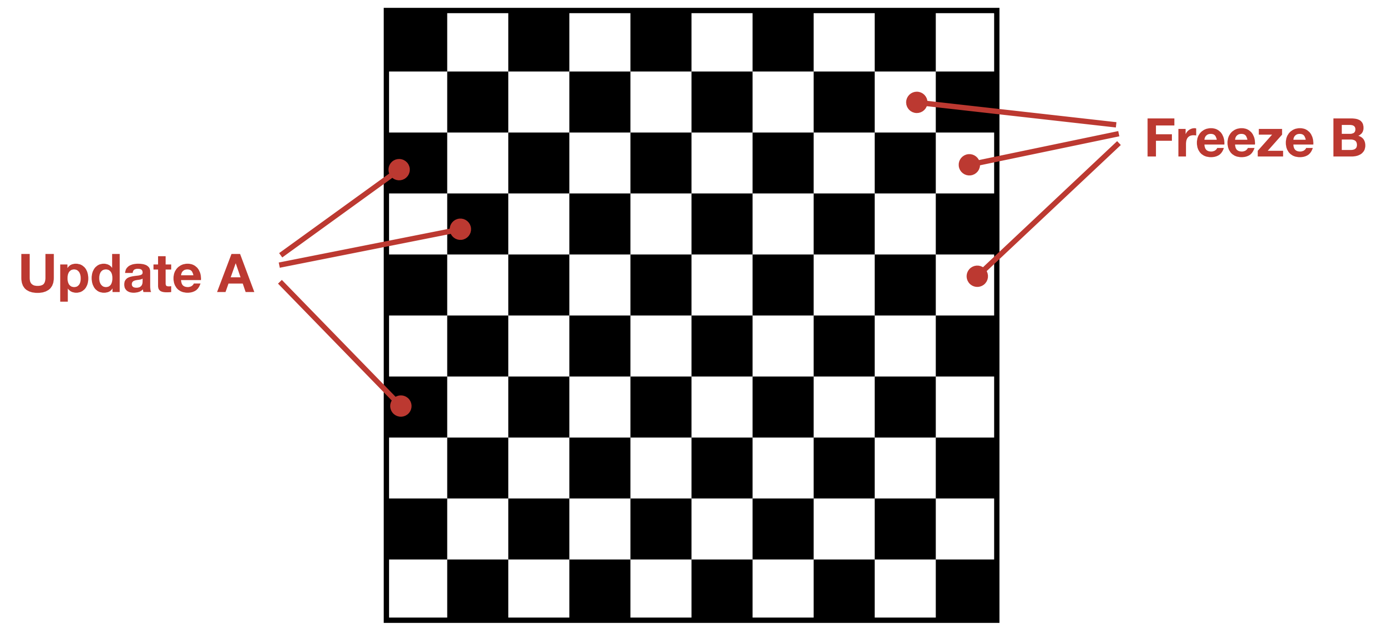
Checkerboard masking pattern m

Real scalar field $\phi(x) \in \mathbb{R} \;\approx\;$ grayscale image

Real NVP coupling layer: [Dinh, Sohl-Dickstein, Bengio **1605.08803**]



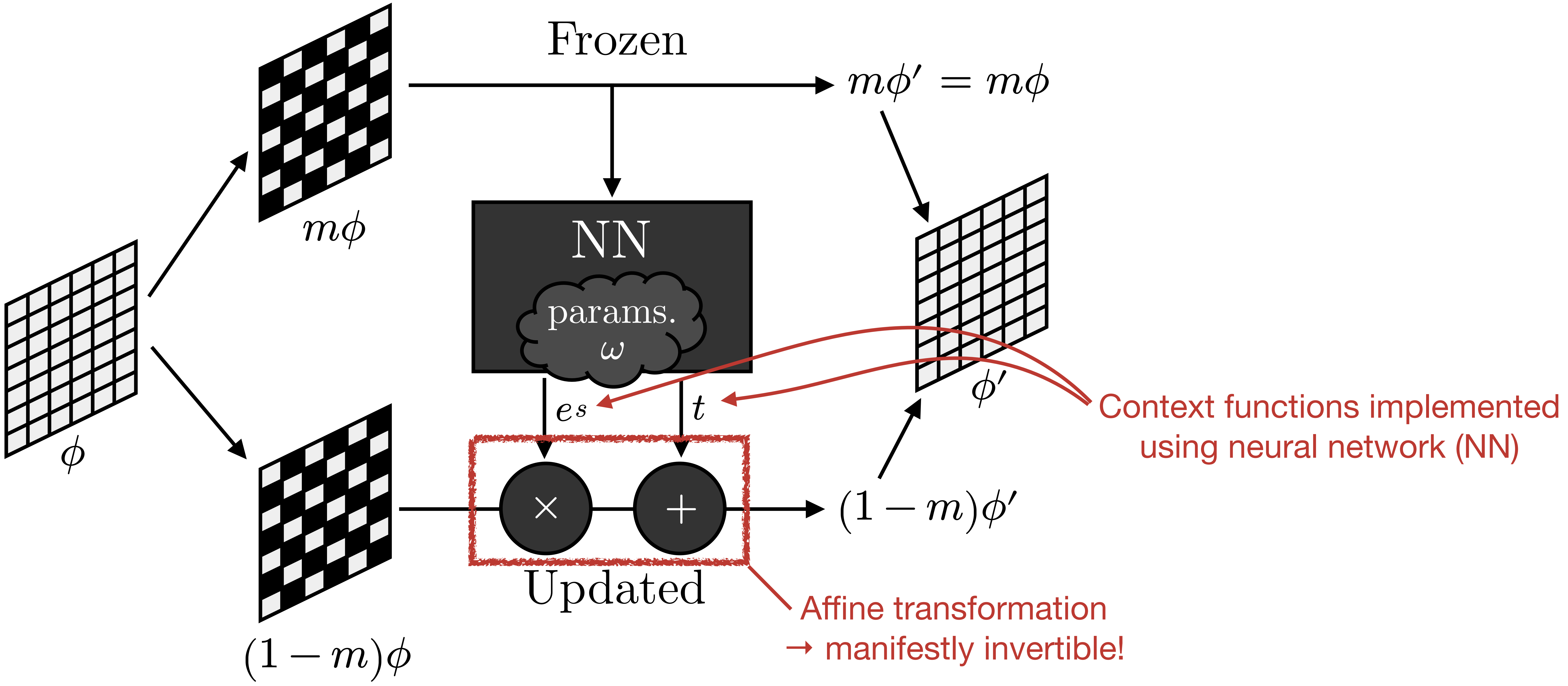
Ex: RNVP for scalar fields



Checkerboard masking pattern m

Real scalar field $\phi(x) \in \mathbb{R} \;\approx\;$ grayscale image

Real NVP coupling layer: [Dinh, Sohl-Dickstein, Bengio **1605.08803**]



Optimizing the model

See also Self-Learning Monte Carlo (SLMC) methods:
[Huang, Wang **PRB95 (2017) 035105**;
Liu, et al. **PRB95 (2017) 041101**;
… and many more …]

Must not require a large number of samples from real distribution to optimize!

Self-training:

- Gradient-based methods applied to loss function to optimize model params ω
  - E.g. Adam optimizer [Kingma, Ba **1412.6980**]
- Loss function = modified **Kullback-Leibler (KL)** divergence

Measures difference between probability distributions

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

Constant shift removes unknown normalization

$$D'_{\mathrm{KL}}(q\,||\,p) := \int \mathscr{D}U\, q(U)\big[\log q(U) + S(U)\big] \geq -\log Z$$

(Using $p(U) = e^{-S(U)}/Z$)

- To estimate loss for grad descent, draw samples **from the model**, measure sample mean of $\big[\log q(U) + S(U)\big]$

Exactness: Flow-based MCMC

Markov chain constructed using Independence Metropolis accept/reject on model proposals.

“Embarrassingly parallel” step!

- **Independent** proposals U' from model distribution q

- **Accept** proposal U', making it next elt of Markov chain, with probability

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

- If **rejected**, duplicate previous elt of Markov chain
  - Only need to compute observables on duplicated elts once!

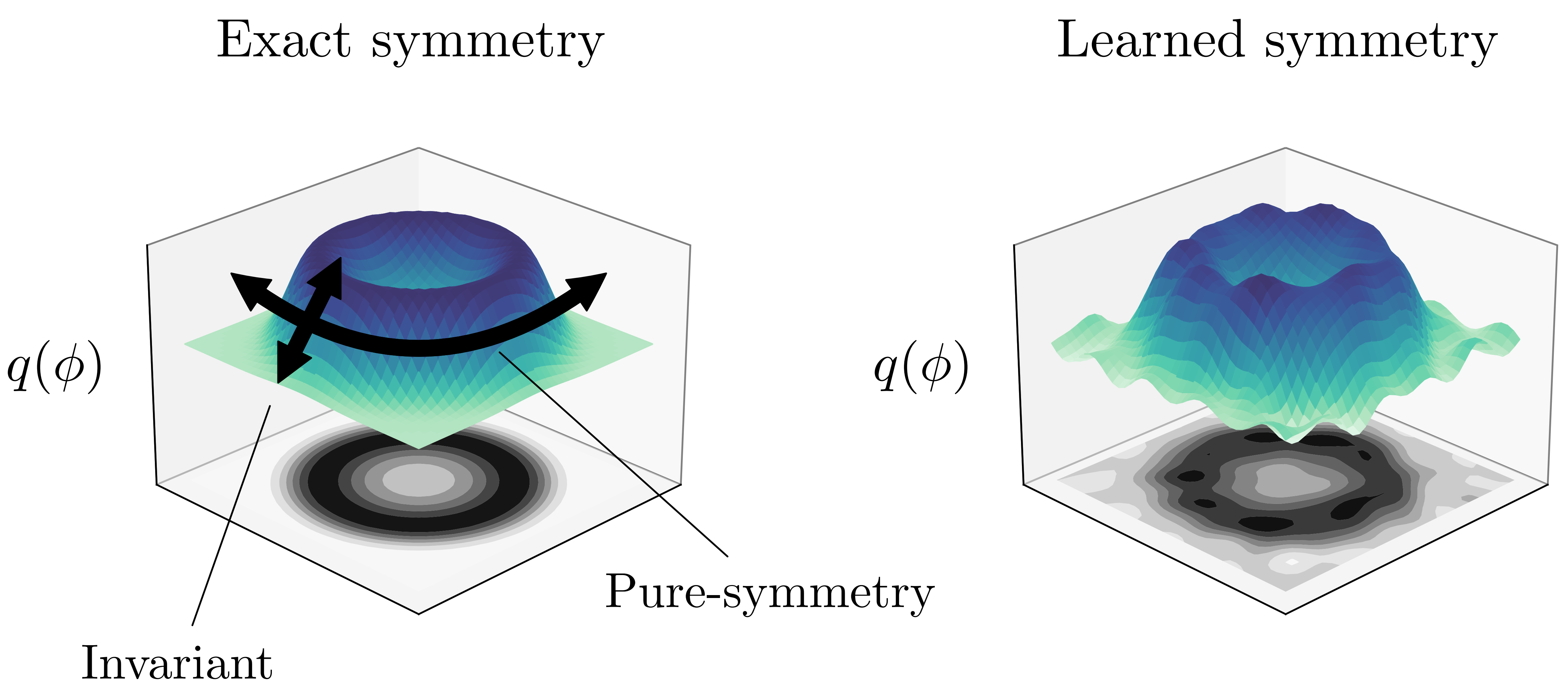
Symmetries in flows

Invariant prior + **equivariant** flow = symmetric model [Cohen, Welling 1602.07576]

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

Symmetries…

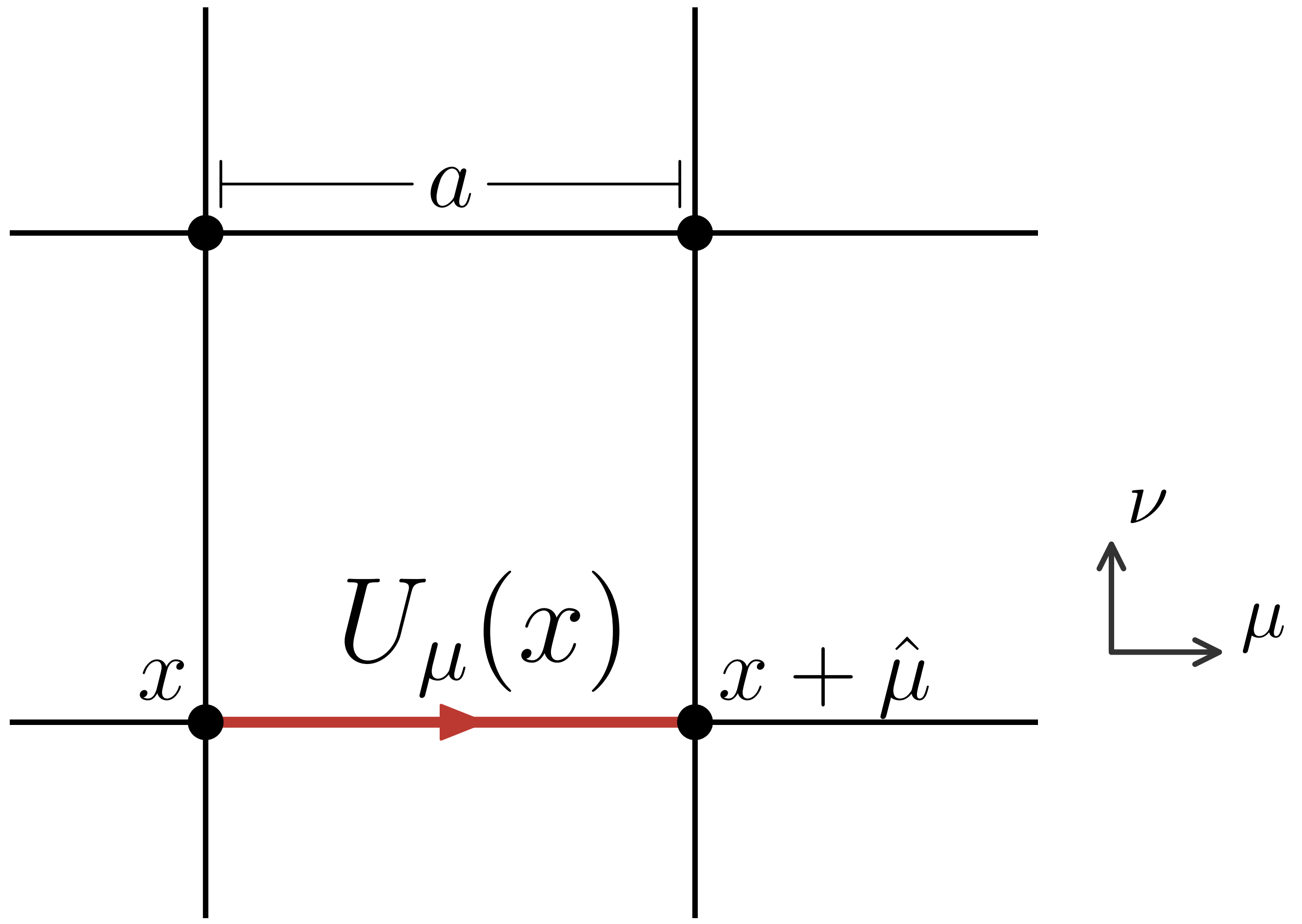
- Reduce data complexity of training
- Reduce model parameter count
- See [D. Müller, Fri] and [M. Favoni, Fri]



Gauge symmetries in flows

Choose to act on the un-fixed link representation $U_\mu(x)$.

Carefully construct architecture to enforce…



Gauge-invariant prior:

Not very difficult!
Uniform distribution works.

With respect to Haar measure

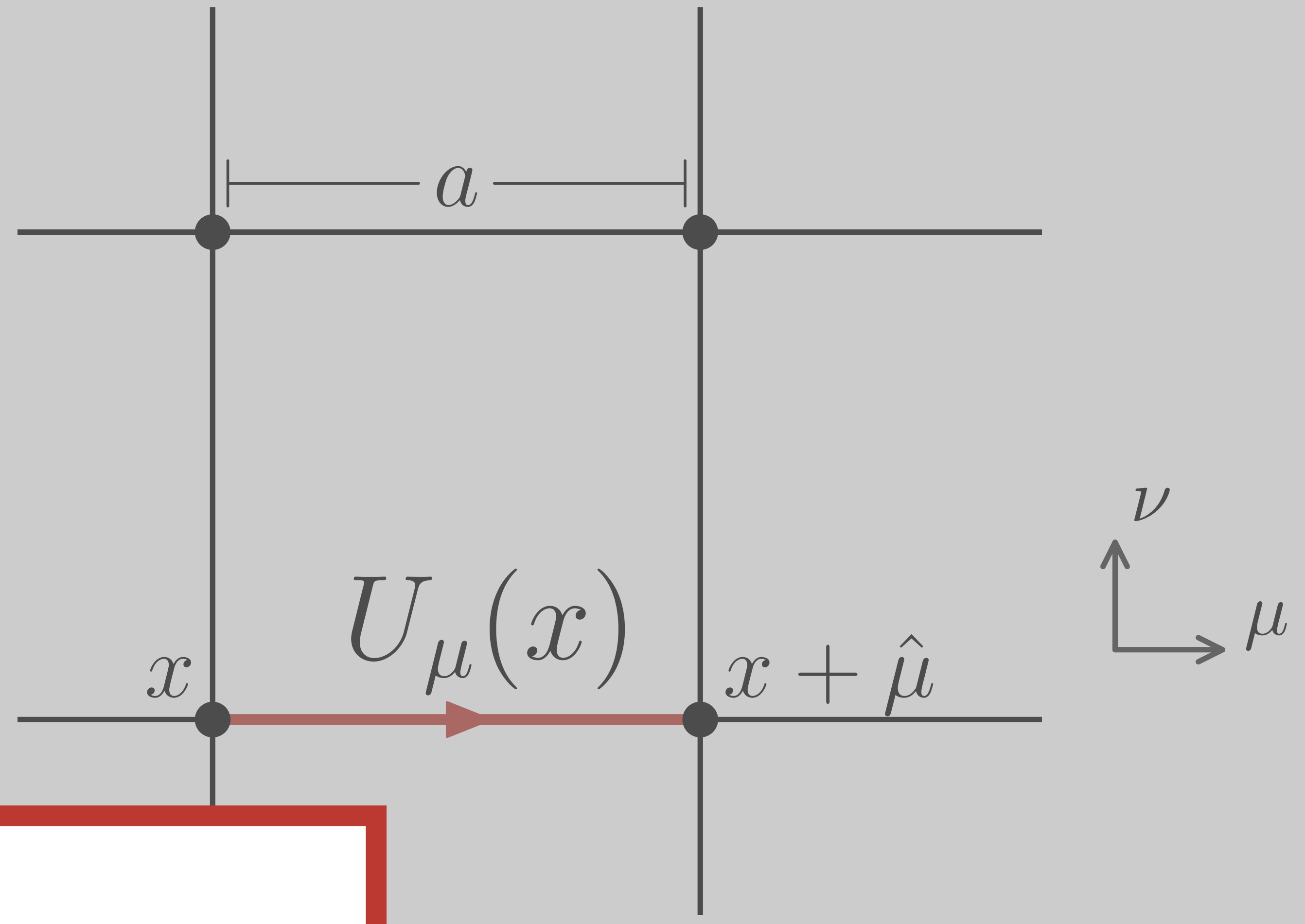
$$r(U) = 1$$

Gauge-equivariant flow:

Coupling layers acting on (untraced) Wilson loops.

Loop transformation easier to satisfy.

Gauge symmetries in flows

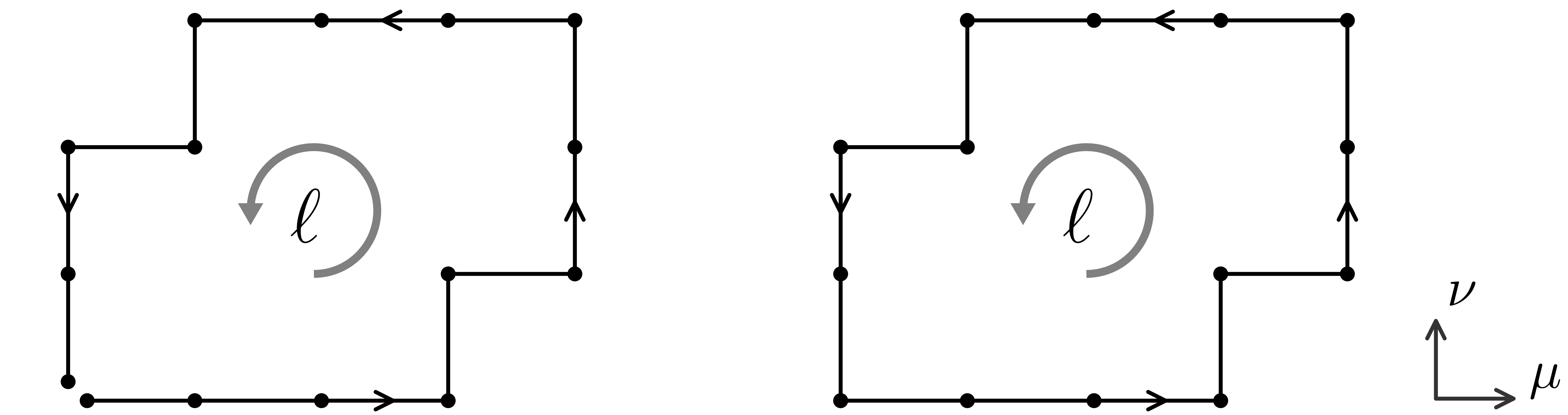


Choose to act on the un-fixed link representation $U_\mu(x)$.

