ChatJesseT

Type your question below.

Ask

Question: *write your prospective on Probing the Frontiers of Nuclear Physics with AI at EIC*

As a theoretical physicist and the Director of the NSF AI Institute for Artificial Intelligence and Fundamental Interactions (IAIFI), I am thrilled to share my perspective on probing the frontiers of nuclear physics with AI at the Electron-Ion Collider (EIC). The EIC is an exciting new facility that will allow us to study the structure and behavior of atomic nuclei in unprecedented detail. With its high-energy electron and ion beams, the EIC will enable us to explore the mysteries of the strong nuclear force and the fundamental particles that make up atomic nuclei. Now, how does AI fit into this picture? Well, AI has the potential to revolutionize the way we analyze and interpret the vast amounts of data that will be produced by the EIC. Just like a particle physicist sifting through mountains of data from the Large Hadron Collider, AI algorithms can help us extract meaningful patterns and insights from the complex data landscape of nuclear physics. Imagine a team of AI algorithms, armed with neural networks and deep learning techniques, working tirelessly to unravel the secrets of the strong nuclear force. These algorithms can help us identify rare events, classify different types of nuclear reactions, and even predict the properties of exotic nuclear states that have never been observed before. But AI is not just a tool for data analysis. It can also assist in the design and optimization of experiments at the EIC. By simulating and modeling different scenarios, AI algorithms can help us identify the most promising avenues for discovery, saving time and resources in the process

Ask Jesse Thaler AI bot at https://chatjesset.com/

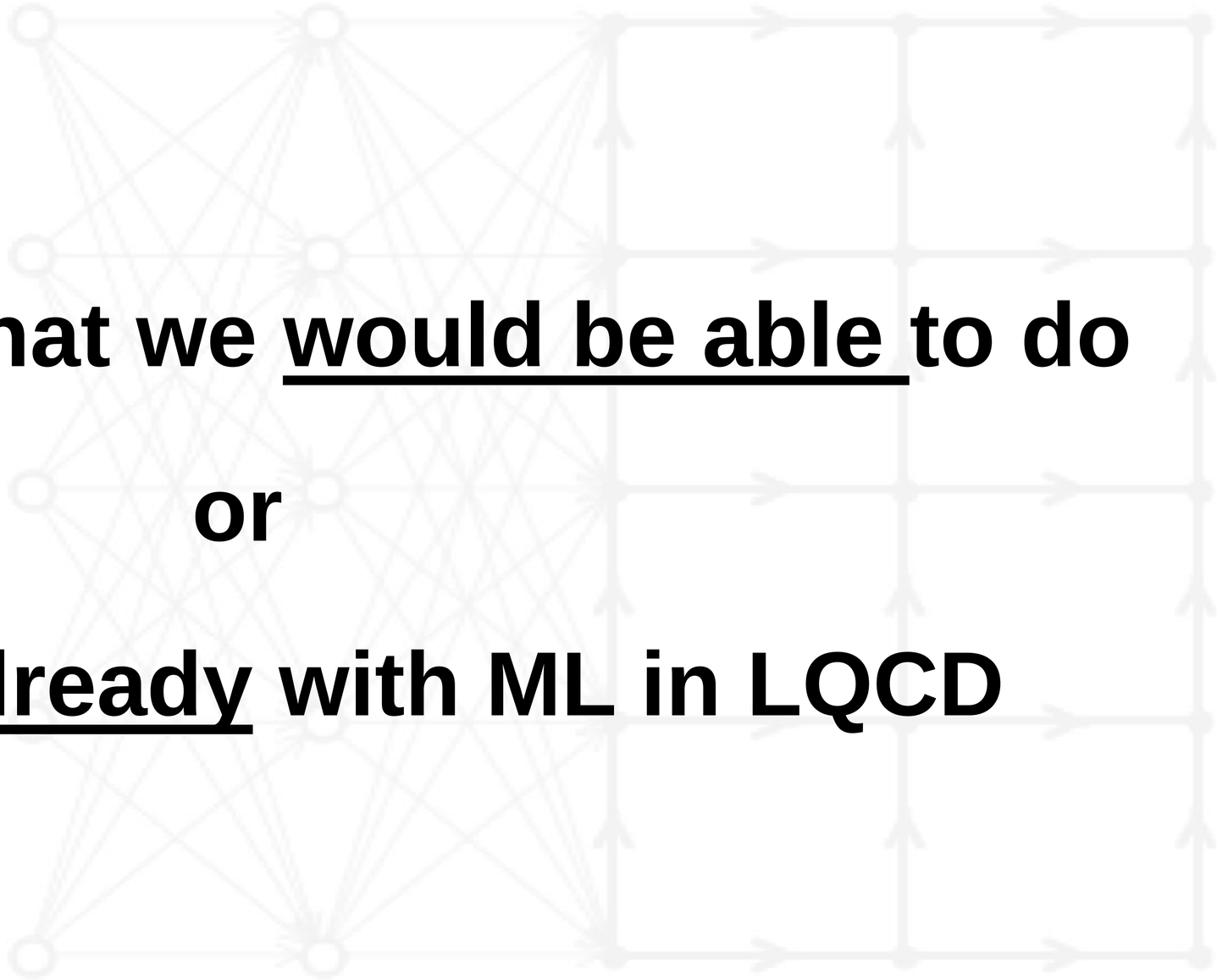
NSF AI Institute for
Artificial Intelligence & Fundamental Interactions