Carefully cons

Open loop

Closed loop



$W_\ell(x) \to \Omega(x) W_\ell(x) \Omega^\dagger(x)$

$\mathrm{tr}\, W_\ell(x) \to \mathrm{tr}\, W_\ell(x)$

Gauge

Not

Uniform

With respect to Haar measure

iant flow:

acting on

on loops.

tion easier

to satisfy.

Gauge-equivariant coupling layer

Compute a **field of Wilson loops** $W_\ell(x)$.

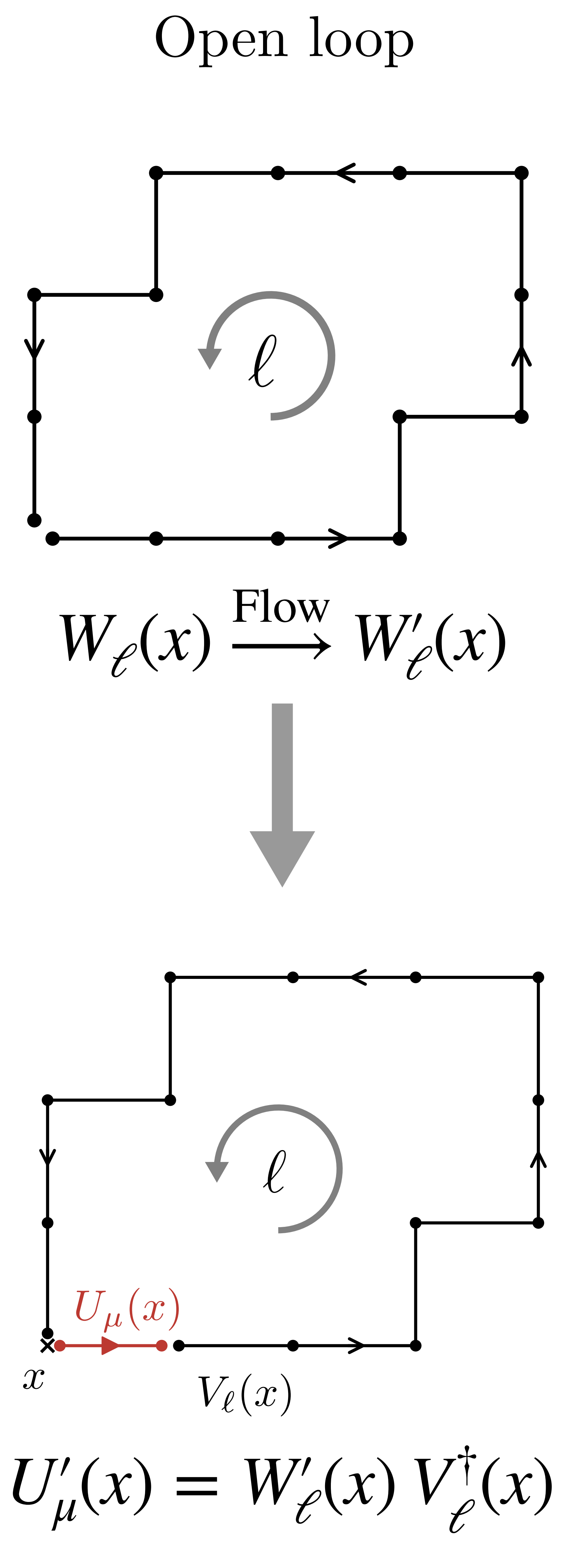
Inner coupling layer [function of $W_\ell(x)$]

- "Actively" update a subset of loops.*
- Condition on "frozen" closed loops.

Gauge invariant!

Outer coupling layer [function of $U_\mu(x)$]

- Solve for link update to satisfy actively updated loops.
- Other loops in $W_\ell(x)$ may "passively" update.



[GK, Albergo, Boyda, Cranmer, Hackett, Racanière, Rezende, Shanahan **PRL125 (2020) 121601**]
[Boyda, GK, Racanière, Rezende, Albergo, Cranmer, Hackett, Shanahan **PRD103 (2021) 074504**]

Gauge-equivariant coupling layer

Compute a **field of Wilson loops** $W_\ell(x)$.

Inner coupling layer [function of $W_\ell(x)$]

- "Actively" update a subset of loops.*
- Condition on "frozen" closed loops.

Gauge invariant!

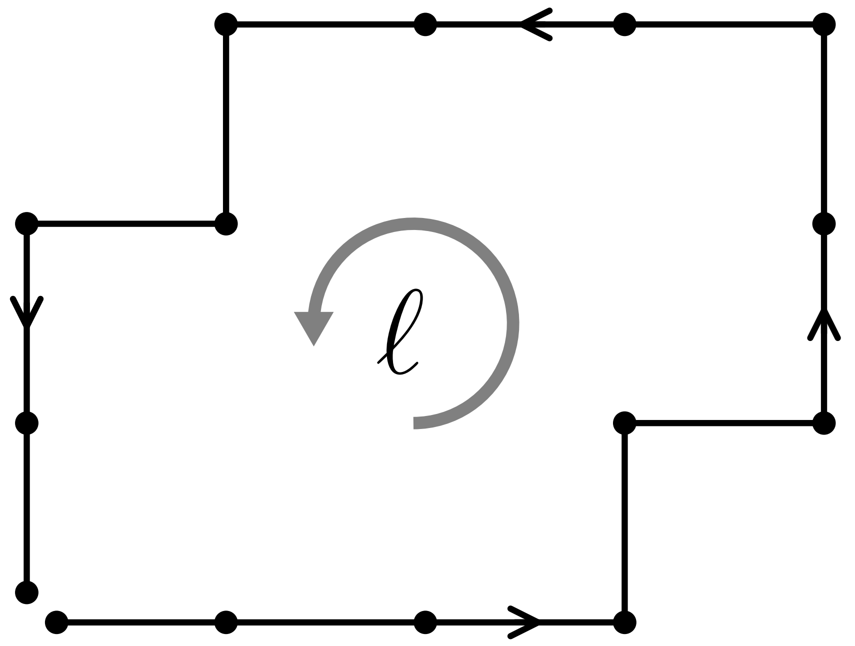
* This **"kernel"** must satisfy:

$$h(W_\ell^\Omega(x)) = h^\Omega(W_\ell(x))$$

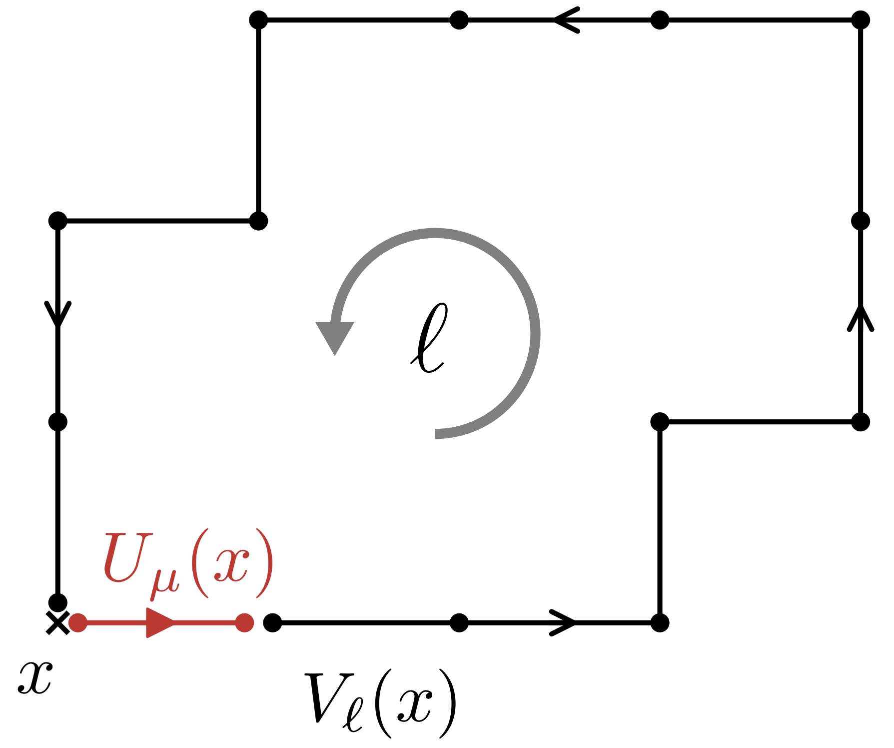
Outer coupling layer [function of $U_\mu(x)$]

- Solve for link update to satisfy actively updated loops.
- Other loops in $W_\ell(x)$ may "passively" update.

Open loop



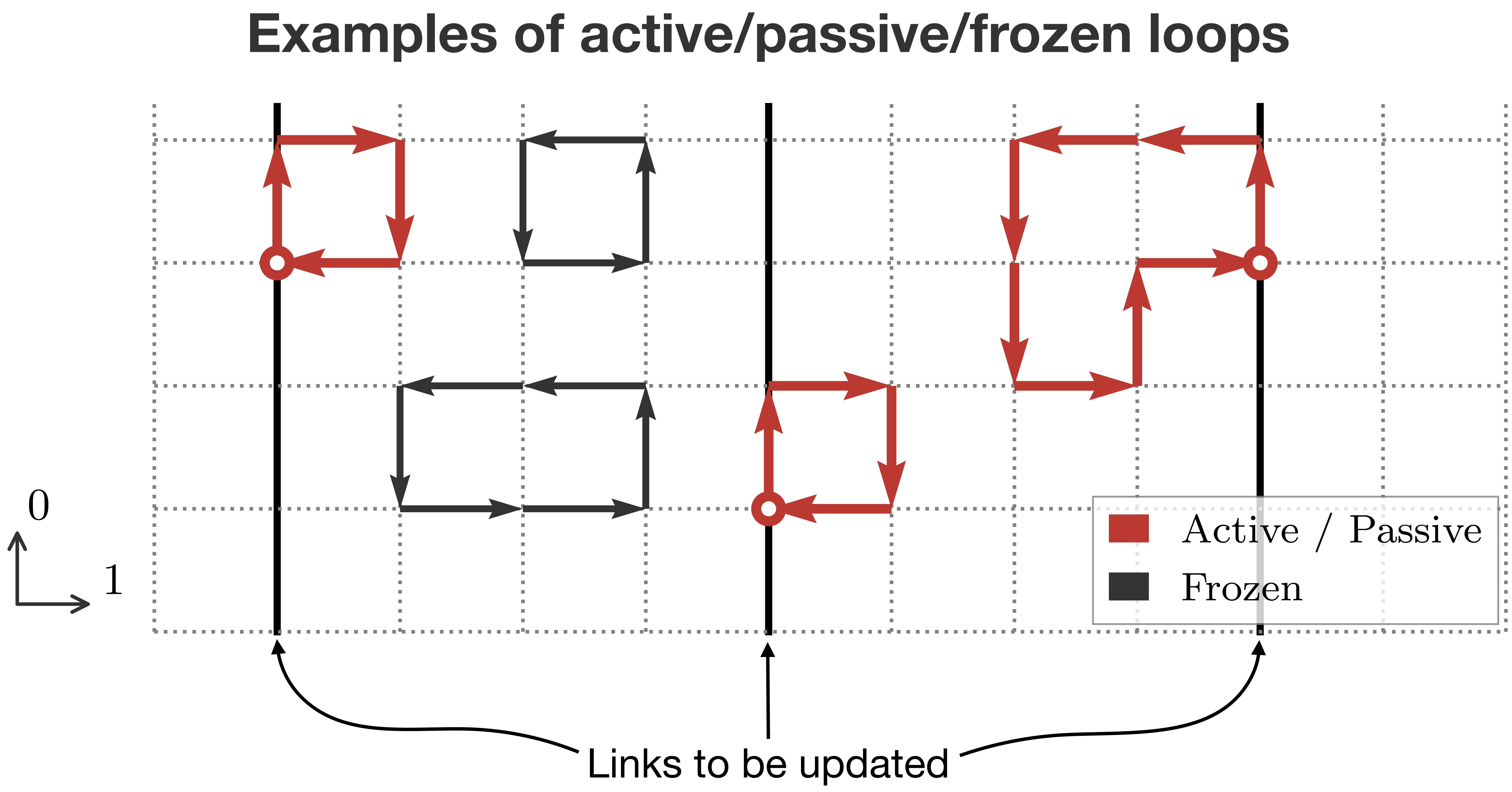
$$W_\ell(x) \xrightarrow{\text{Flow}} W'_\ell(x)$$



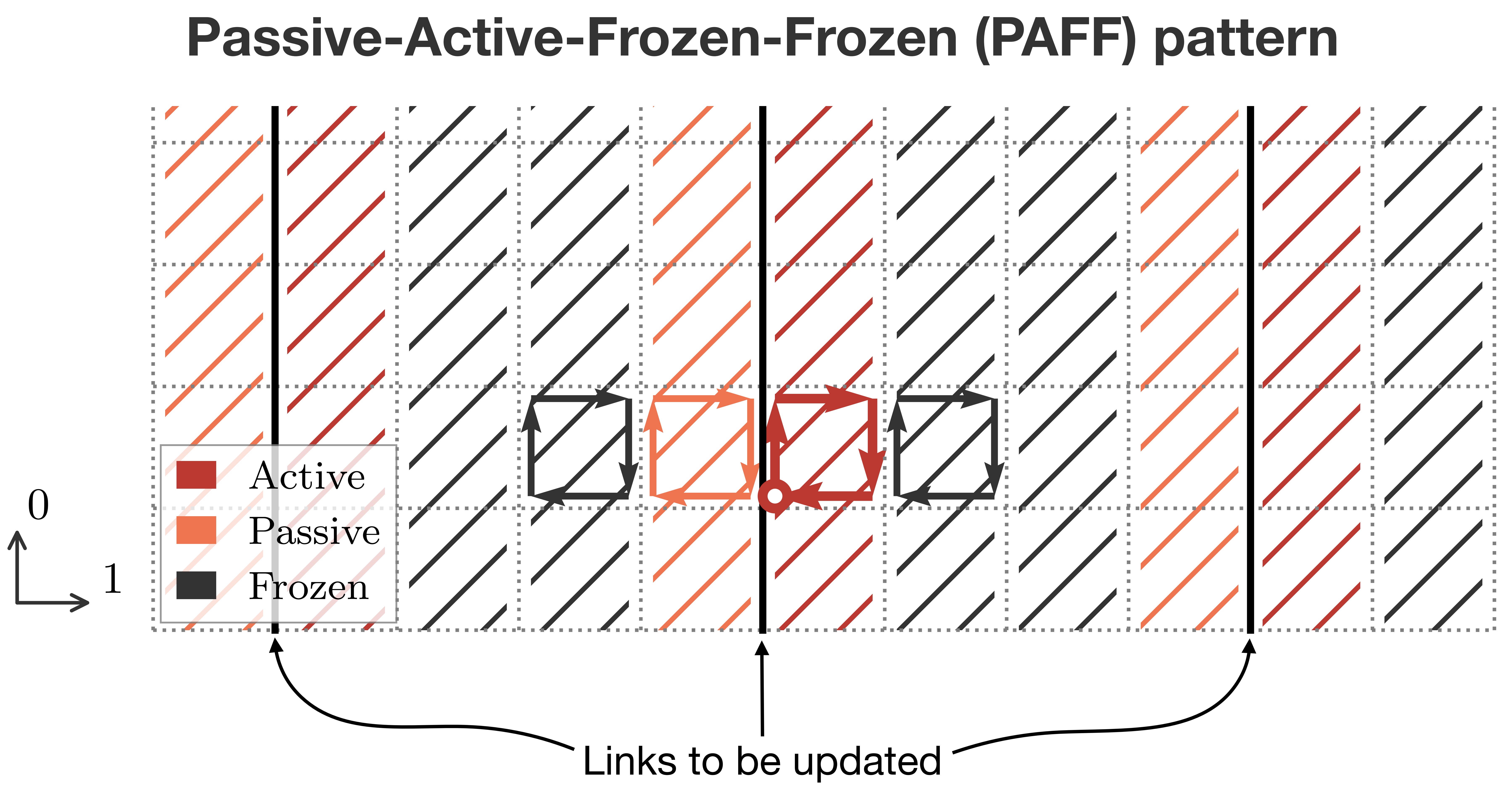
$$U'_\mu(x) = W'_\ell(x)\, V_\ell^\dagger(x)$$

[GK, Albergo, Boyda, Cranmer, Hackett, Racanière, Rezende, Shanahan **PRL125 (2020) 121601**]
[Boyda, GK, Racanière, Rezende, Albergo, Cranmer, Hackett, Shanahan **PRD103 (2021) 074504**]

Active, passive, and frozen loops



Active, passive, and frozen loops

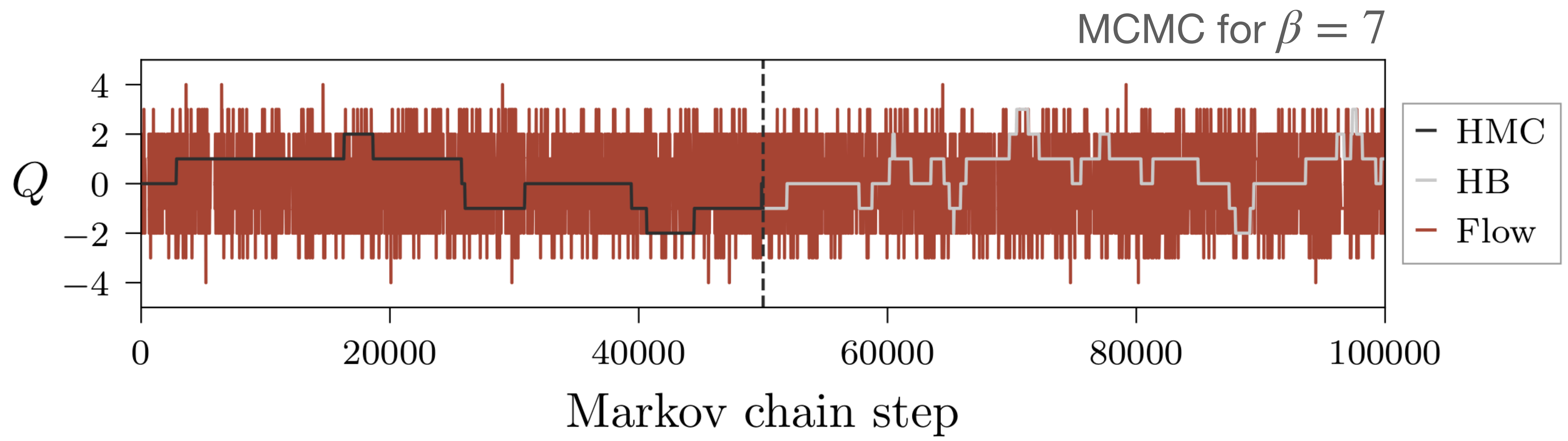


Results for U(1) gauge theory

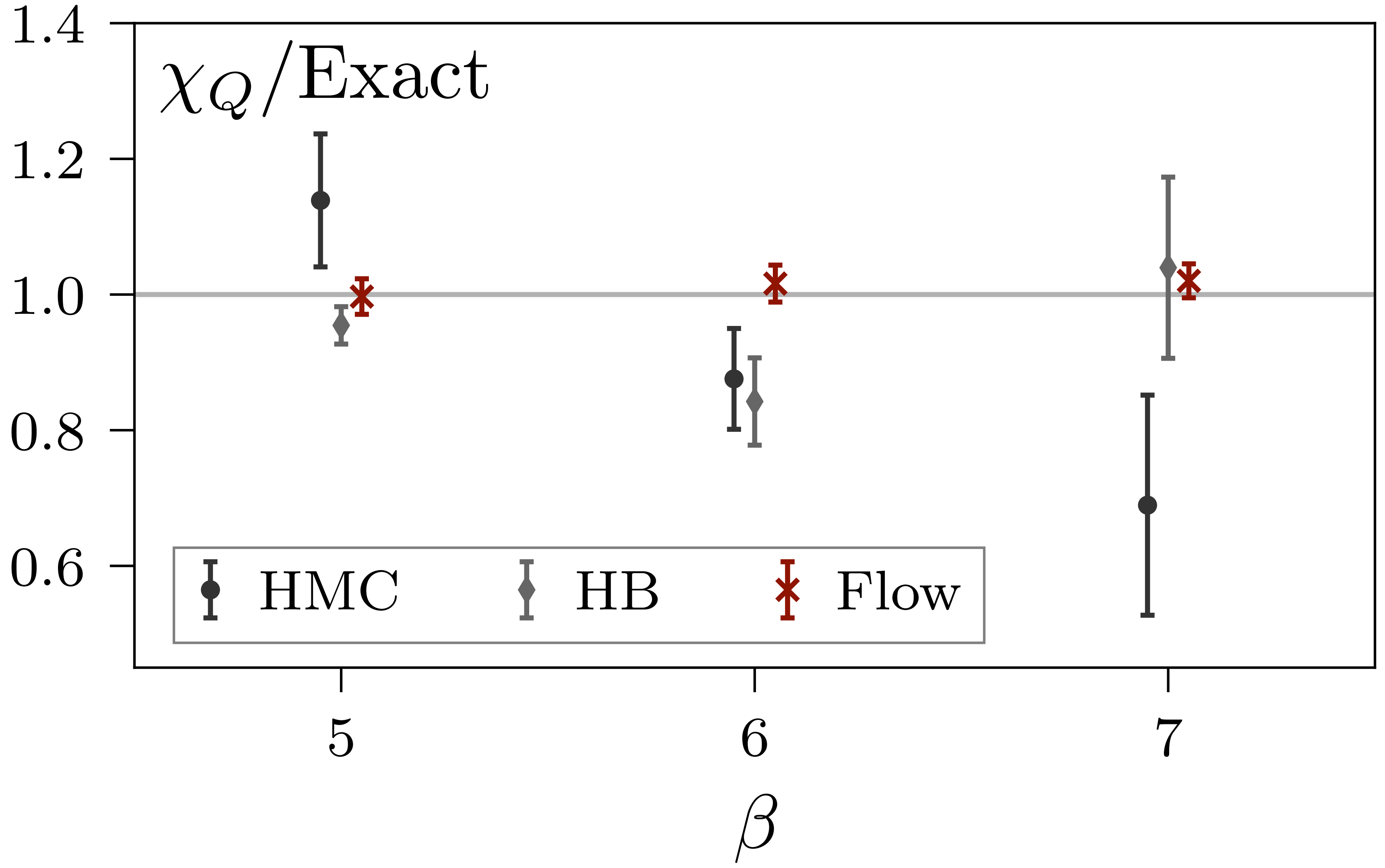
$$S(U) = -\beta \sum_x \sum_{\mu<\nu} \mathrm{Re}\, P_{\mu\nu}(x)$$
$$P_{\mu\nu}(x) = U_\mu(x) U_\nu(x+\hat{\mu}) U_\mu^\dagger(x+\hat{\nu}) U_\nu^\dagger(x)$$

There is exact lattice topology in 2D.