Developing expressive ML models for scaling lattice QCD

Workshop: Probing the Frontiers of Nuclear Physics with AI at the EIC
September 25-26, 2023, @Stony Brook University

Denis Boyda
IAIFI, MIT

On behalf of collaboration: P. Shanahan(MIT), D. Hackett(MIT), F. Romero-López(MIT), S. Racanière(DM), D. Rezende(DM), A. Razavi(DM), A. Botev(DM), G. Kanwar(UB), K. Cranmer(NYU), M. Albergo(NYU)



Examples of what we would be able to do

or

we can do already with ML in LQCD

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

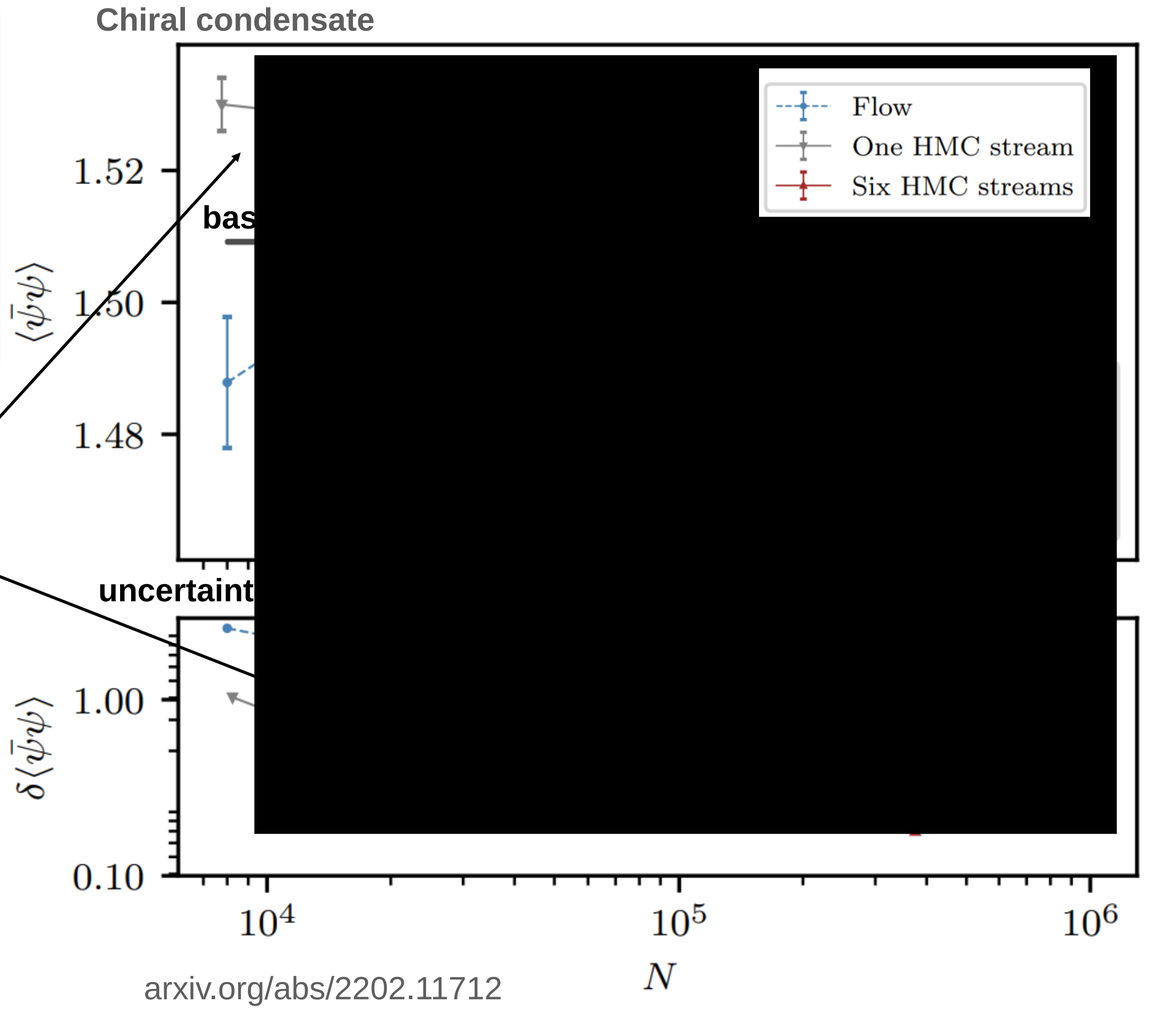
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

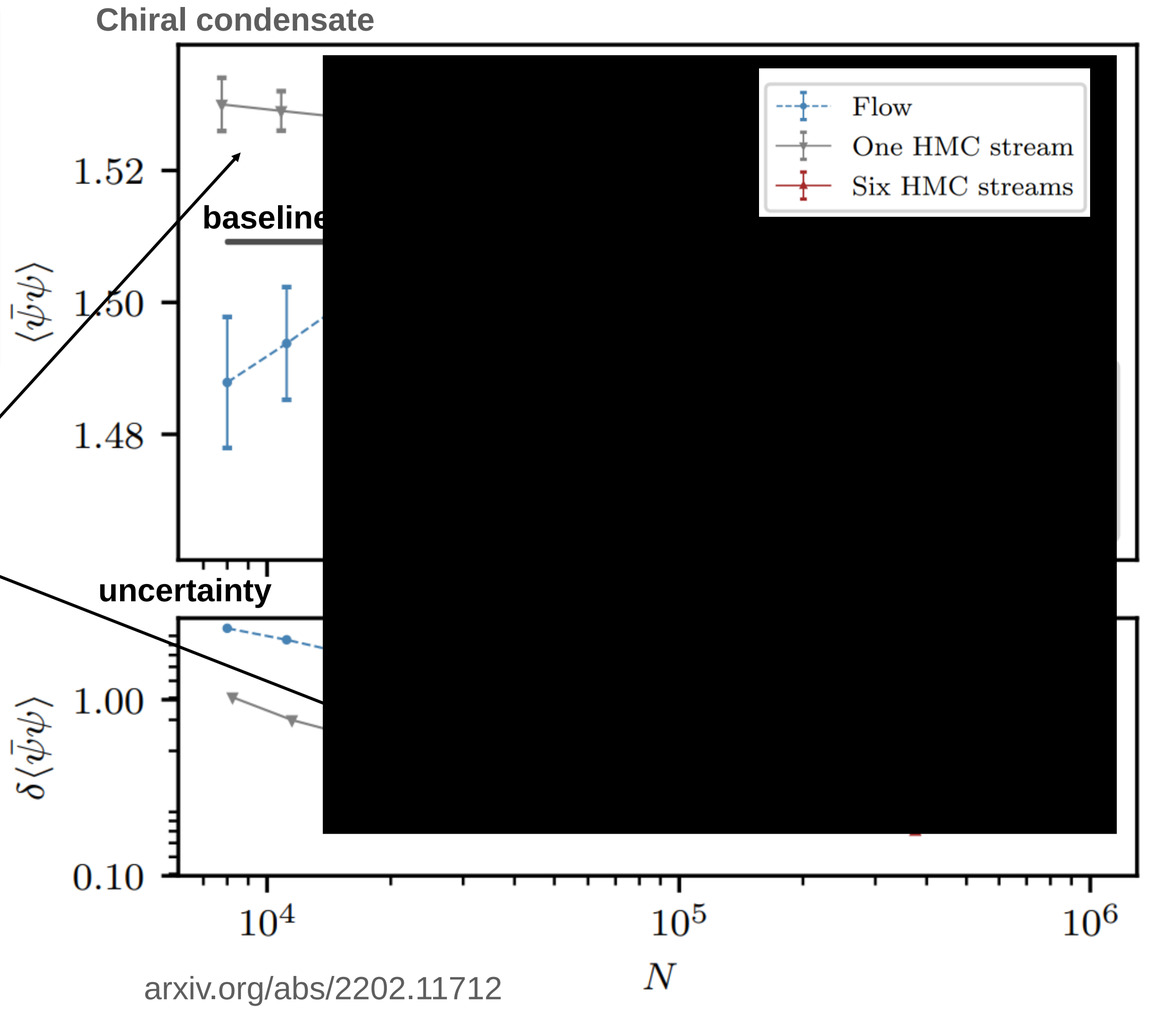
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