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



- Compared **flow**, **analytical**, **HMC**, and **heat bath** on 16×16 lattices for $\beta = \{1,\ldots,7\}$
- Topo freezing in HMC and heat bath
- Gauge-equiv flow-based model at each β
- Flow-based MCMC observables agree



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

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

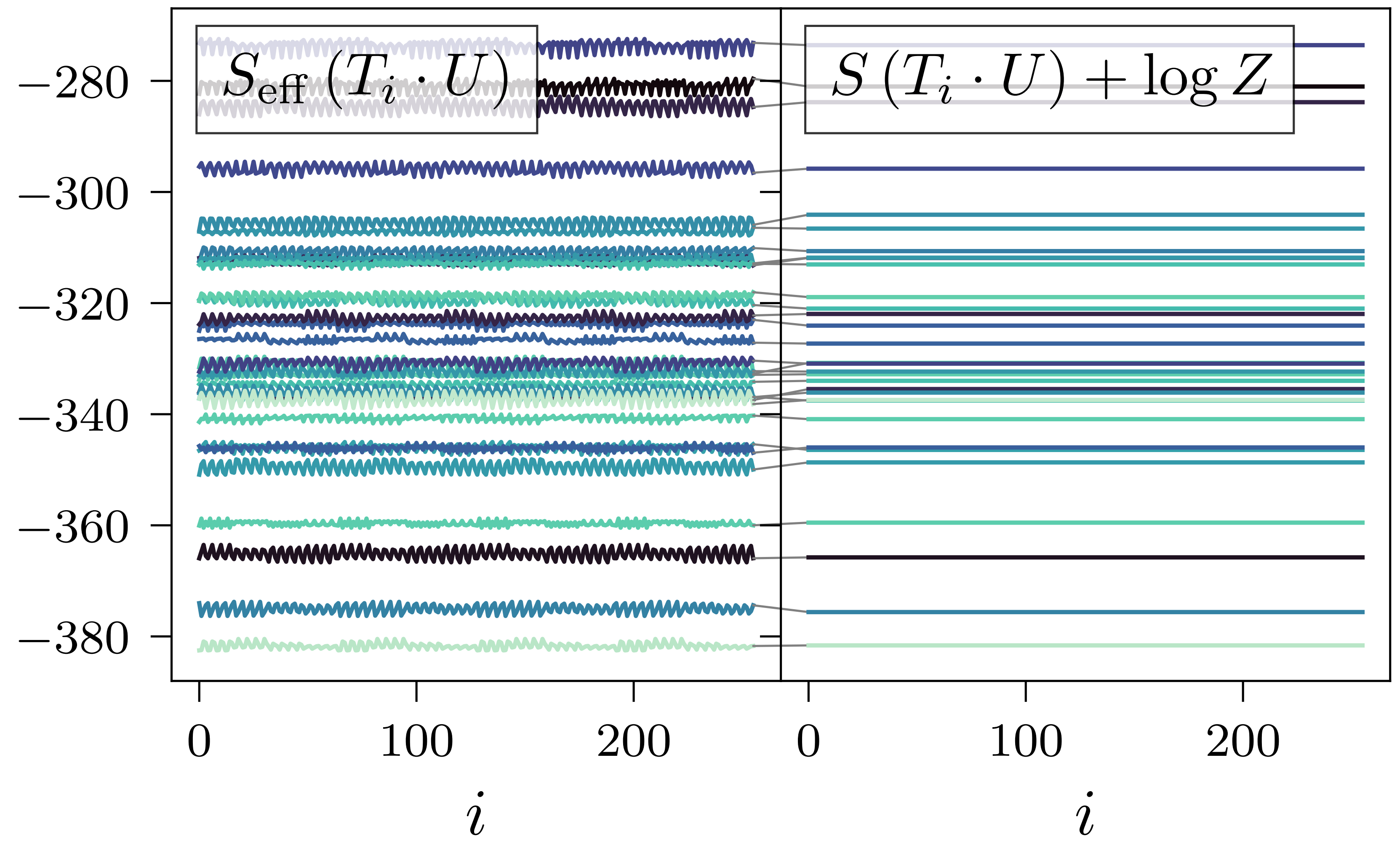
Results for SU(2) and SU(3) gauge theory

- Similar study over 2D 16×16 lattices
- Flow-based MCMC observables agree with analytical
- **High-quality models:** autocorrelation time in flow-based Markov chain τ_{int} = 1—4

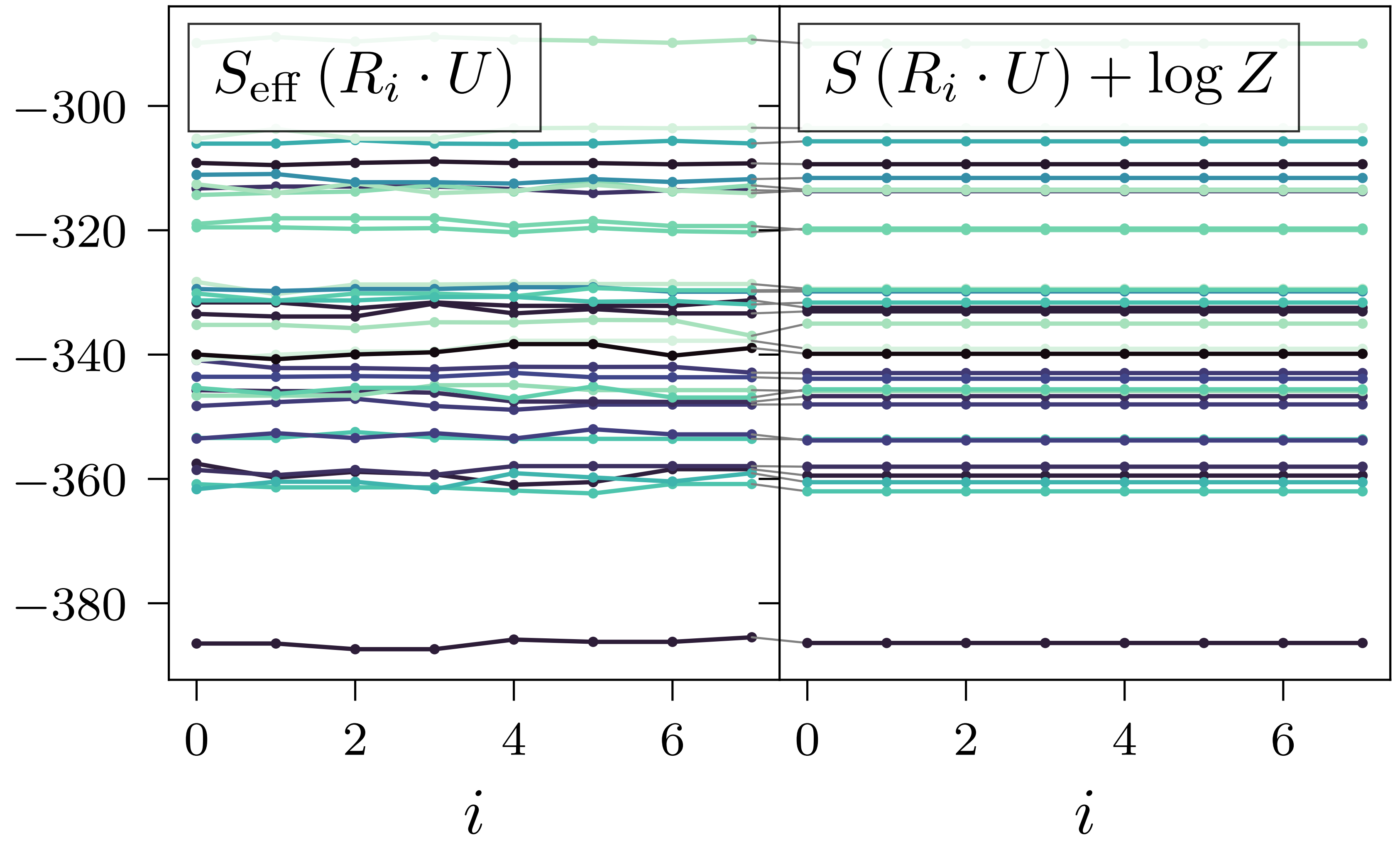
	SU(2)			SU(3)		
β	1.8	2.2	2.7	4.0	5.0	6.0
ESS(%)	91	80	56	88	75	48

Measure of "effective" # samples from target dist for each sample drawn from model (100% = perfect model)

Exact translational subgroup; residual learned



Rotation and reflection symmetry learned



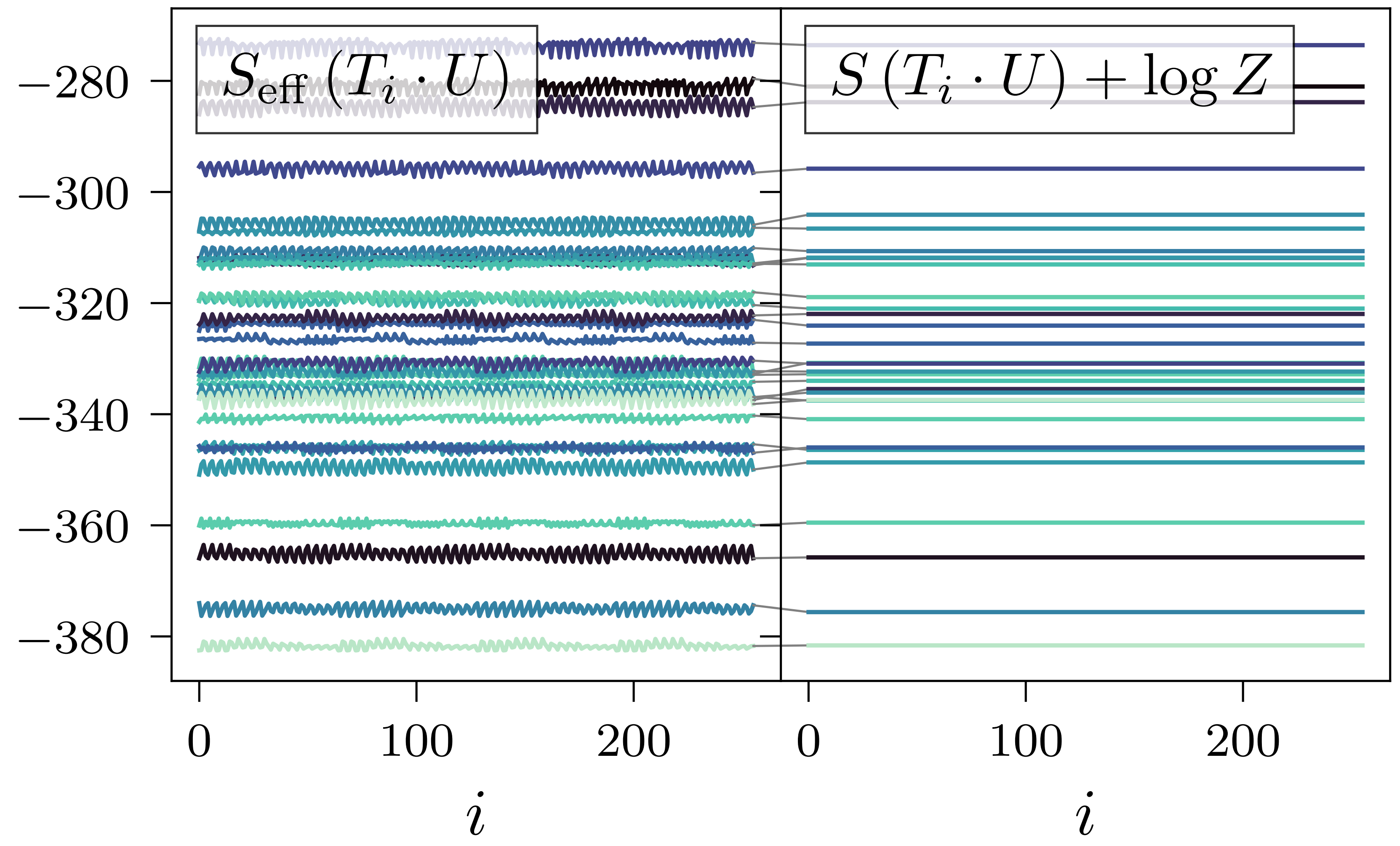
Results for SU(2) and SU(3) gauge theory

- Similar study over 2D 16×16 lattices
- Flow-based MCMC observables agree with analytical
- **High-quality models:** autocorrelation time in flow-based Markov chain τ_{int} = 1—4

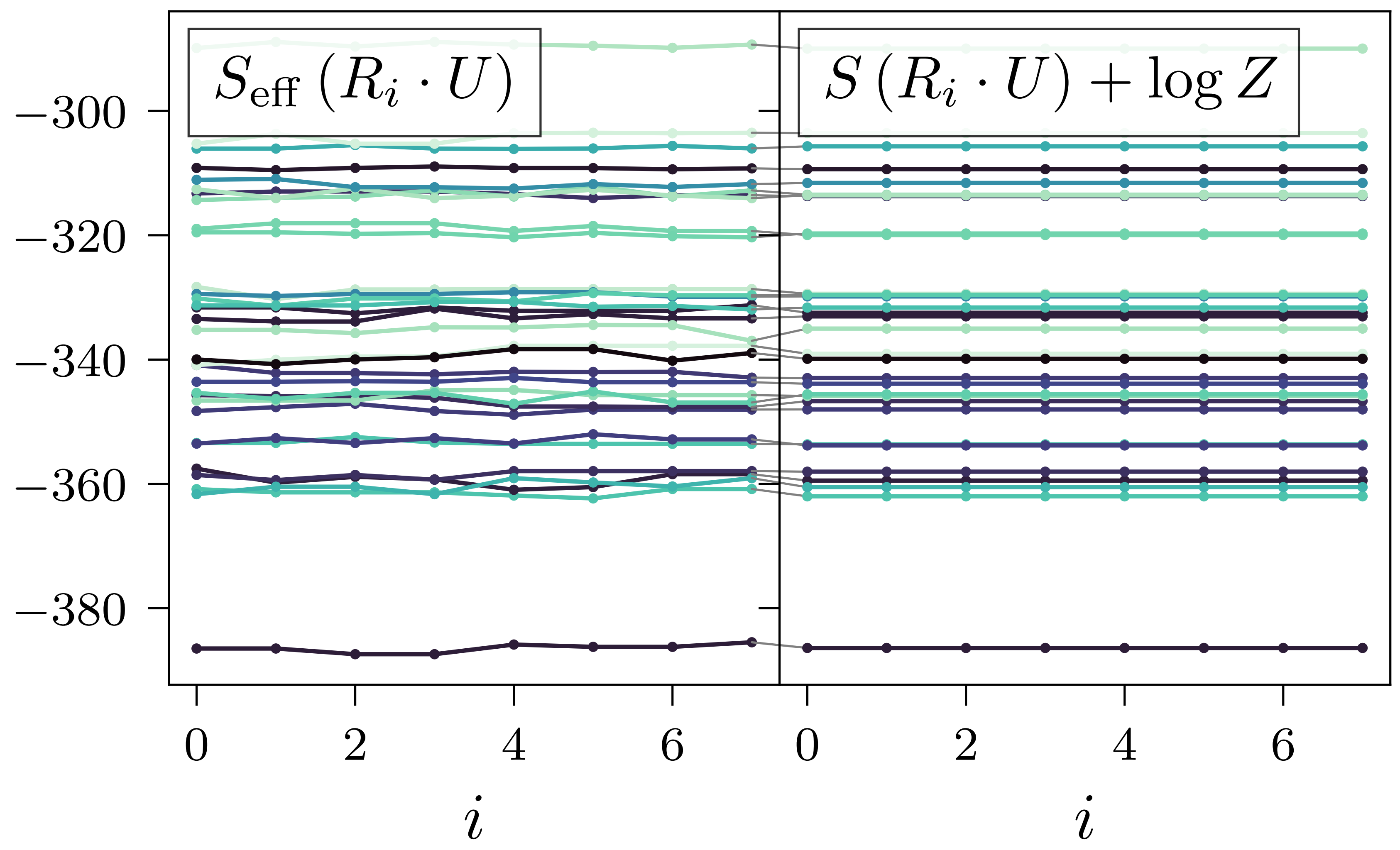
	SU(2)			SU(3)		
β	1.8	2.2	2.7	4.0	5.0	6.0
ESS(%)	91	80	56	88	75	48

Measure of "effective" # samples from target dist for each sample drawn from model (100% = perfect model)

Exact translational subgroup; residual learned



Rotation and reflection symmetry learned



Promising early results. No theoretical obstacle to scaling to 4D $SU(N)$ lattice gauge theory.

Fermions in field theory

Grassmann representation in path integral means…

- … we cannot sample fermion fields
- … integrating out fermions results in costly fermion determinants

$$\int \mathcal{D}\psi \mathcal{D}\bar{\psi} \prod_f e^{-\bar{\psi}_f D_f \psi_f} = \prod_f \det D_f$$

Pseudofermions used in standard MCMC for theories with dynamical fermions.

$$\int \mathcal{D}\psi \mathcal{D}\bar{\psi} \prod_f e^{-\bar{\psi}_f D_f \psi_f} \propto \int \mathcal{D}\varphi \mathcal{D}\varphi^\dagger \prod_k e^{-\varphi_k^\dagger \mathcal{M}_k^{-1} \varphi_k}$$

Starting point for flow-based sampling

5 ways to marginalize

Any could in principle be learned by flow-based models.

Below: Bosonic part of action written generically as $S_B(\phi)$

Name	Probability density
Joint[A]	$p(\phi, \varphi) = \frac{1}{Z} \exp(-S_B(\phi) - \varphi^\dagger [\mathcal{M}(\phi)]^{-1} \varphi)$
ϕ-marginal	$p(\phi) = \frac{Z_\mathcal{N}}{Z} \exp(-S_B(\phi)) \boxed{\det \mathcal{M}(\phi)}$
φ-conditional[A,B]	$p(\varphi\|\phi) = \frac{1}{Z_\mathcal{N} \det \mathcal{M}(\phi)} \exp(-\varphi^\dagger [\mathcal{M}(\phi)]^{-1} \varphi)$
φ-marginal[C]	$p(\varphi) = \frac{1}{Z} \boxed{\int d\phi \, \exp(-S_B(\phi) - \varphi^\dagger [\mathcal{M}(\phi)]^{-1} \varphi)}$
ϕ-conditional[A]	$p(\phi\|\varphi) = \frac{\exp(-S_B(\phi) - \varphi^\dagger [\mathcal{M}(\phi)]^{-1} \varphi)}{\int d\phi \, \exp(-S_B(\phi) - \varphi^\dagger [\mathcal{M}(\phi)]^{-1} \varphi)}$

Expensive to evaluate det exactly

Can actually be sampled directly (e.g. pseudofermion refresh in HMC)

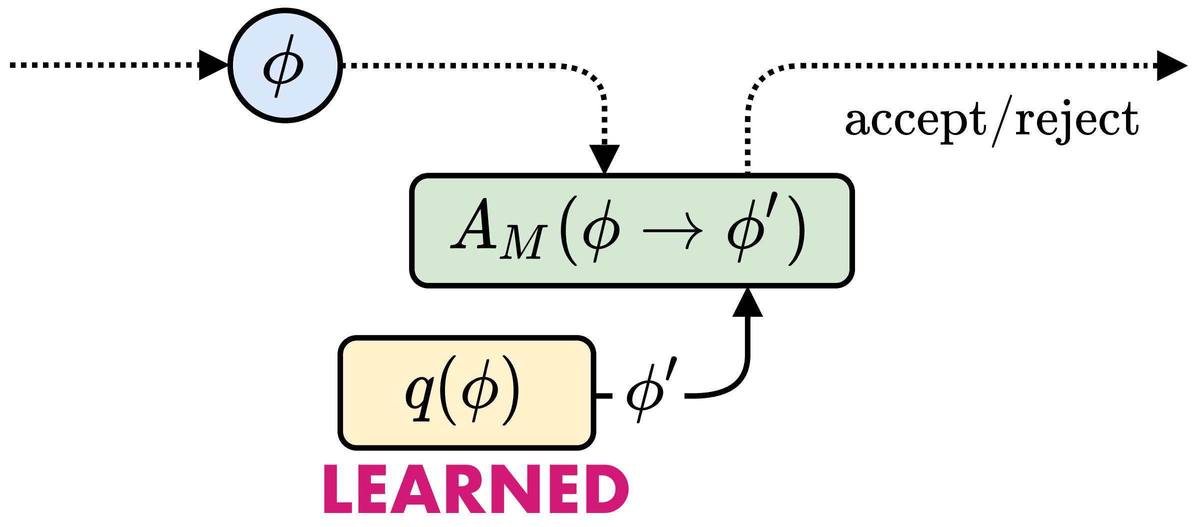
Intractable density (even unnormalized)

[Albergo, GK, Racanière, Rezende, Urban, Boyda, Cranmer, Hackett, Shanahan **2106.05934**]

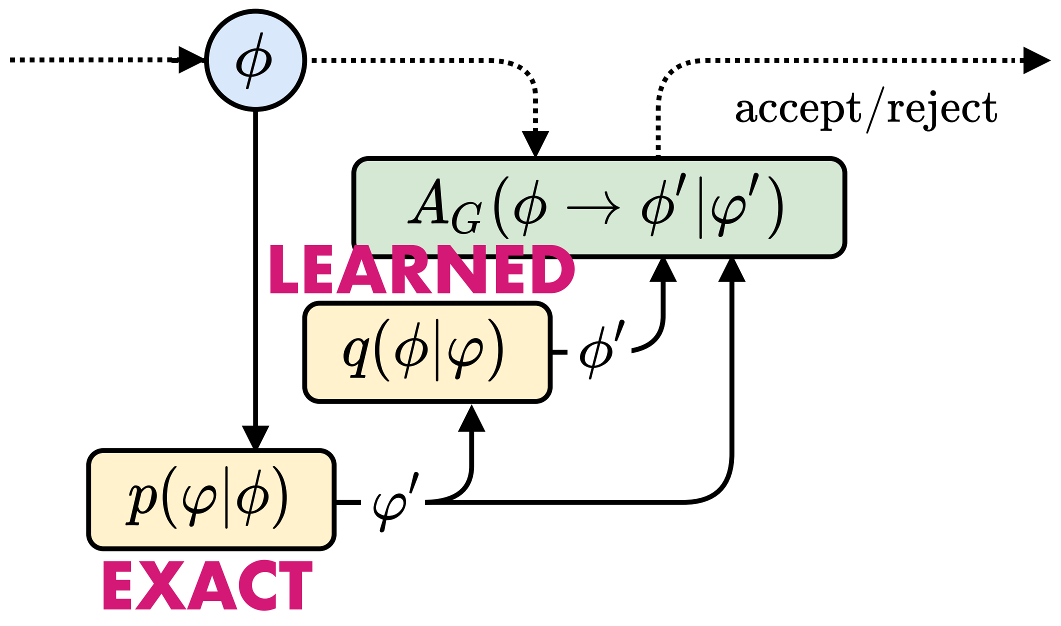
Proposed exact sampling schemes

Using a variety of learned densities $q(\ldots)$ — Best choice not yet clear!

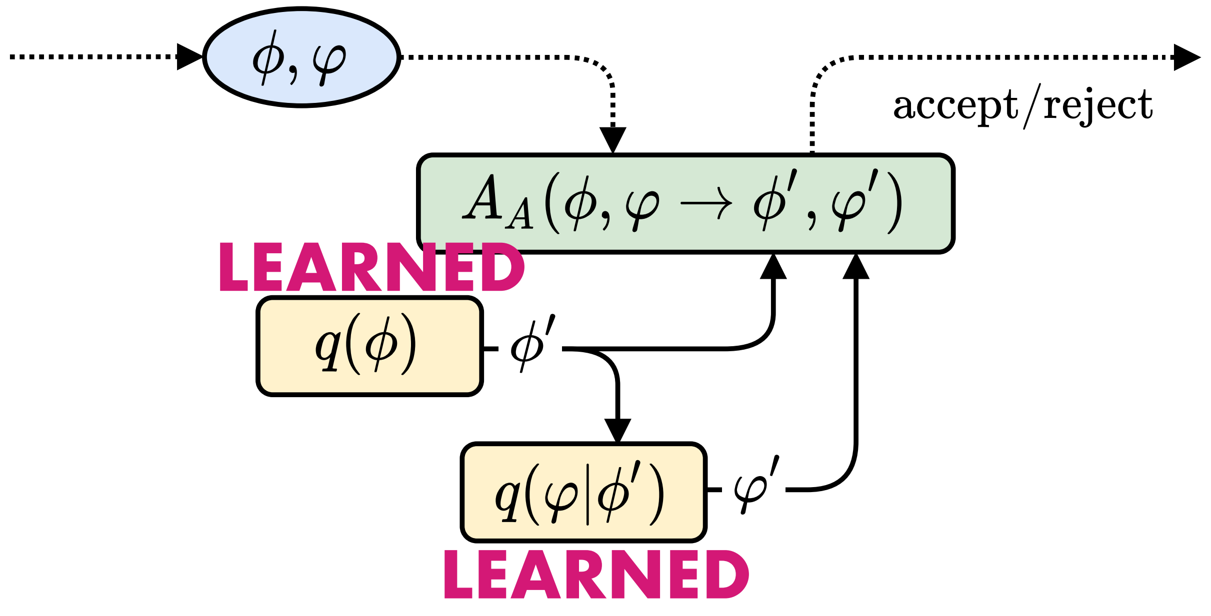
(1) ϕ-marginal



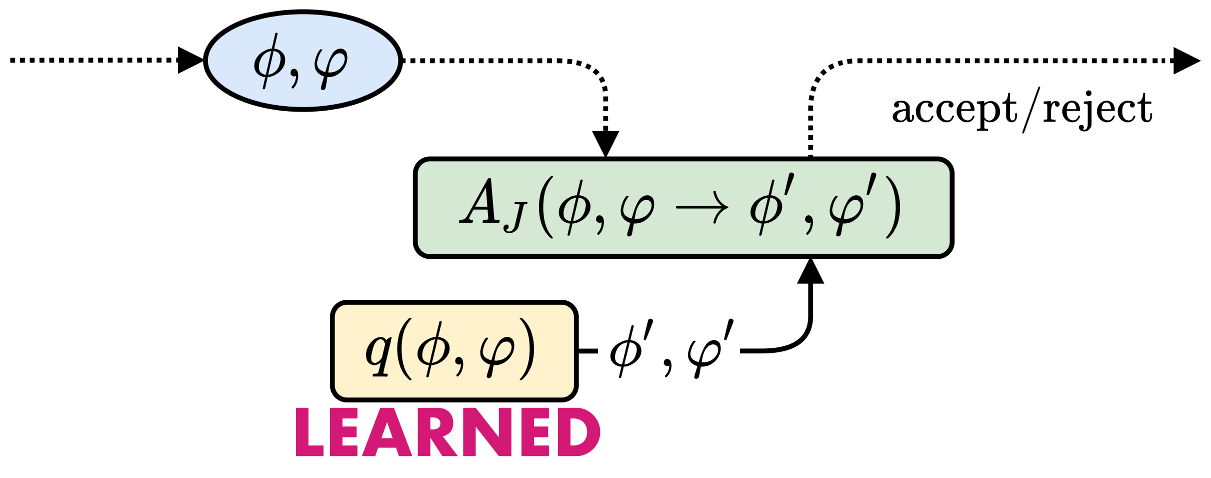
(2) Gibbs



(3) Autoregressive



(4) Joint



Key takeaways:

- **Exact** regardless of quality of modeled densities $q(\ldots)$
- Can define sampler over
  … bosonic fields alone (ϕ) or
  … bosonic + PF fields (ϕ, φ)
- For **Gibbs**, even a perfect model may have residual autocorrelations

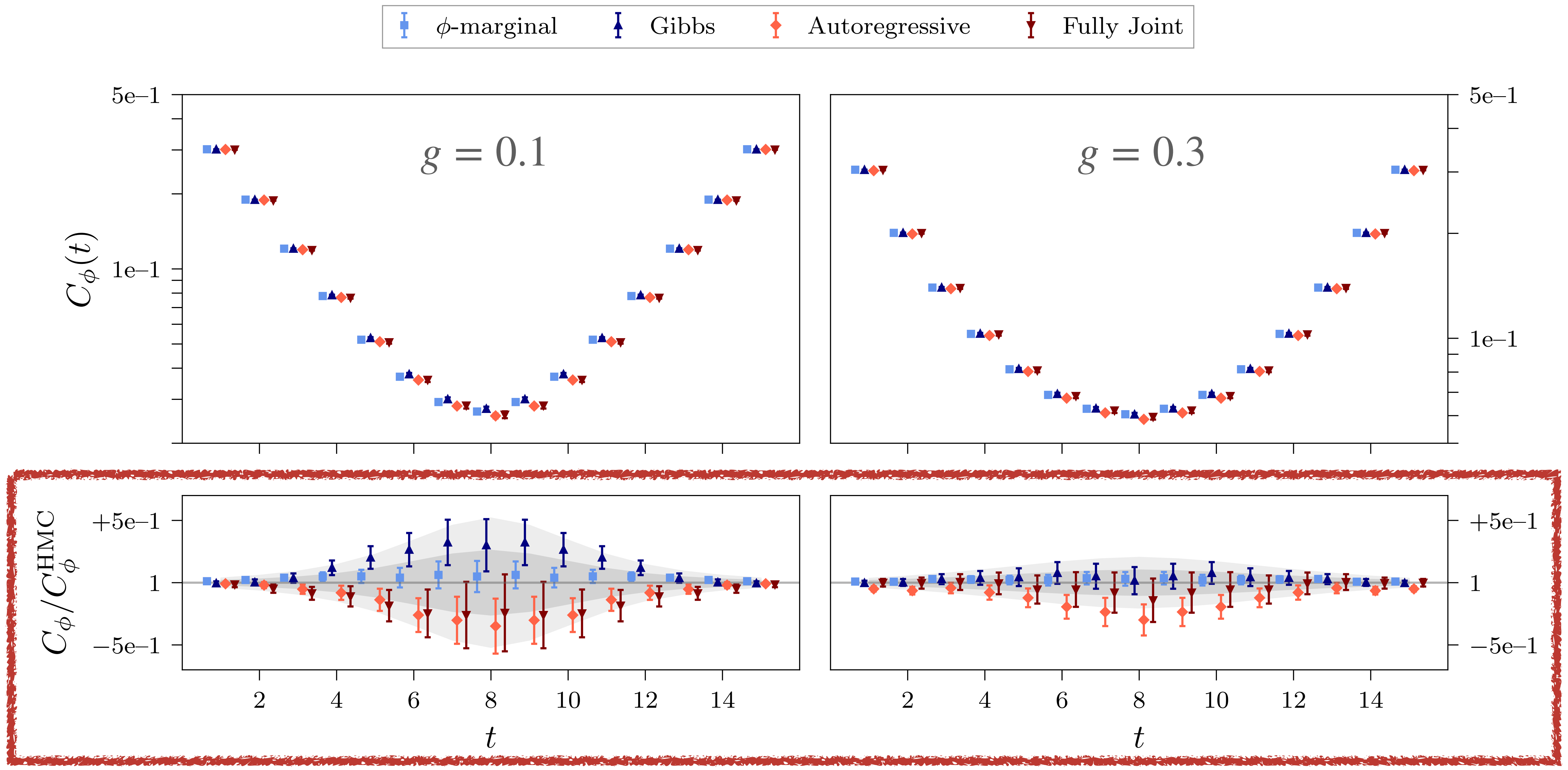
Results for Yukawa model

Staggered Dirac op with Yukawa coupling $g\phi\bar{\psi}\psi$ and mass term $M\bar{\psi}\psi$

Studied 2D ϕ^4 model coupled via Yukawa interaction to staggered ψ

$$S(\phi,\psi) = \sum_{x\in\Lambda} [-2\sum_{\mu=1}^{d} \phi(x)\phi(x+\hat{\mu}) + (m^2+2d)\phi(x)^2 + \lambda\phi(x)^4] + \sum_{f=1}^{N_f} \bar{\psi}_f D_f[\phi]\psi_f$$

- 16×16 lattices
- Two degenerate fermions ($N_f = 2$)
- Massless ($M = 0$)
- Variety of models, all 4 sampling schemes



Correlation functions effectively reproduced

Summary and Outlook

Gauge symmetry encoded in flow models using:

- Gauge equivariant coupling layers
- Kernels for $U(1)$ and $SU(N)$

Several building blocks for models targeting theories with **dynamical fermions**.

Effective models produced for $U(1)$, $SU(2)$, $SU(3)$ lattice gauge theory and a ϕ^4 Yukawa model in 1+1D.

Future directions:

1. Higher spacetime dims
2. Tuning of training hyperparameters
3. Efficient model architectures at scale?

Summary and Outlook

Gauge symmetry encoded in flow models using:

- Gauge equivariant coupling layers
- Kernels for $U(1)$ and $SU(N)$

Several building blocks for models targeting theories with **dynamical fermions**.

Effective models produced for $U(1)$, $SU(2)$, $SU(3)$ lattice gauge theory and a ϕ^4 Yukawa model in 1+1D.

Future directions:

1. Higher spacetime dims
2. Tuning of training hyperparameters
3. Efficient model architectures at scale?

See also:

Approaches to multimodal sampling and mixed HMC + flow-based sampling:

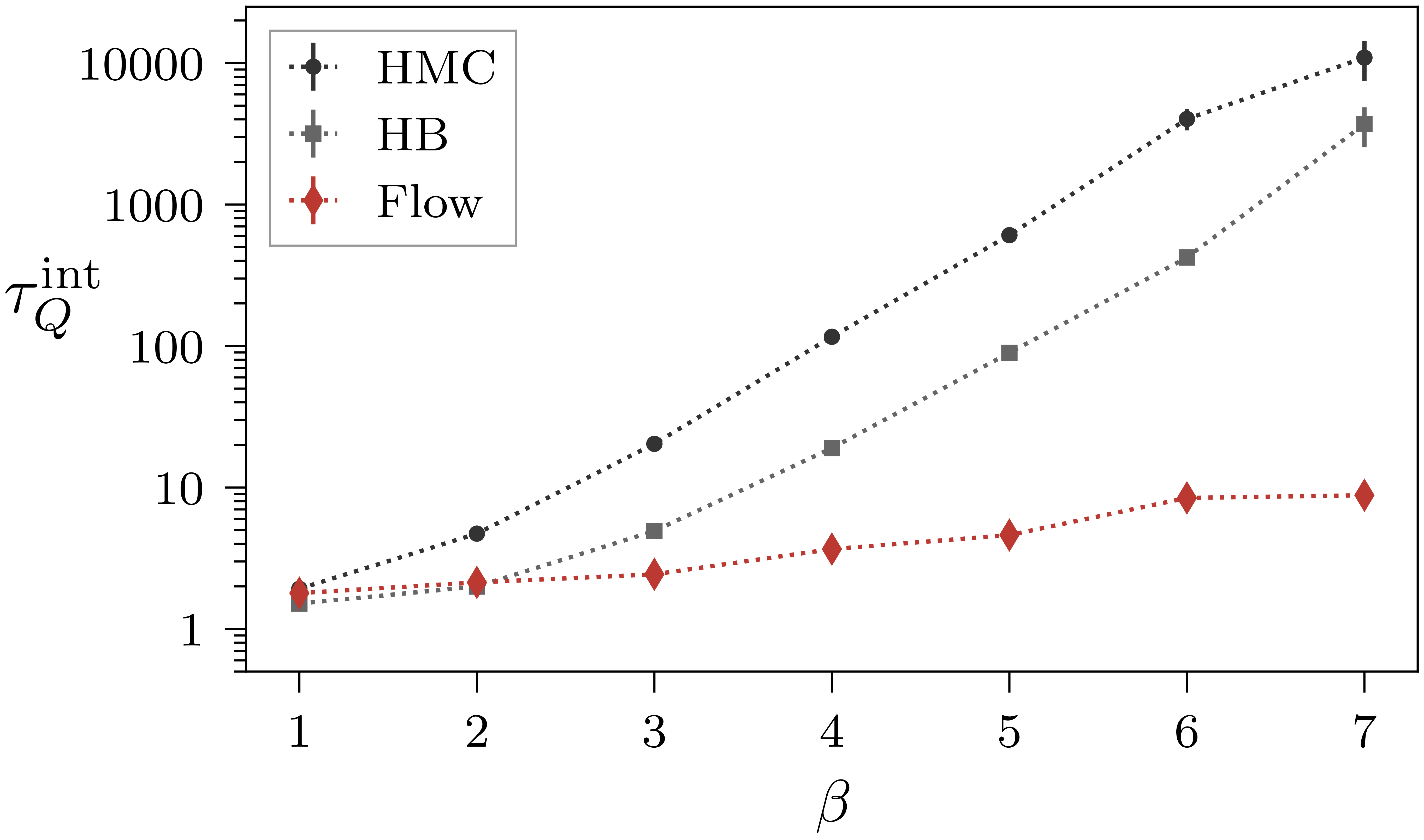
[Hackett, Hsieh, Albergo, Boyda, Chen, Chen, Cranmer, GK, Shanahan; **2107.00734**]

Jupyter notebook tutorial:

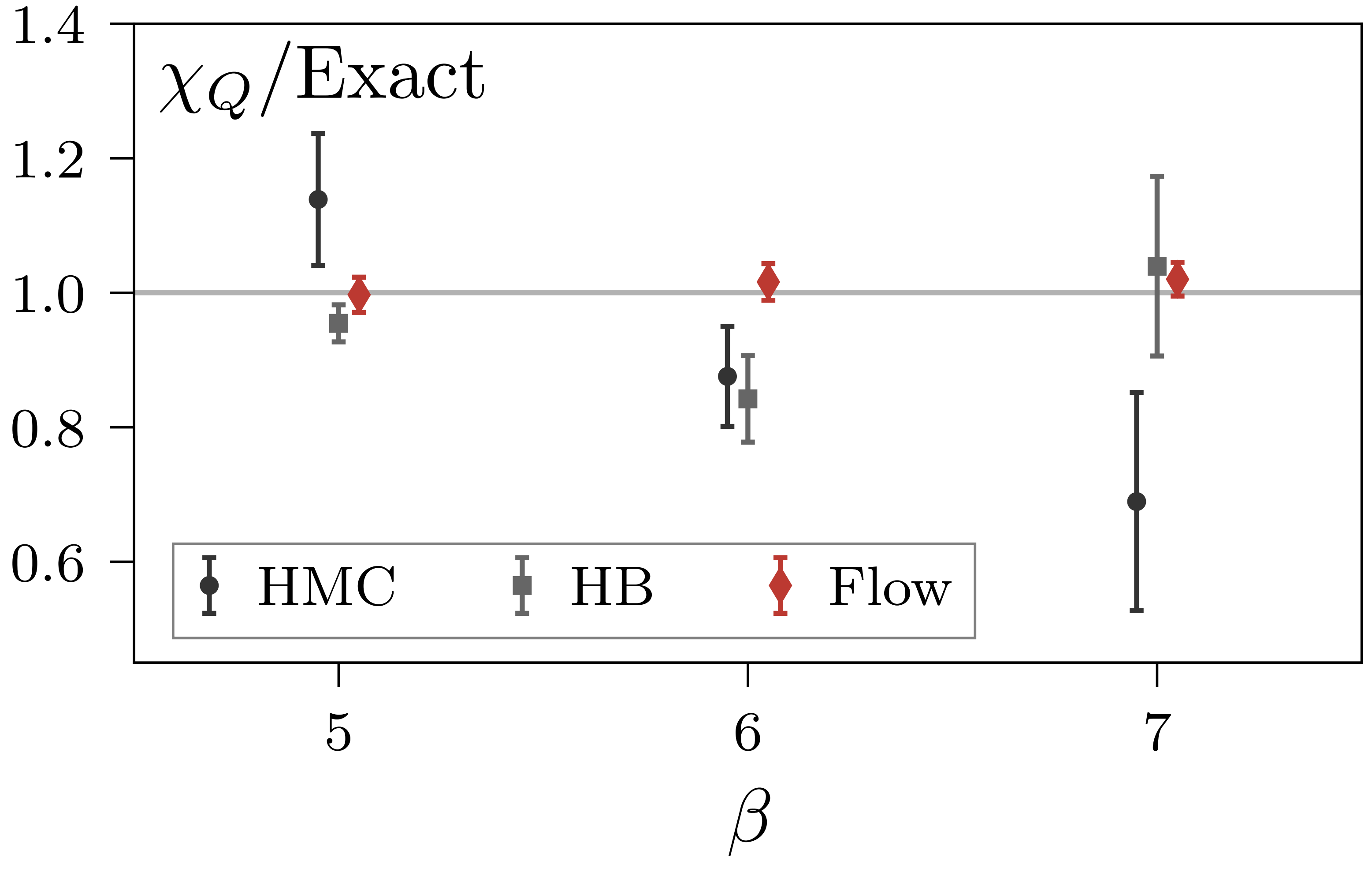
[Albergo, Boyda, Hackett, GK, Cranmer, Racanière, Rezende, Shanahan; **2101.08176**]