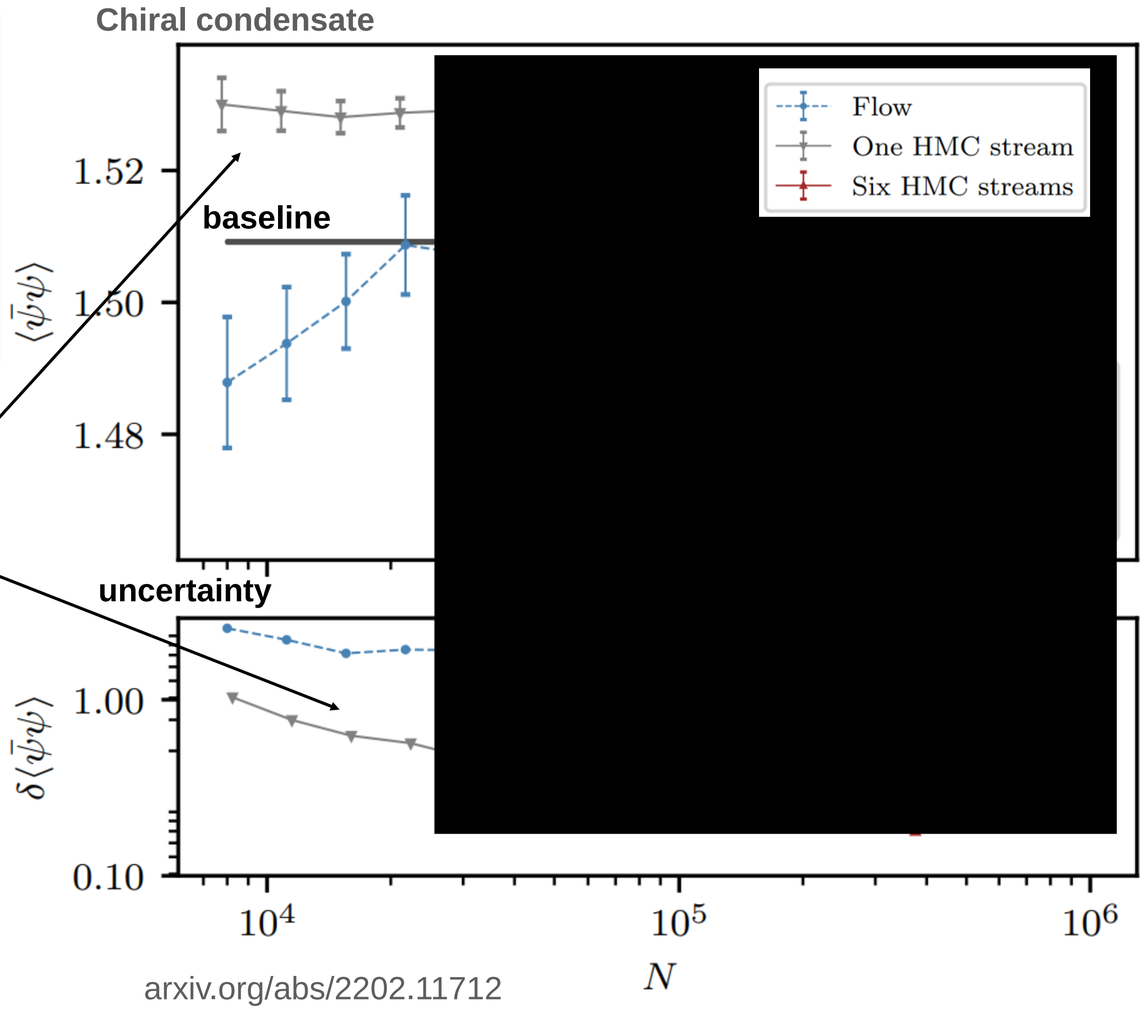
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

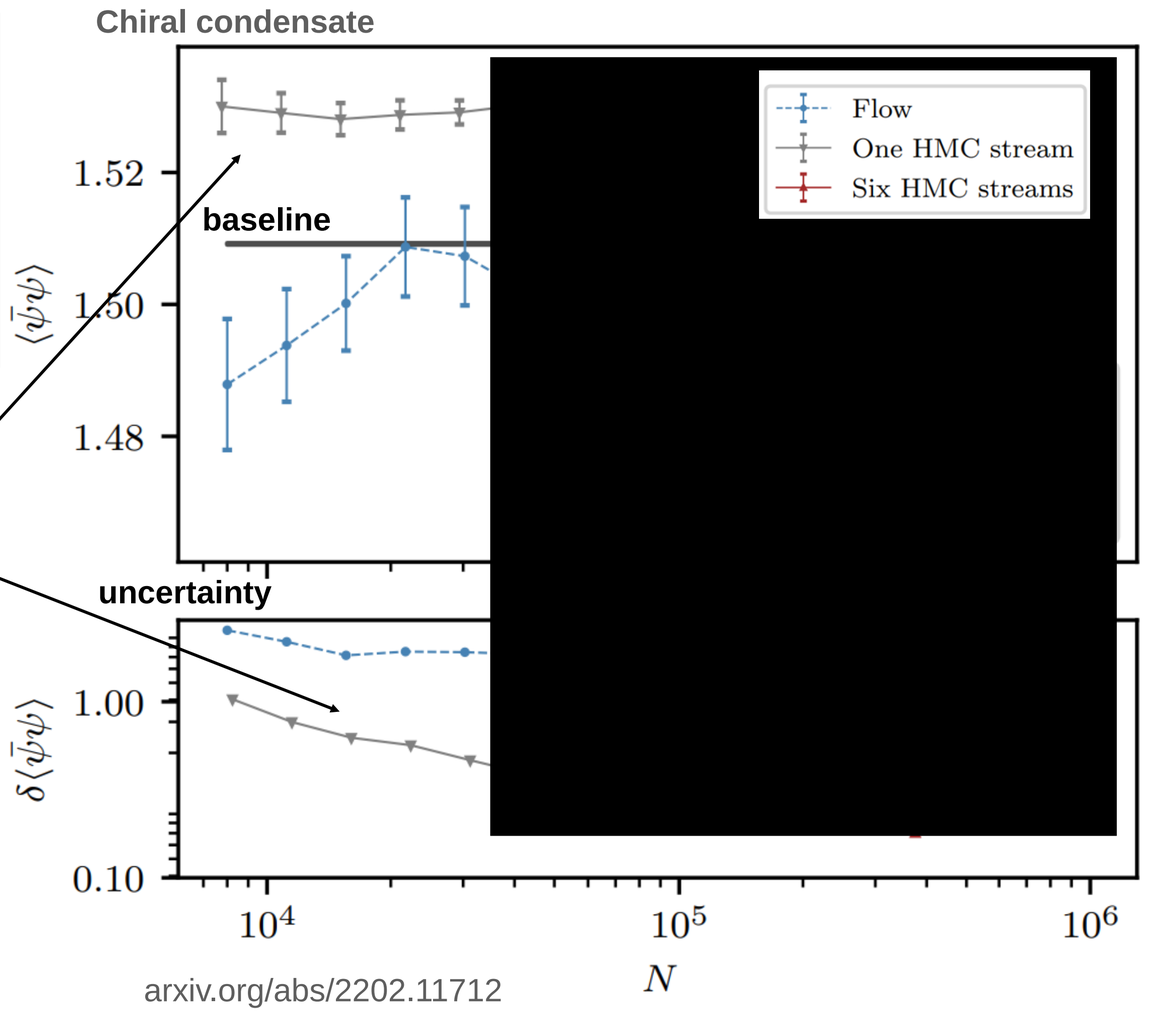
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