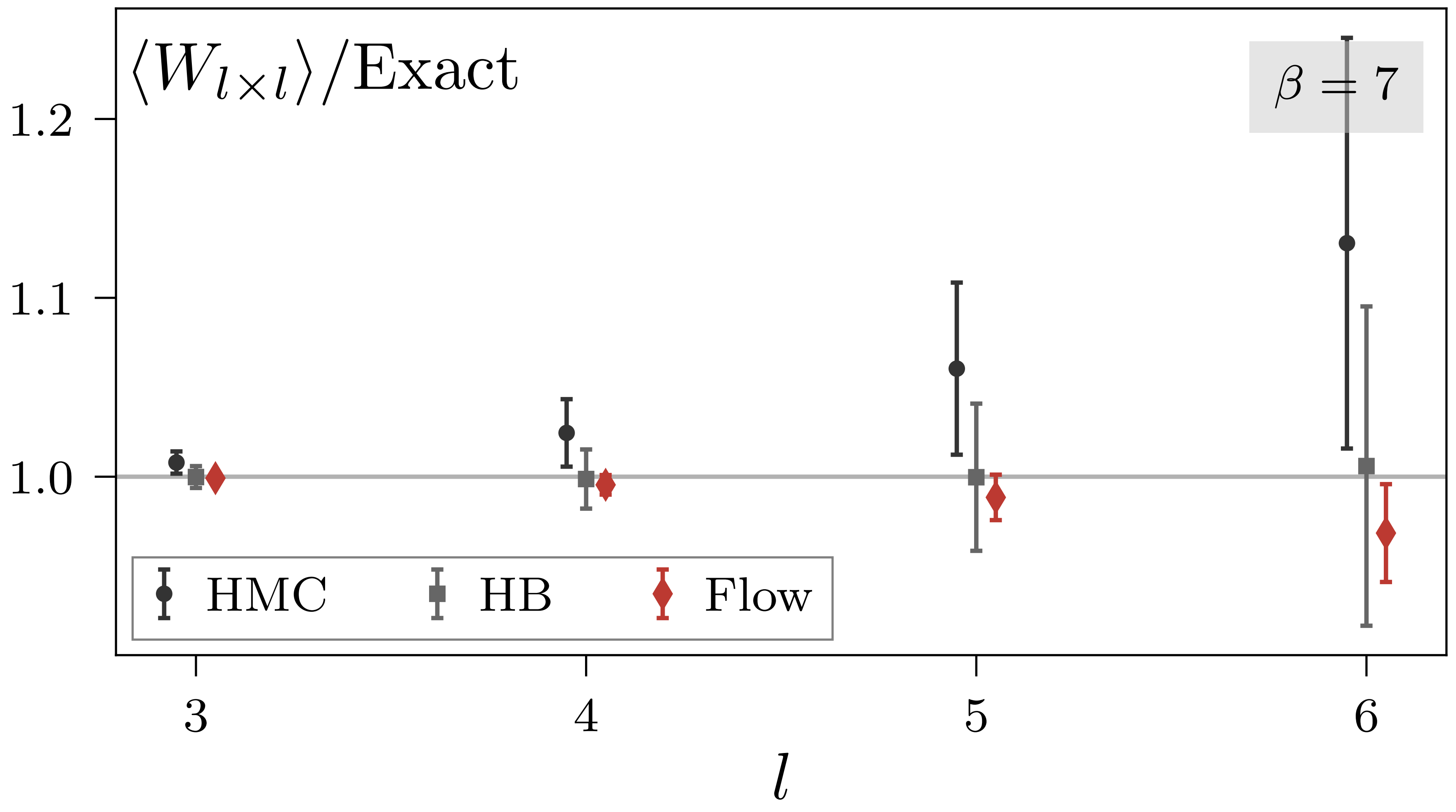
Backup Slides

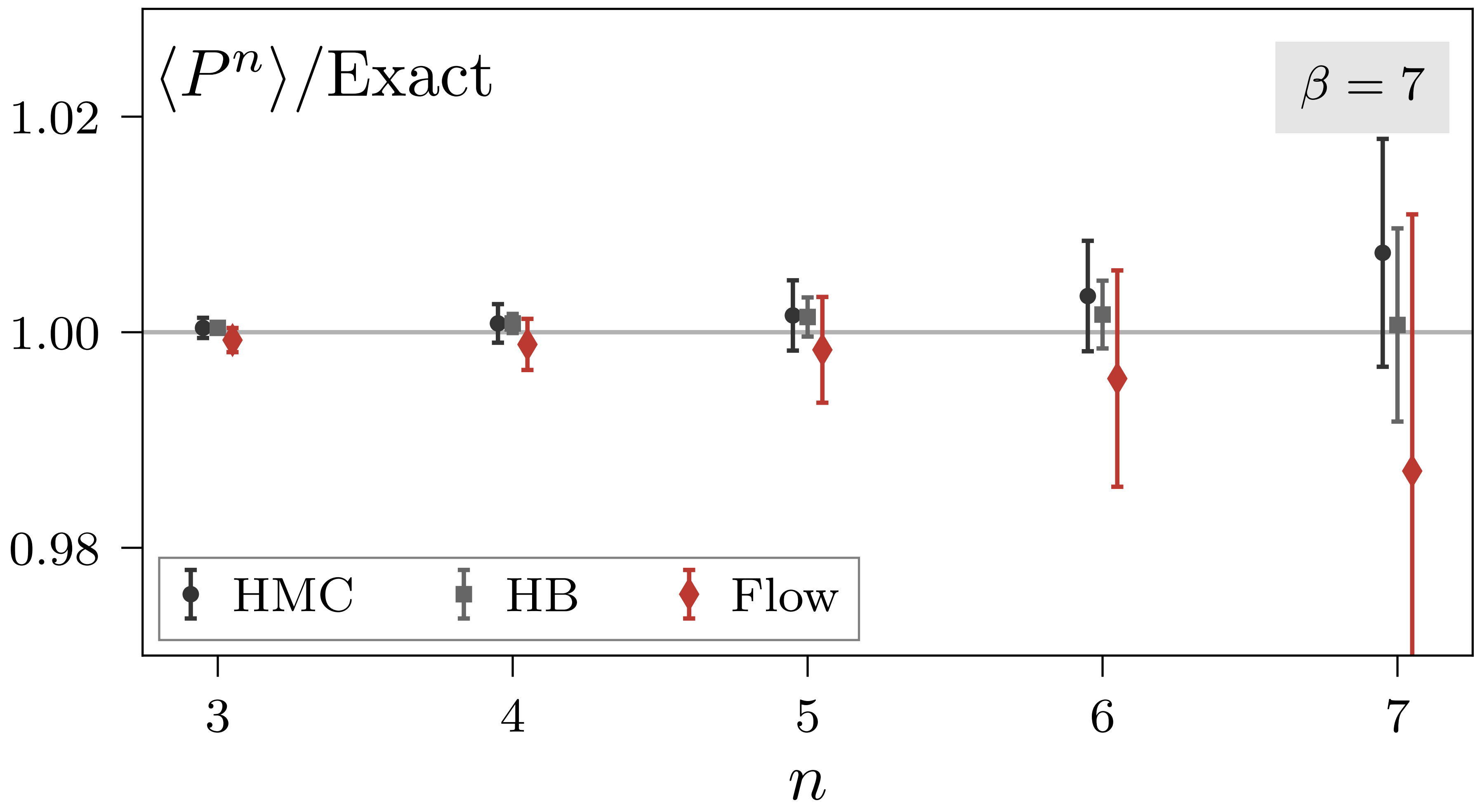
U(1) topological freezing mitigated



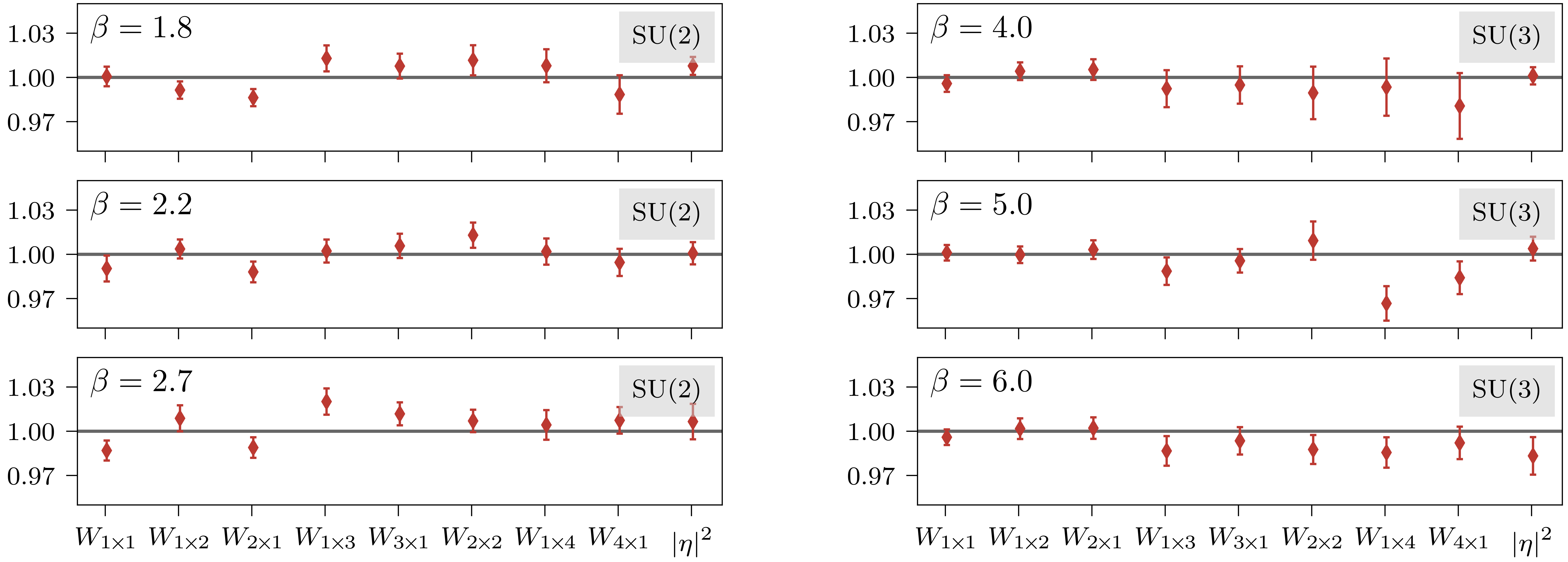
U(1) observables







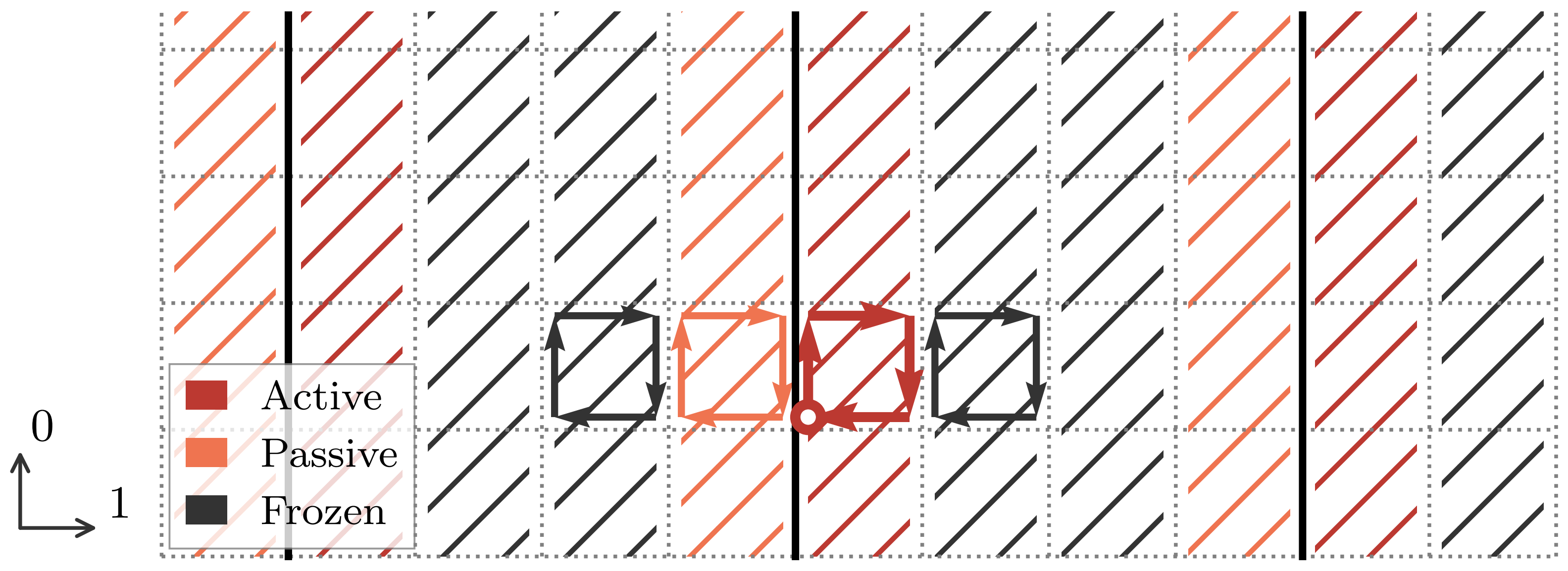
SU(N) observables



Learning SU(2) and SU(3) gauge theory

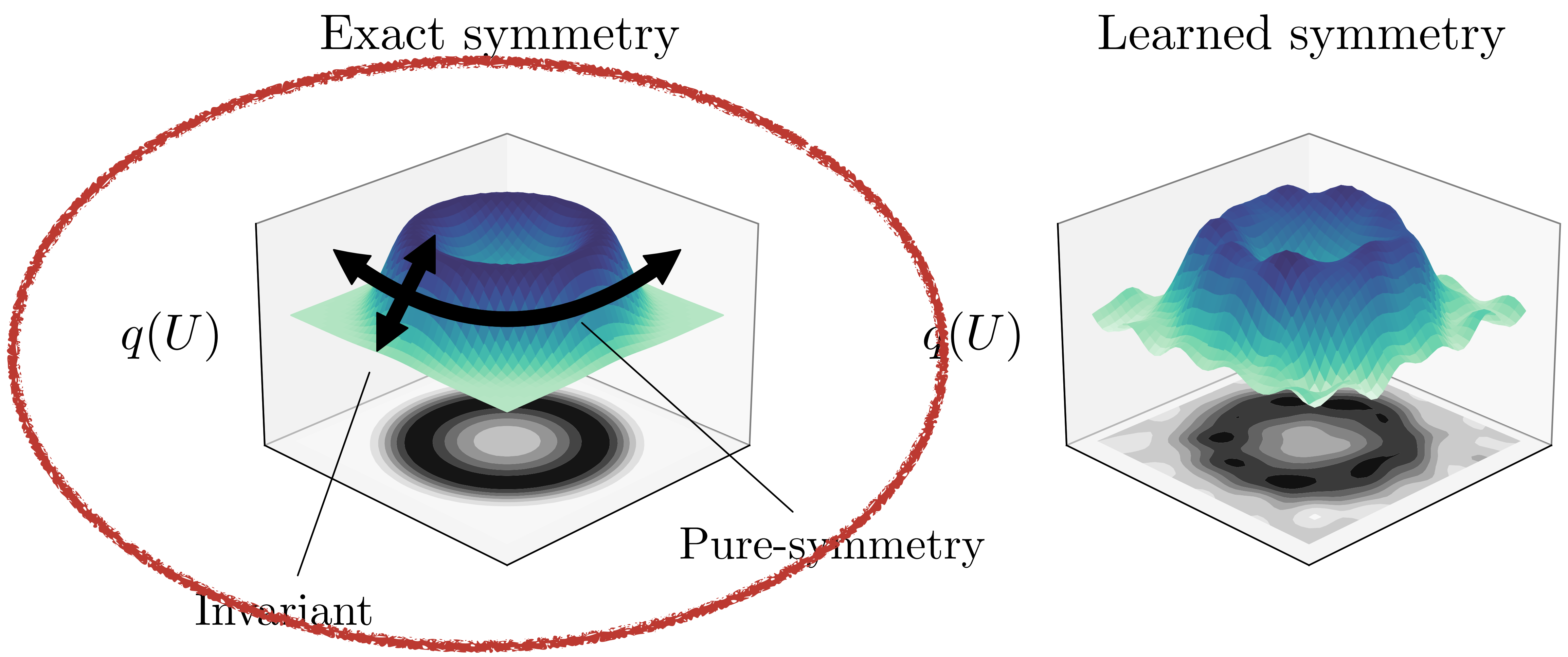
Normalizing flows trained for 2D lattice gauge theory on 16×16 lattices.

- Approx matched 't Hooft couplings, giving
  $\beta = \{1.8, 2.2, 2.7\}$ for $SU(2)$ and
  $\beta = \{4.0, 5.0, 6.0\}$ for $SU(3)$



- 48 **PAFF coupling layers**, update all links 6 times
- No equivalent topo freezing, studied **absolute model quality** instead

All flow-based models exactly gauge-equiv by construction



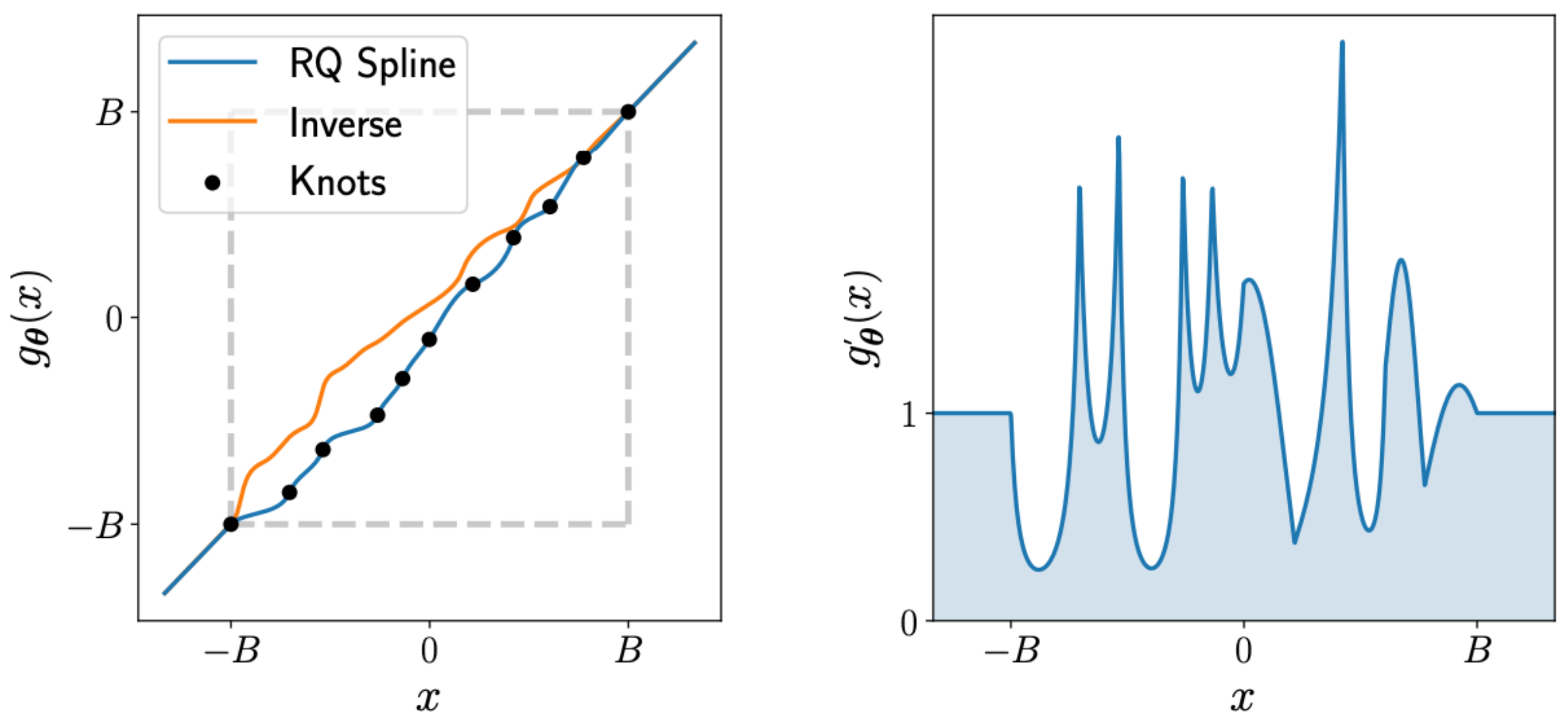
U(1) kernels

Conjugation equivariance trivially satisfied: $h(\Omega W \Omega^\dagger) = h(W) = \Omega h(W) \Omega^\dagger$.

Invertible maps on U(1) variables:

- **Periodic / compact domain** must be addressed.
- For details, see:
  [Rezende, Papamakarios, Racanière, Albergo, GK, Shanahan, Cranmer; **ICML (2020) 2002.02428**]

[Durkan, Bekasov, Murray, Papamakarios **1906.04032**]



Non-compact projection:

- Map $\theta \to x \in \mathbb{R}$, e.g. $\arctan(\theta/2)$
- Transform $x \to x'$ as usual
- Map $x' \to \theta' \in [-\pi, \pi]$

Circular invertible splines:

- Spline "knots" trainable fns
- Identify endpoints π and $-\pi$
- Number of knots $\leftrightarrow$ expressivity

SU(N) kernels: strategy

SU(N) matrix-conj. equivariance is **non-trivial**.

$$h(\Omega W \Omega^\dagger) = \Omega h(W) \Omega^\dagger$$

Useful observations:

- Conjugation only rotates eigenvectors.
- Spectrum is invariant.
- Wilson loop spectrum encodes gauge-invariant physics → **This is what we want to transform.**

Strategy: Invertibly transform only the spectrum of W via a “spectral map”.

Or, “spectral flow”.