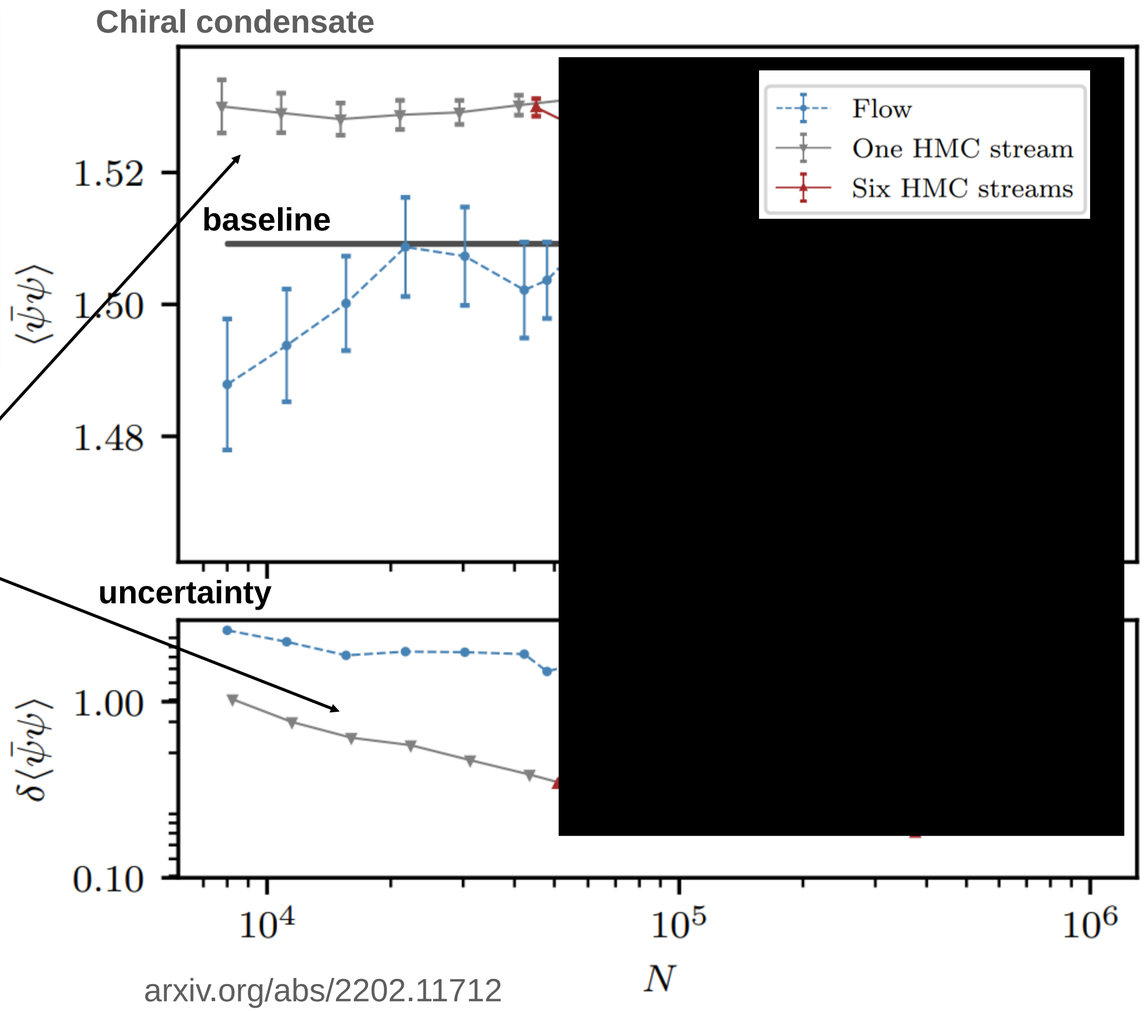
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

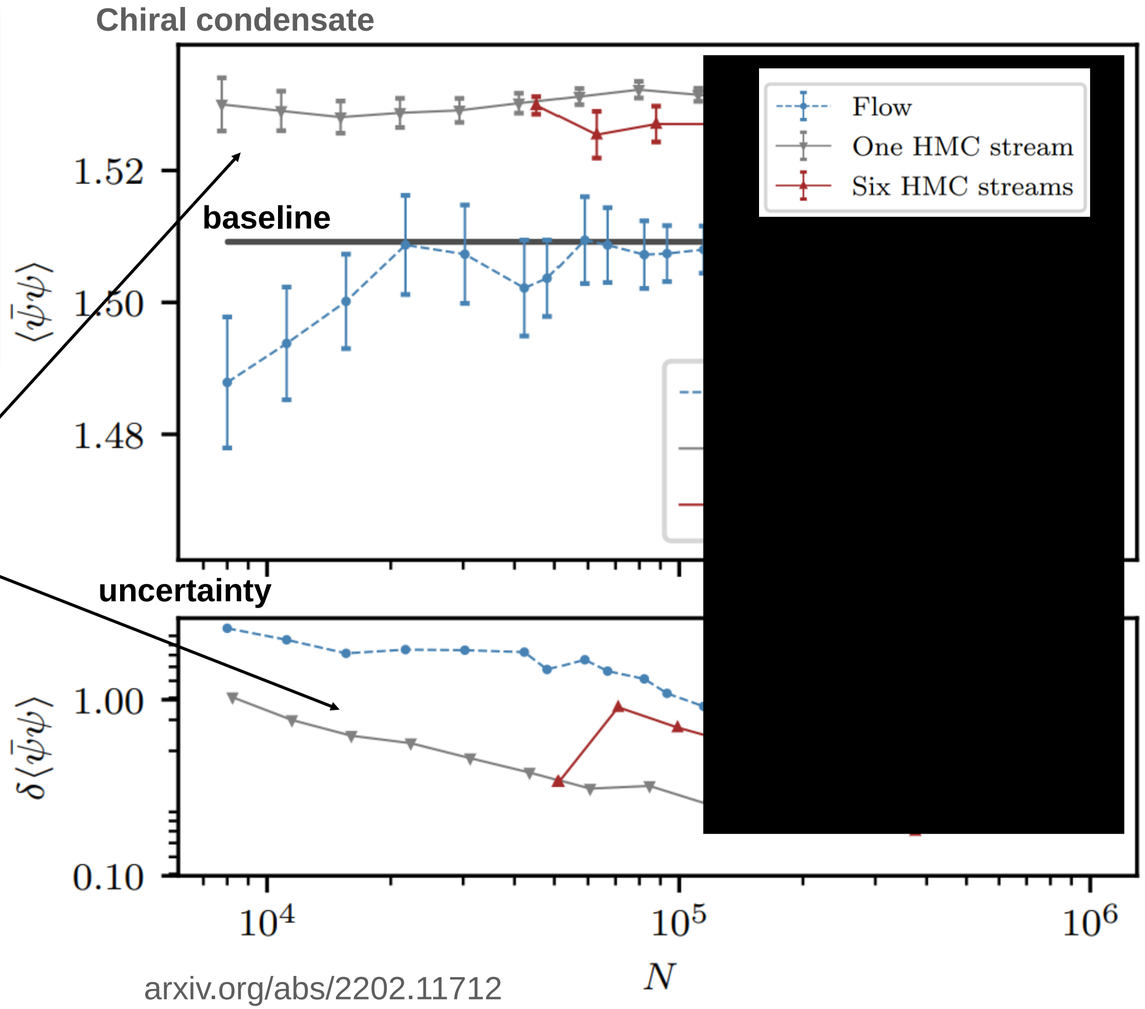
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

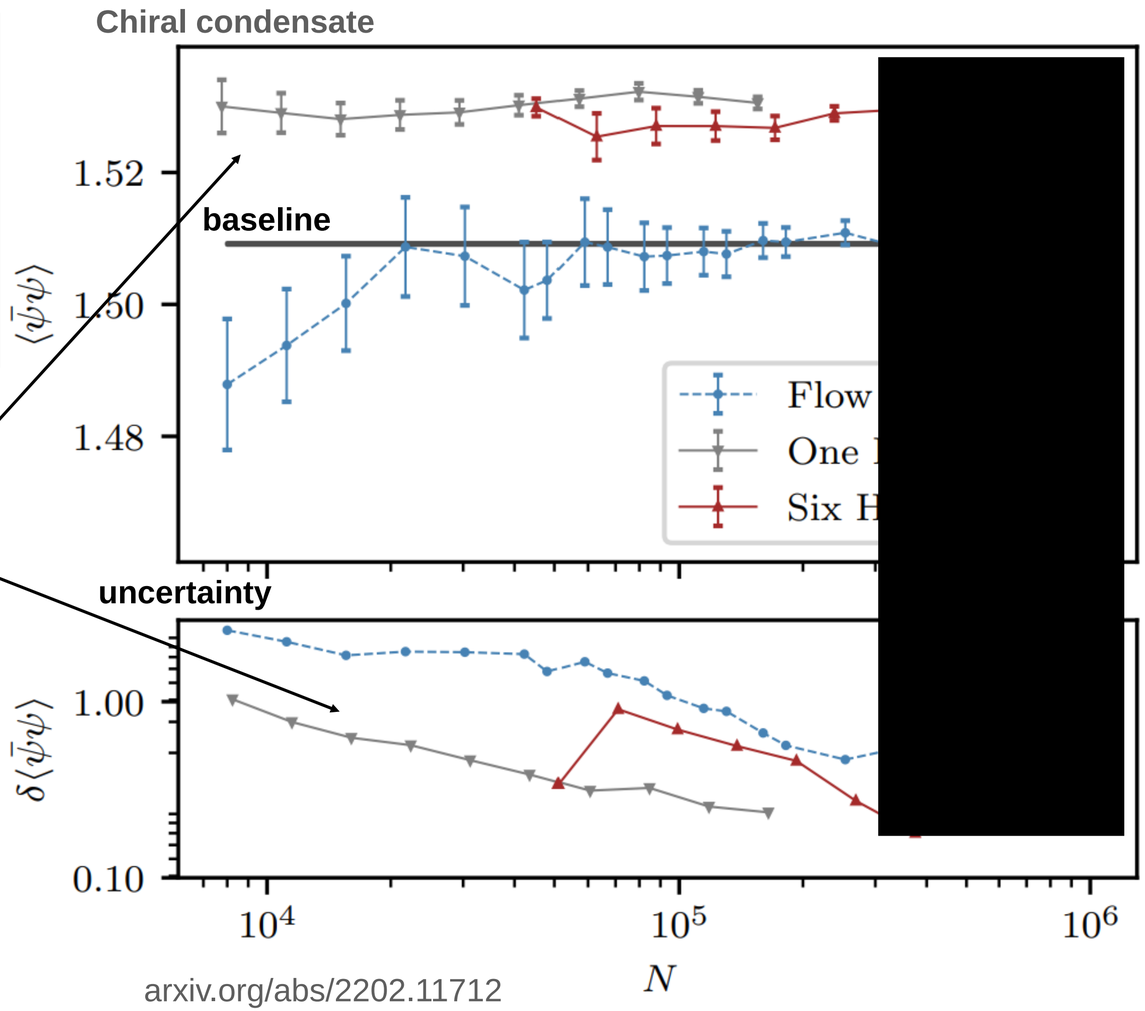
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Improving bias at small sample size