[Boyda, GK, Racanière, Rezende, Albergo, Cranmer, Hackett, Shanahan **PRD103 (2021) 074504**]

SU(N) kernels: strategy

SU(N) matrix-conj. equivariance is **non-trivial**.

$$h(\Omega W \Omega^\dagger) = \Omega h(W) \Omega^\dagger$$

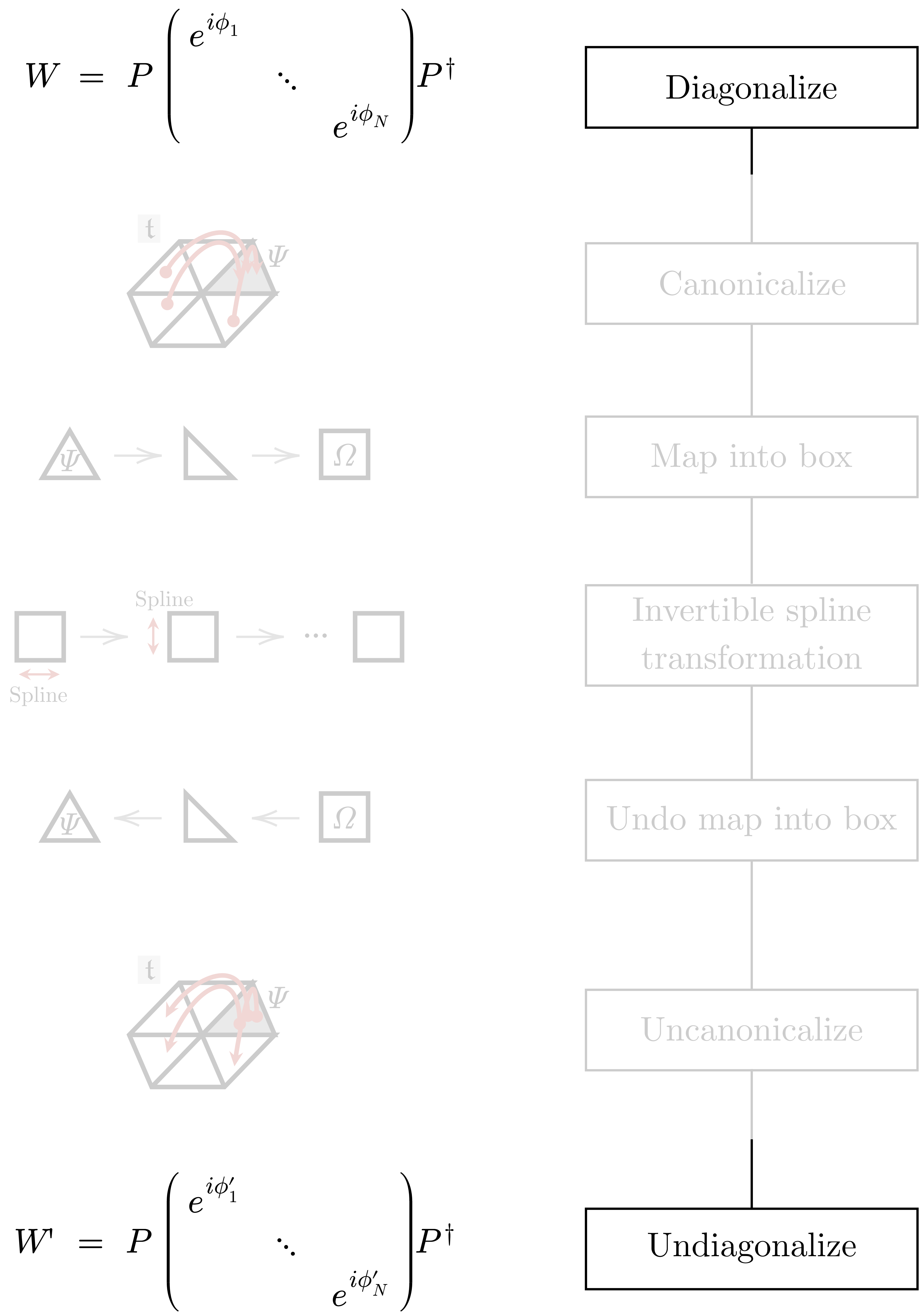
Useful observations:

- Conjugation only rotates eigenvectors.
- Spectrum is invariant.
- Wilson loop spectrum encodes gauge-invariant physics → **This is what we want to transform.**

Strategy: Invertibly transform only the spectrum of W via a "spectral map".

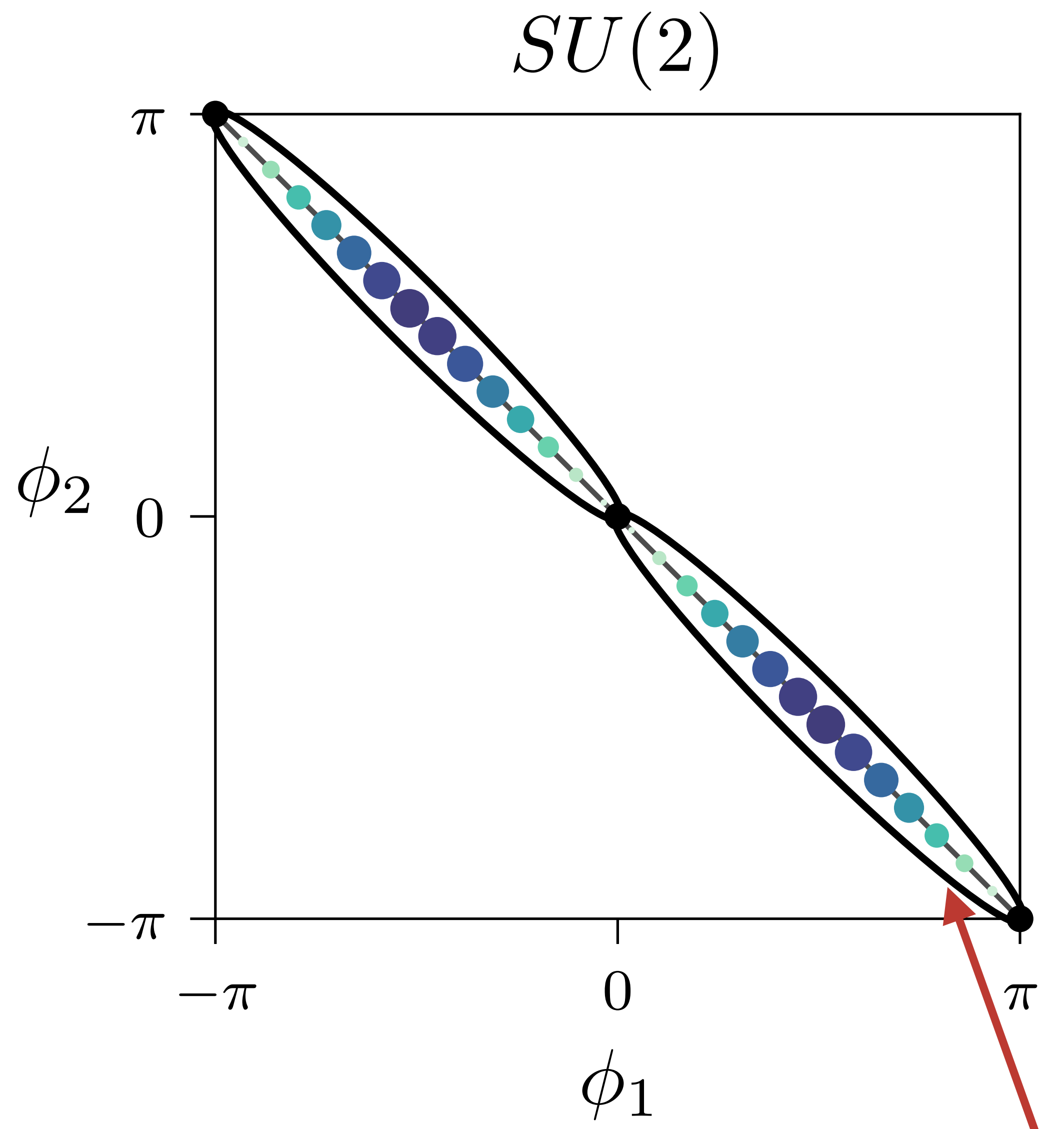
Or, "spectral flow".

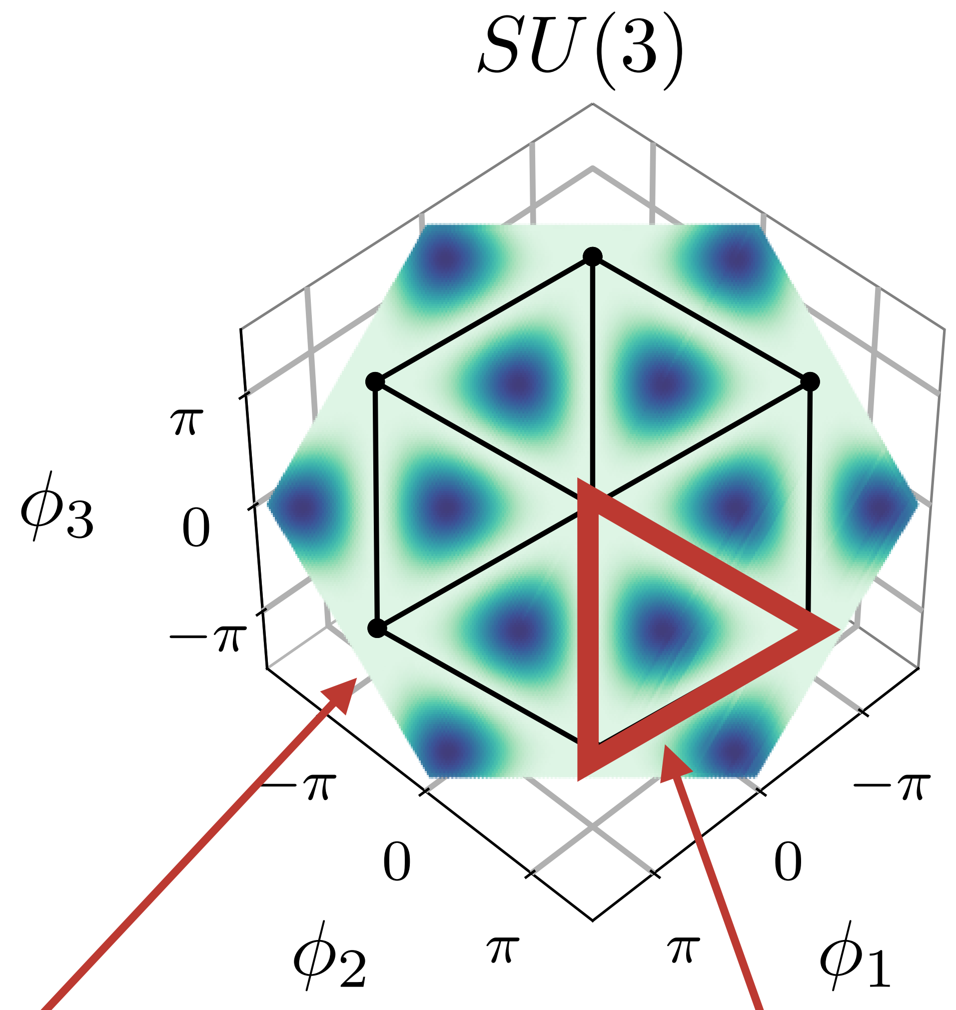
[Boyda, GK, Racanière, Rezende, Albergo, Cranmer, Hackett, Shanahan **PRD103 (2021) 074504**]



SU(N) kernels: Permutation equivariance

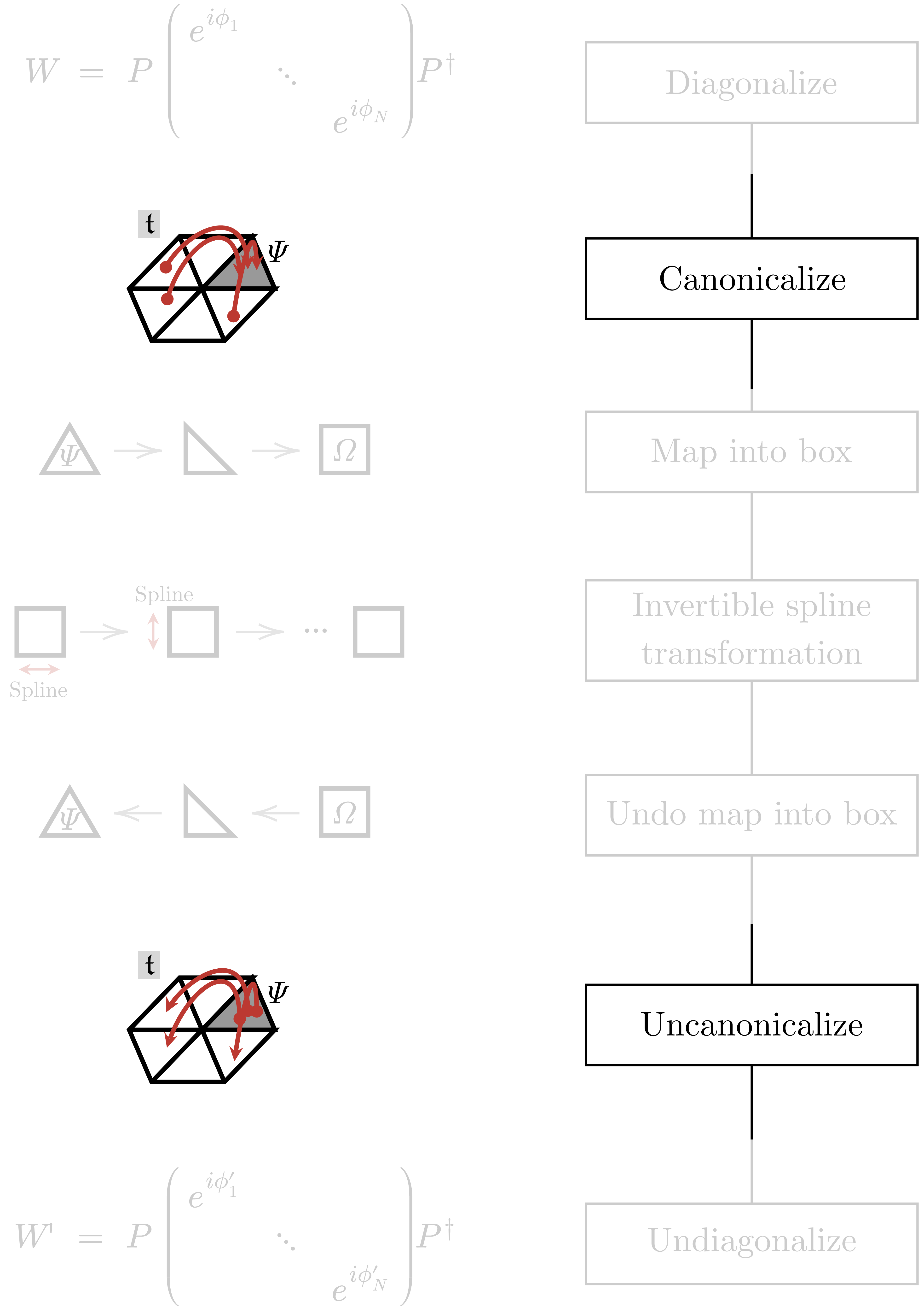
See also [J. Thaler, Wed] for perm-inv NNs





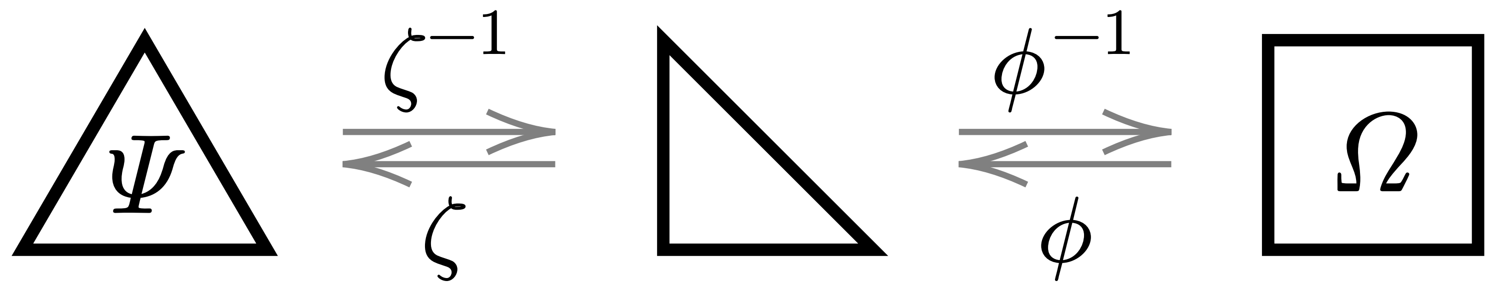
Sub-manifold of $\det = 1$ eigenvalues

"Cell", related by perms of eigenvalues to other cells.



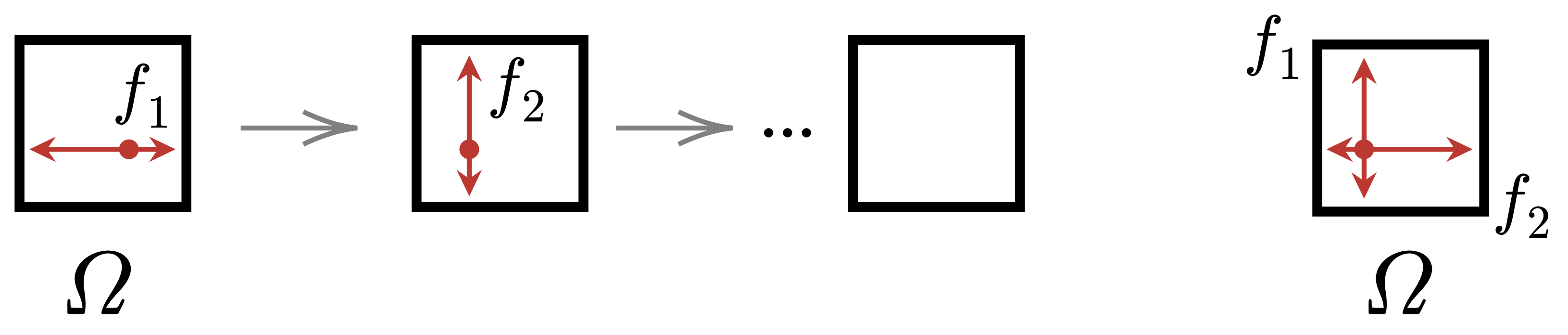
SU(N) kernels:
Transform the canonical cell

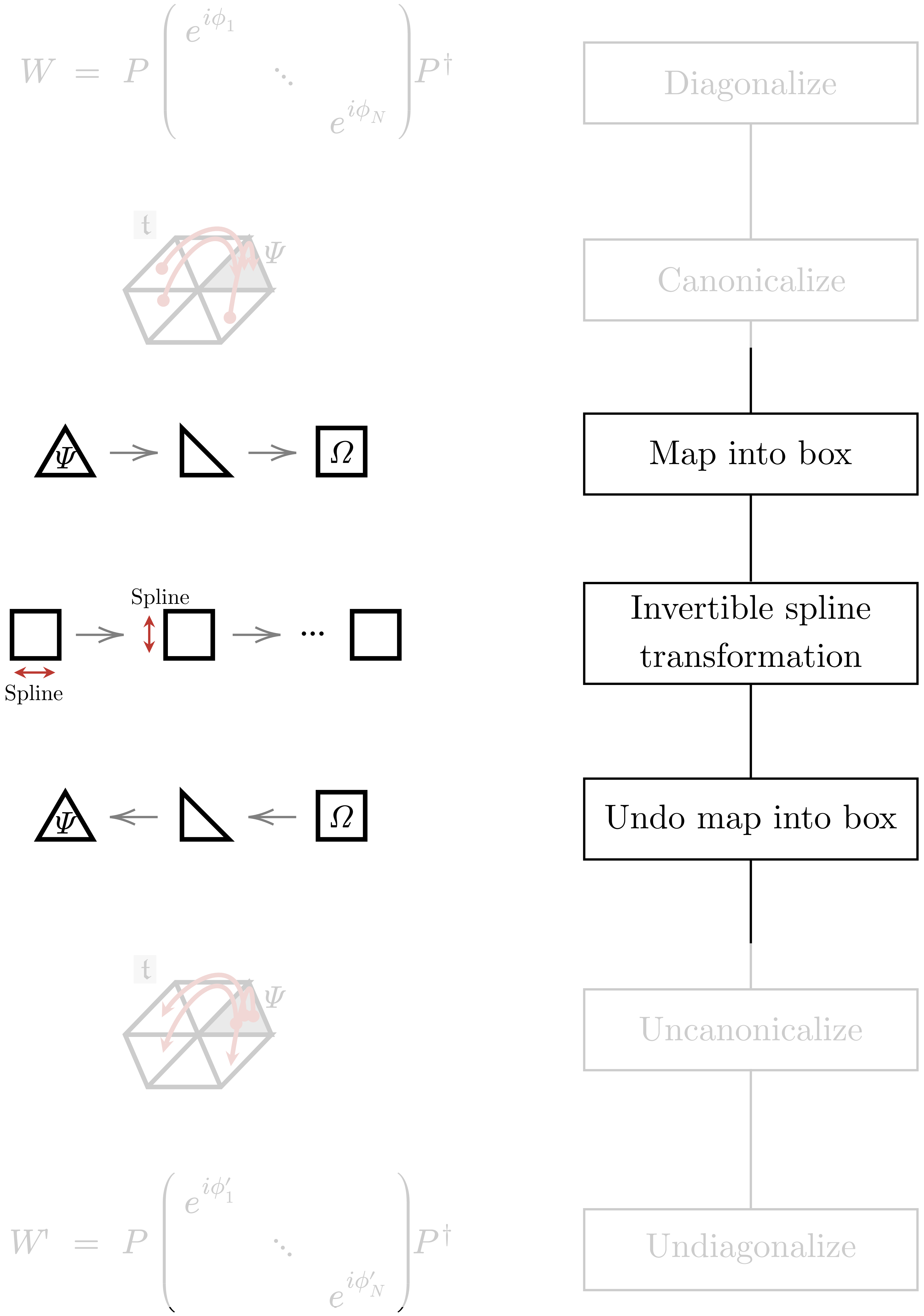
Change variables to rectilinear box Ω



Transform by acting on coords of box Ω, either…

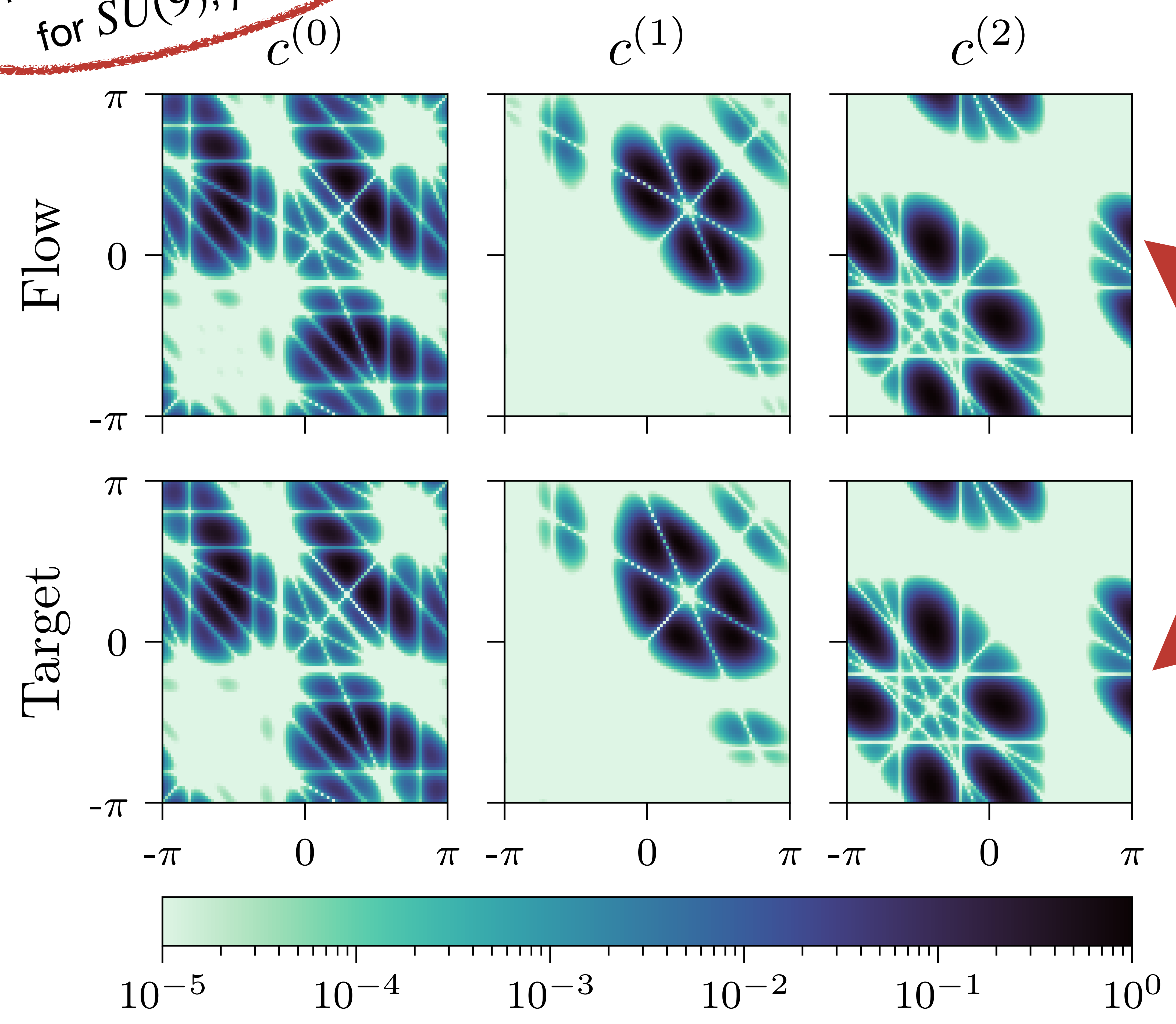
Autoregressive … or … Independent





Testing SU(N) kernels

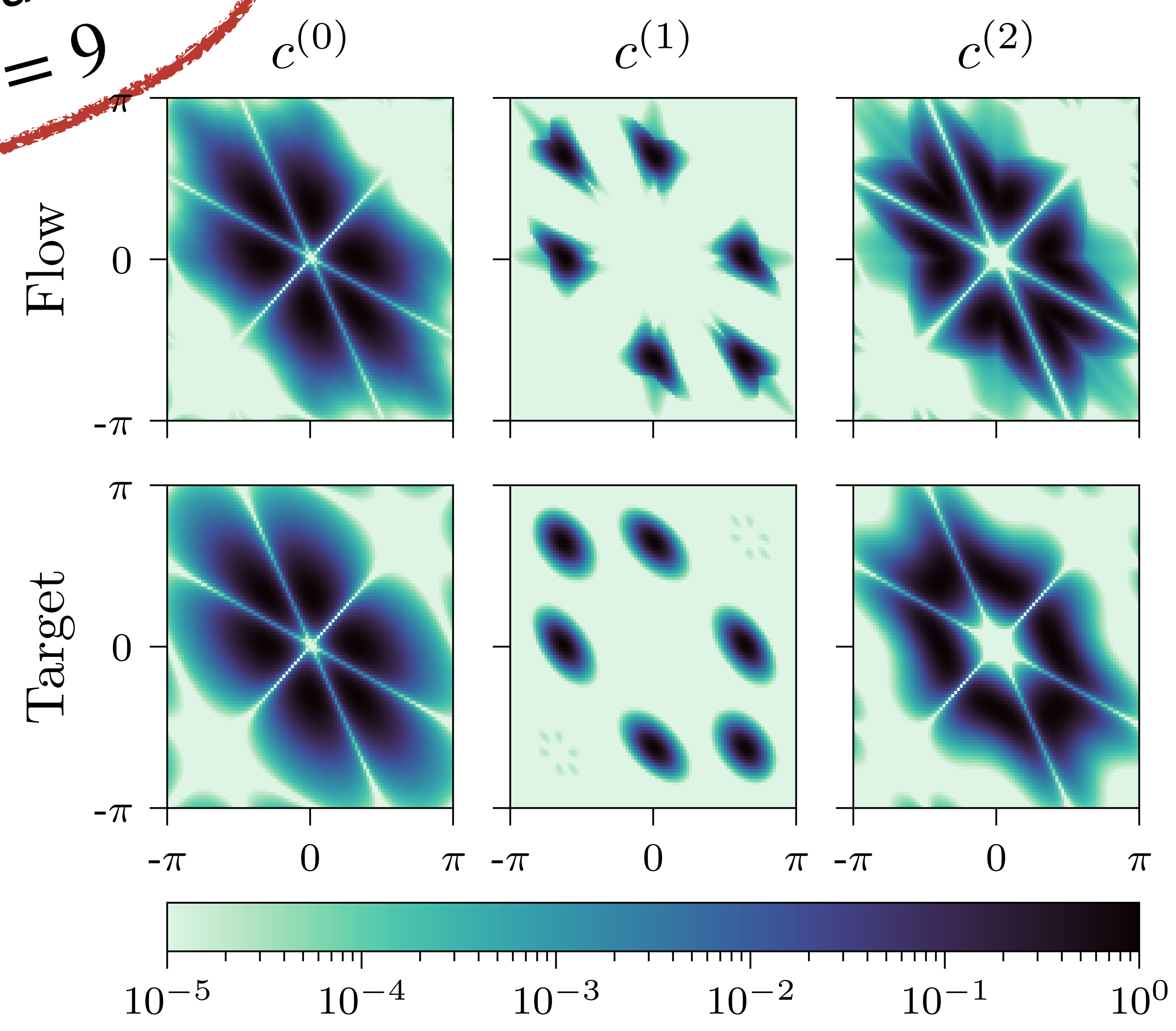
Plaquette distributions for $SU(9)$, $\beta = 9$



Agree!

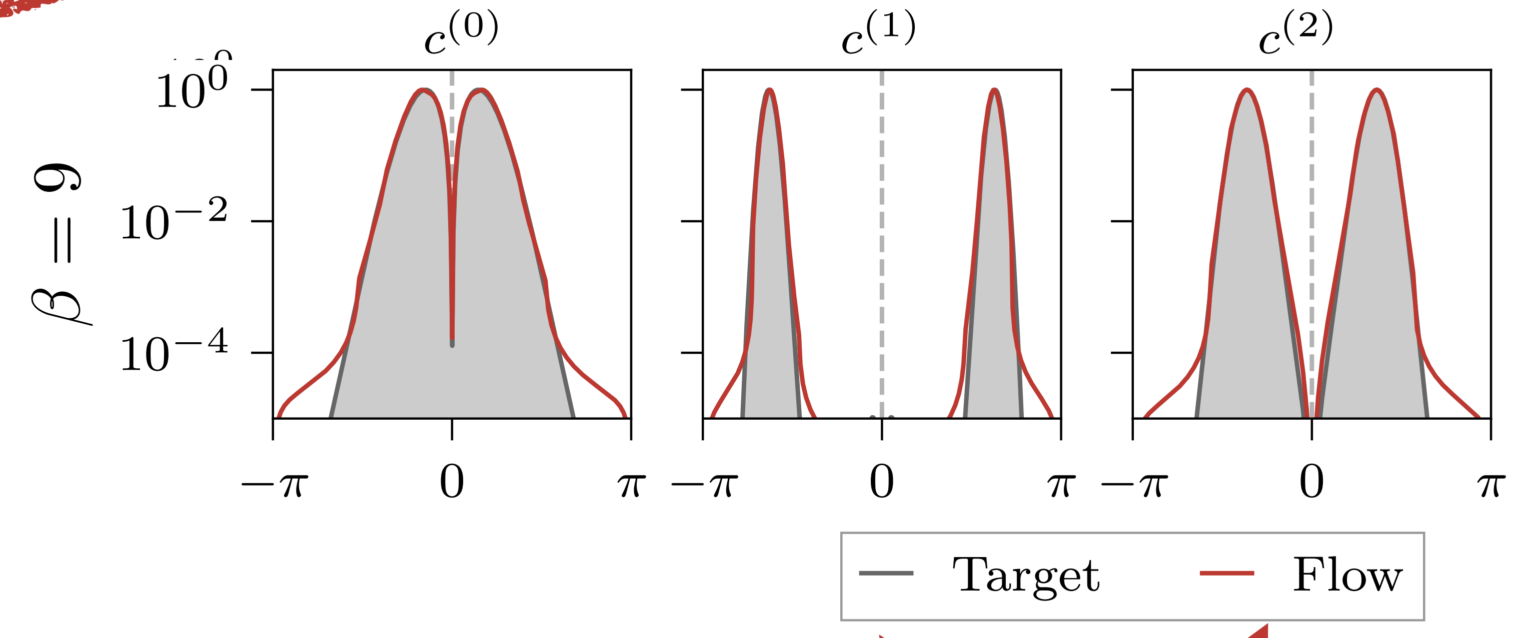
Density has zeros on vertical, horizontal, and diagonal lines where the slice crosses walls of cells

Plaquette distributions for $SU(3)$, $\beta = 9$



Agree!

Plaquette distributions for $SU(2)$, $\beta = 9$

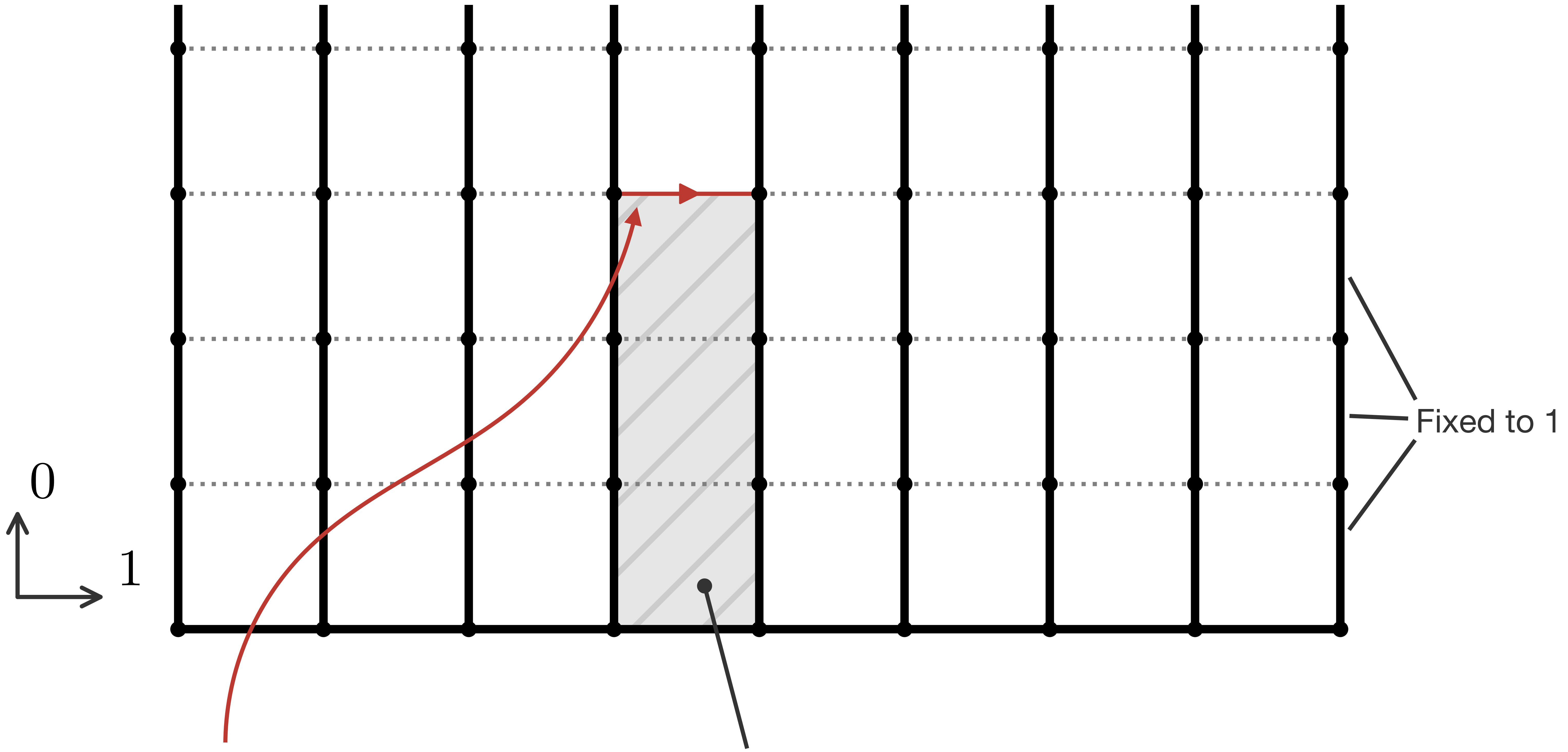


Agree!

Gauge fixing?

Where gauge DoFs are explicitly factored out, e.g. maximal tree

Explicit gauge fixing is at odds with **translational symmetry** + **locality**



Link physically encodes **Wilson loop** around shaded region

Gauge fixing?

Where gauge DoFs are fixed by solving
a constraint, e.g. Landau gauge

Implicit gauge fixing difficult to act on via **flow-based models**

Landau gauge: $U_\mu^{\text{fix}}(x) = \text{argmin}_{U^\Omega} \sum_x \sum_{\mu=1}^{N_d} \text{ReTr}[U_\mu^\Omega(x)]$

Coulomb gauge: $U_\mu^{\text{fix}}(x) = \text{argmin}_{U^\Omega} \sum_x \sum_{\mu=1}^{N_d-1} \text{ReTr}[U_\mu^\Omega(x)]$

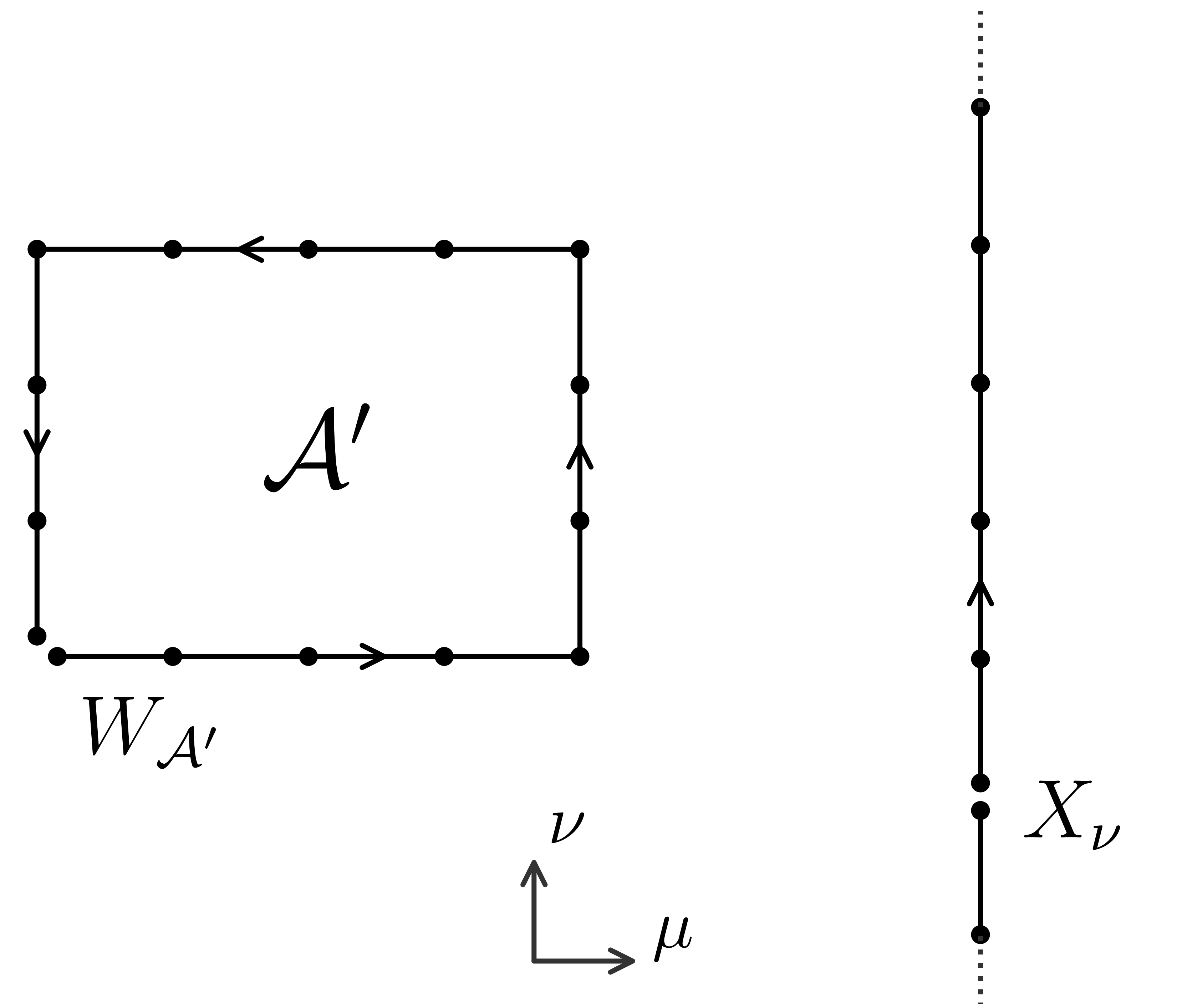
Unclear how to invertibly transform $U_\mu^{\text{fix}}(x)$.

Center symmetry

Using **only contractible loops** in coupling layers enforces center symmetry.

Fundamental fermions:

- Center symmetry explicitly broken
- Must include non-contractible loops (e.g. Polyakov) in the set of frozen and/or transformed loops



Exactness: Reweighting

- Also possible to reweight independently drawn samples:

$$\langle \mathcal{O} \rangle = \frac{\int \mathcal{D}U \, q(U) \left[\mathcal{O}(U) \frac{p(U)}{q(U)} \right]}{\int \mathcal{D}U \, q(U) \left[\frac{p(U)}{q(U)} \right]}$$

- May be preferable when observables $\mathcal{O}(U)$ are efficiently computed, and sampling is expensive.

- Observables $\mathcal{O}(U)$ are expensive in lattice QCD. We prefer resampling or MCMC approaches in these settings.

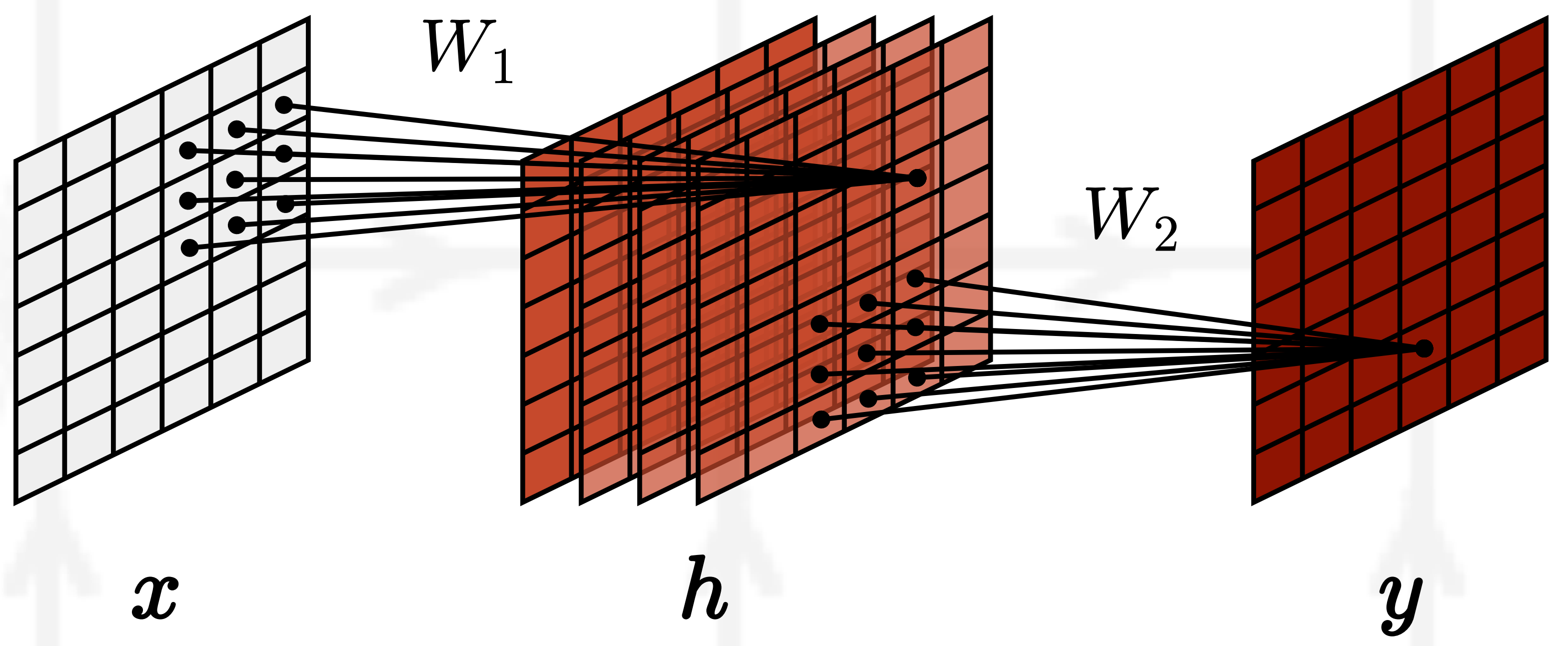
Translational equivariance

1. Make context functions Convolutional Neural Nets:

- Compute output value for each site from linear transform of nearby DOF only

- Reuse same weights, scanning kernel across the lattice

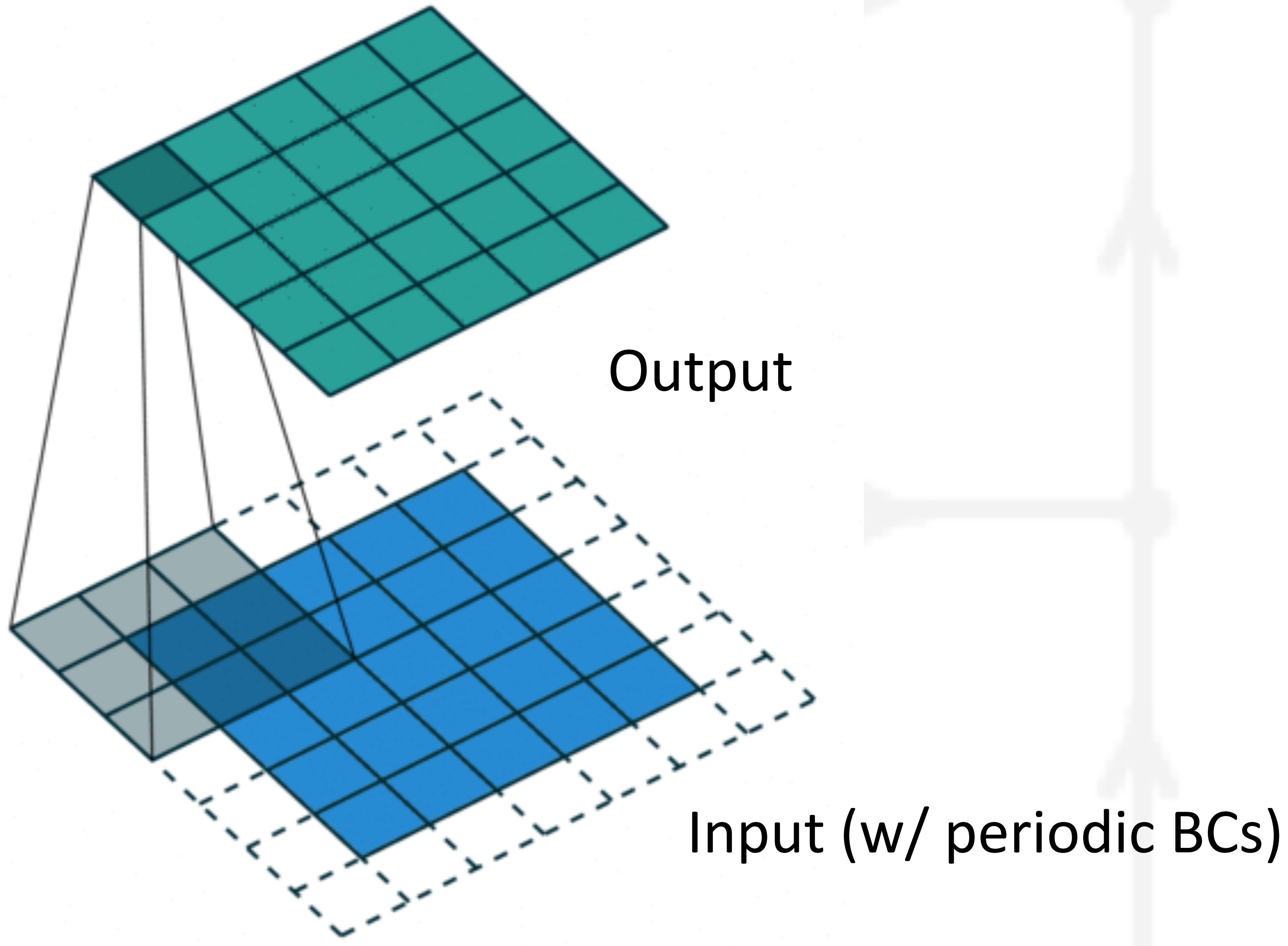
CNNs are equivariant under translations.



2. Make masking pattern (mostly) translationally invariant.

- E.g. checkerboard is symmetric modulo $\mathbb{Z}_2$ even/odd

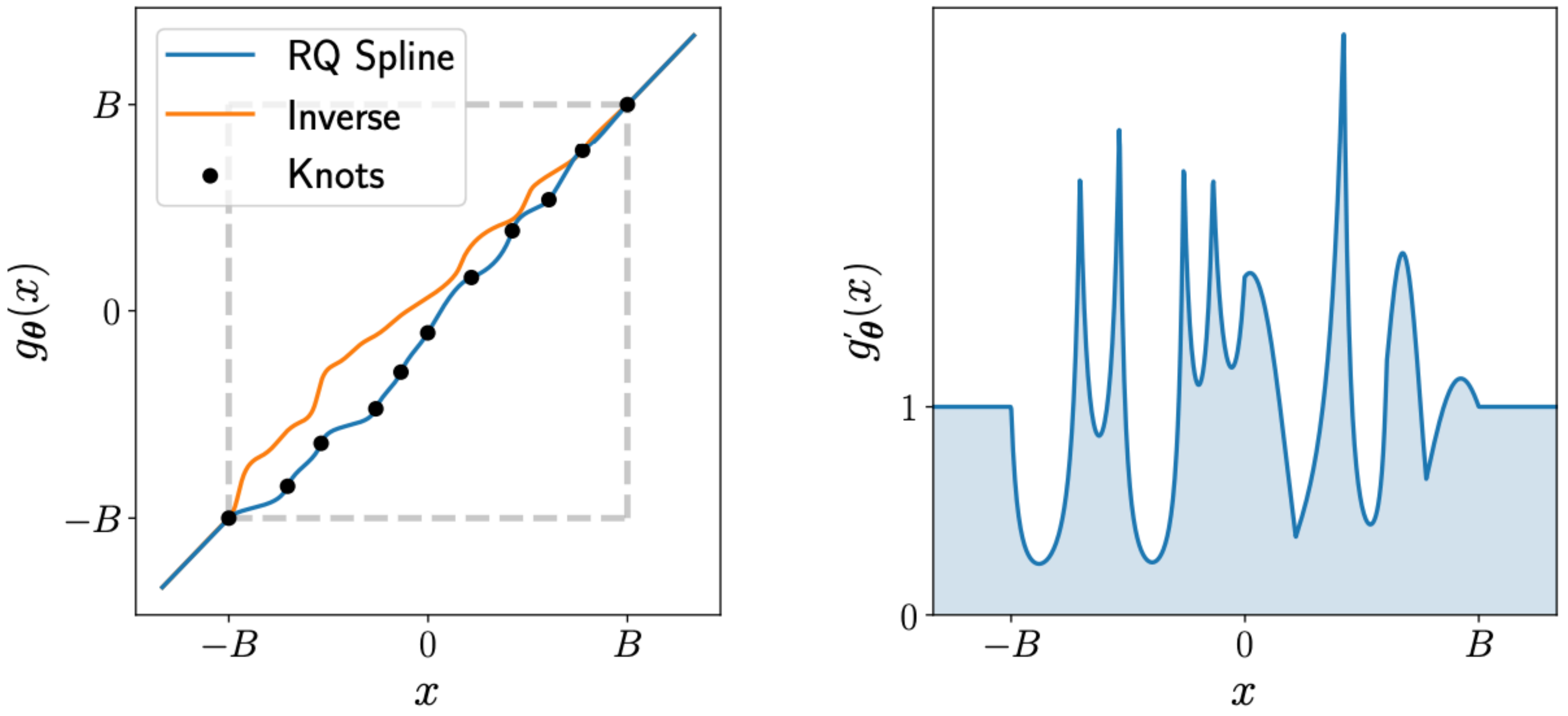
- Gauge theory: translational equiv modulo $\mathbb{Z}_4 \times \mathbb{Z}_4$



[freecodecamp.org]

Details of $SU(2)$ models

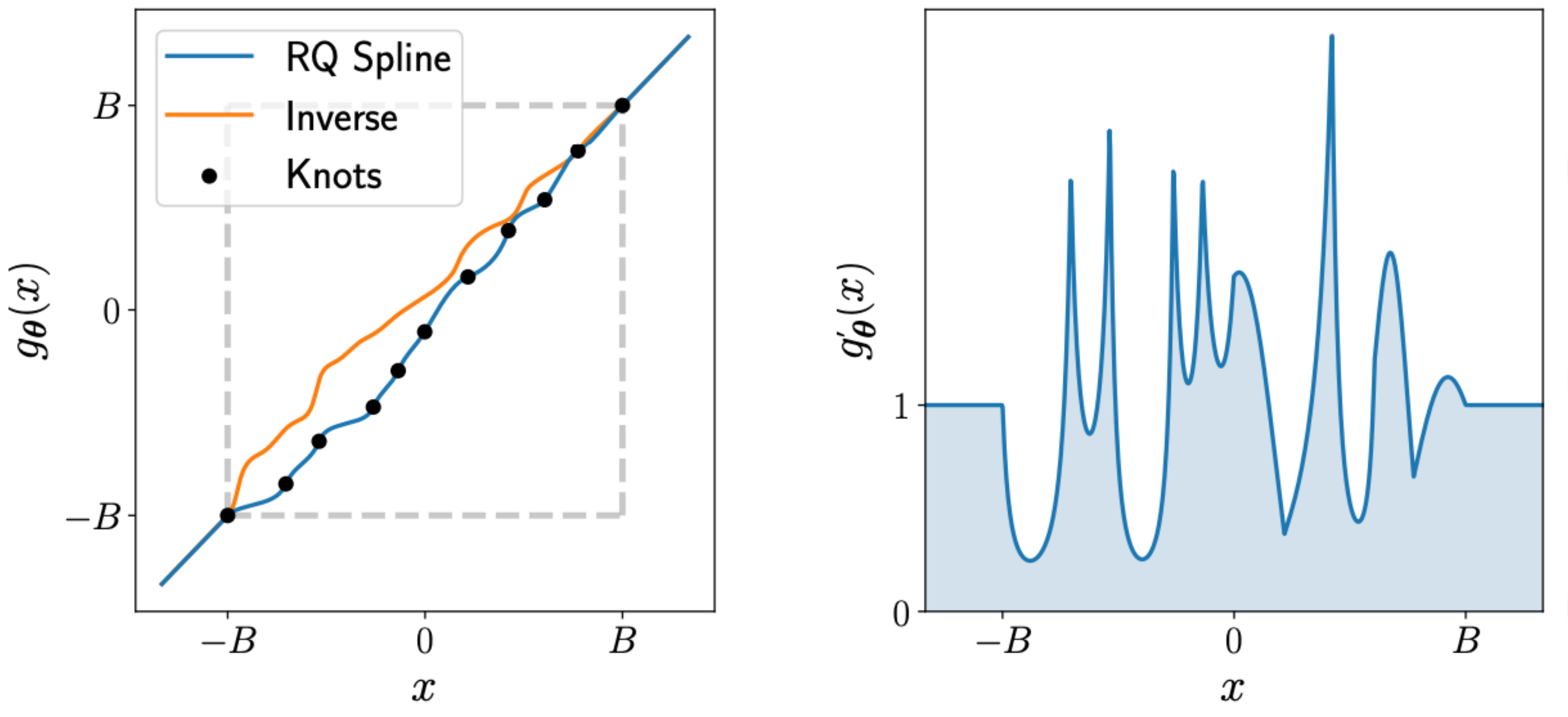
- Inner flow on open box Ω is a spline flow with **4 knots**
  - B and $-B$ boundaries align to 0 and 1 edges of the open box
- CNNs to compute the knot locations
  - 32 hidden channels
  - 2 hidden layers



[Durkan, Bekasov, Murray, Papamakarios 1906.04032]

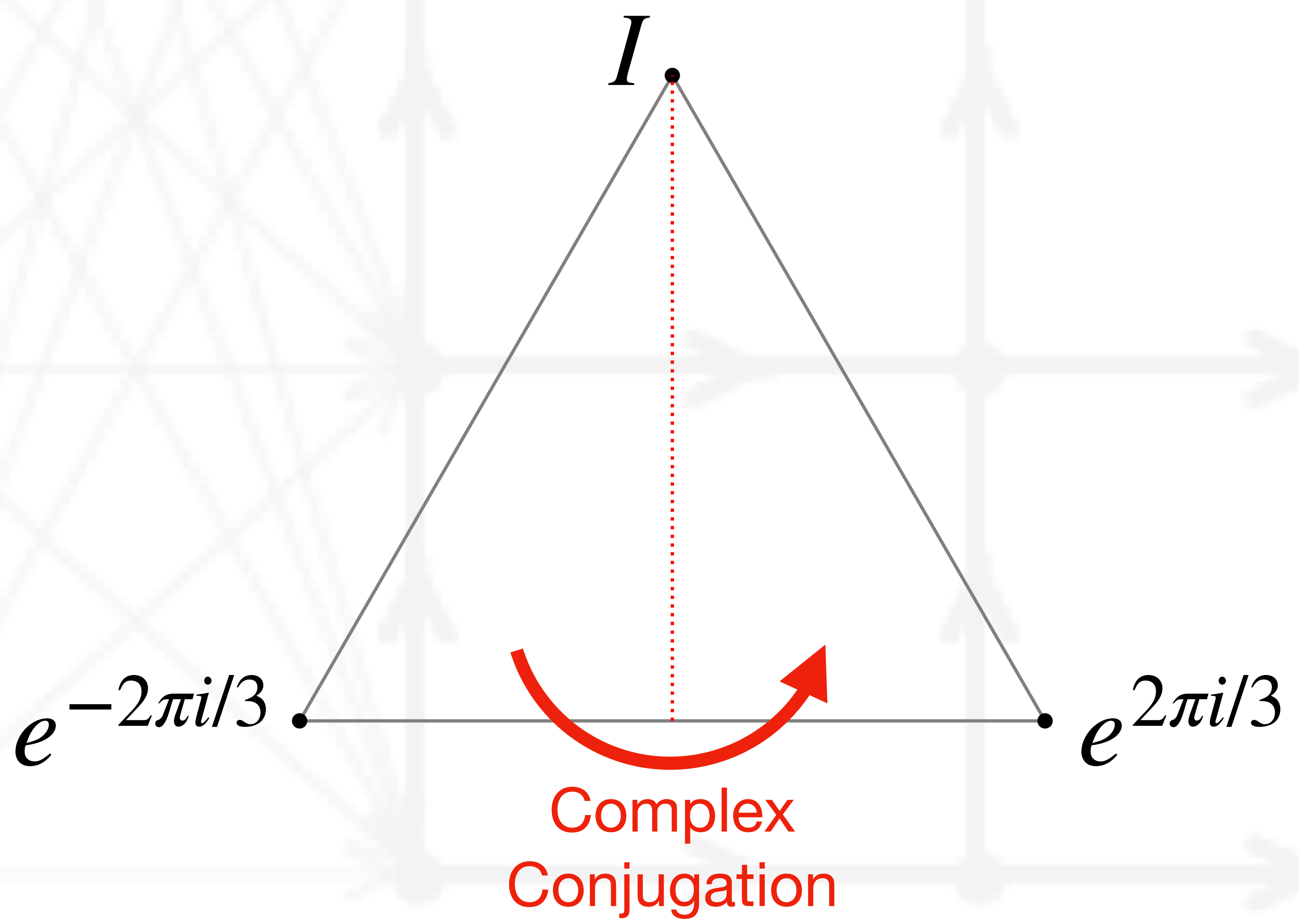
Details of $SU(3)$ models

- Inner flow on open box Ω is a spline flow with **16 knots**
  - B and $-B$ boundaries align to 0 and 1 edges of the open box



[Durkan, Bekasov, Murray, Papamakarios 1906.04032]

- CNNs to compute the knot locations
  - 32 hidden channels
  - 2 hidden layers
- Exact conjugation equivariance also imposed

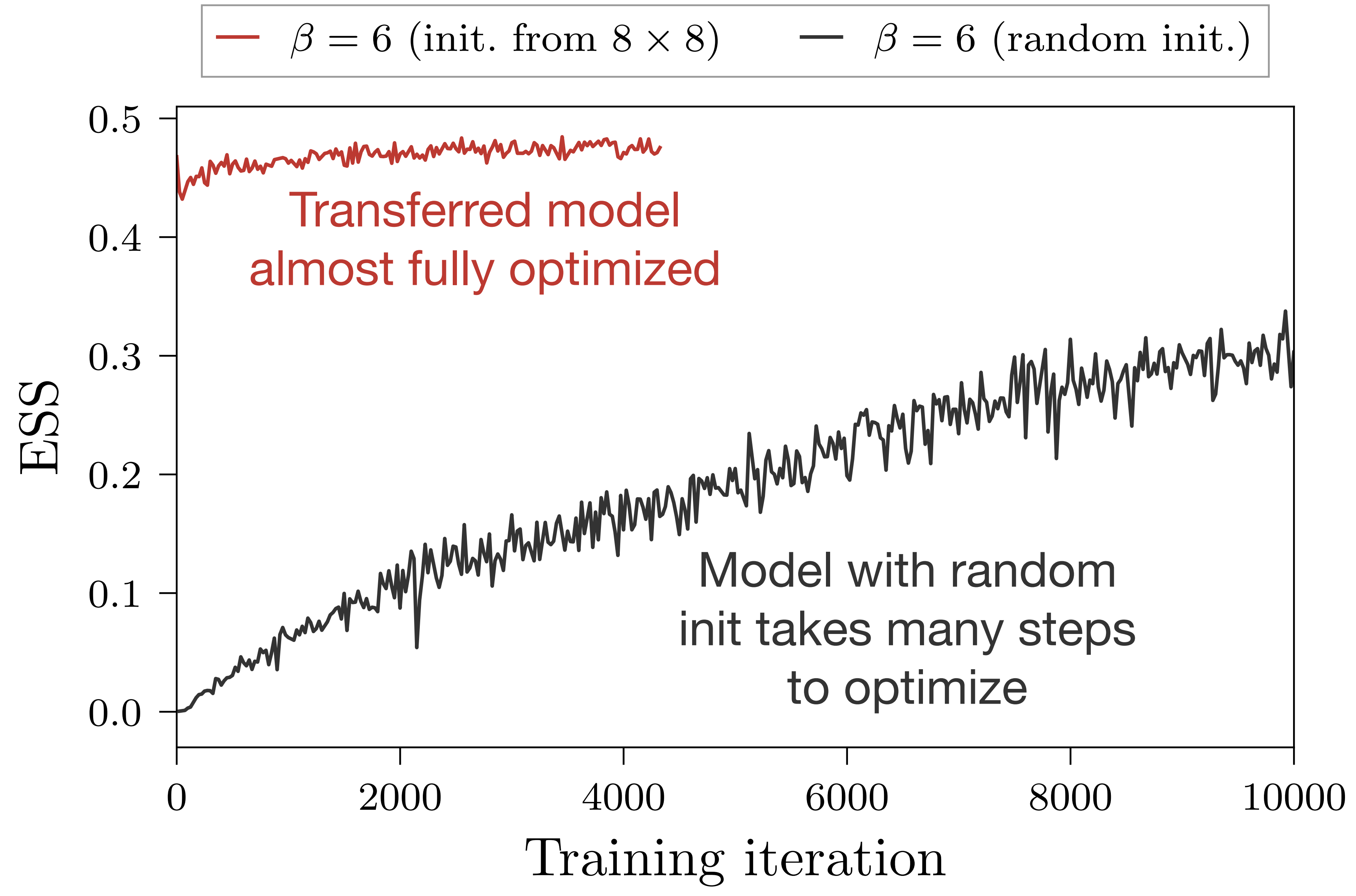


Gauge theory model training

- Adam optimizer ~ stochastic grad. descent with momentum
  - Batches of size 3072 per gradient descent step
  - Monitored value of **effective sample size** (ESS)

$$\mathrm{ESS} = \frac{\left(\frac{1}{n}\sum_i w(U_i)\right)^2}{\frac{1}{n}\sum_i w(U_i)^2}, \quad U_i \sim q(U)$$

$w(U) = p(U)/q(U)$ "reweighting factors"



- **Transfer learning:** model trained first on 8×8 then used to initialize model for training on 16×16