Long autocorrelations unseen until sample size >> autocorrelation time

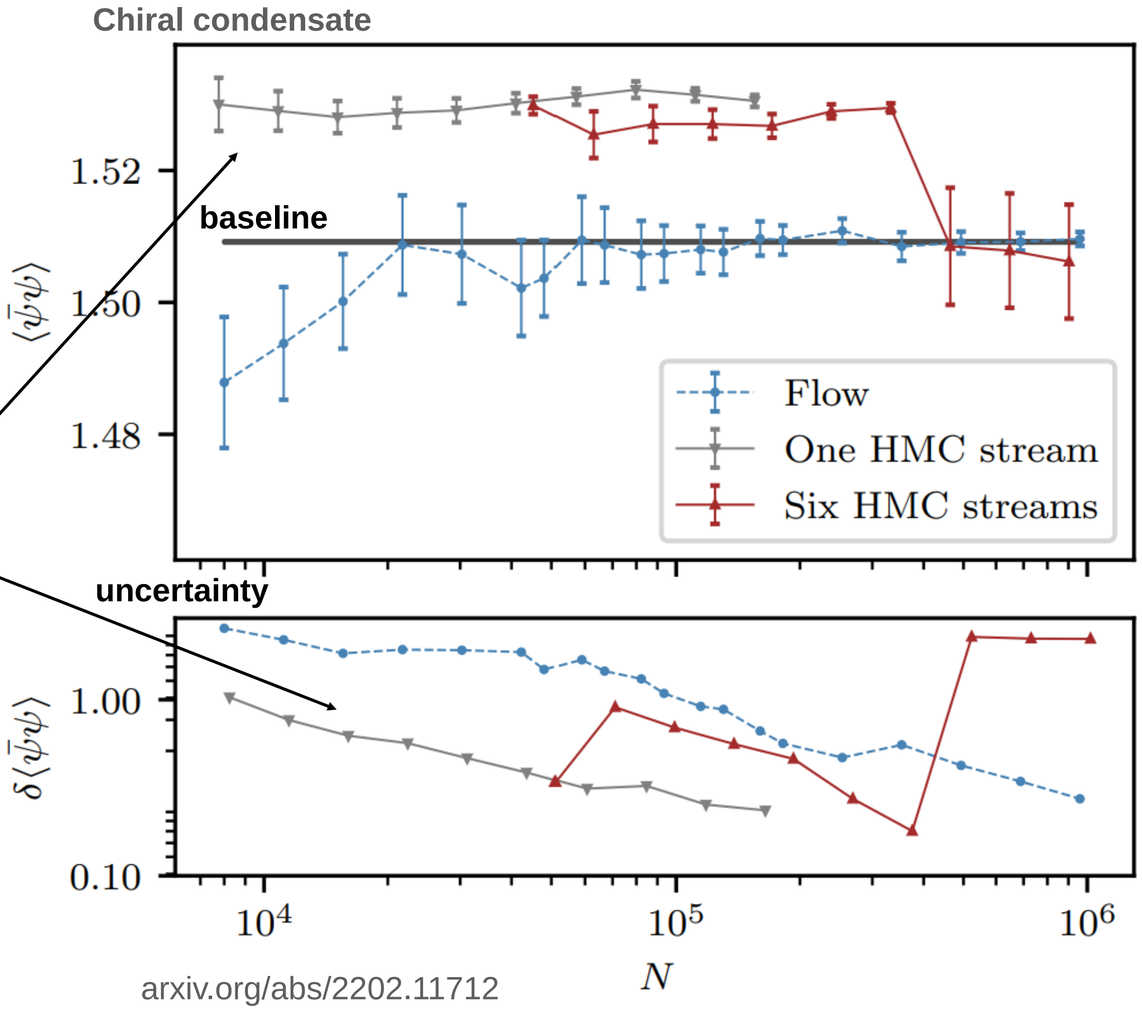
Simulations in a Schwinger model near criticality
- “Toy” QCD (confinement, topological freezing, UV fluctuations)
- HMC stands for Hybrid Monte Carlo algorithm which is state-of-the-art in LQCD
- Flow stands for NF model implemented in arXiv:2202.11712

Uncertainty scales correctly as $O(\frac{1}{\sqrt{N_{indep}}})$

but results are biased

Only at large sample size uncertainty jumps to correct values

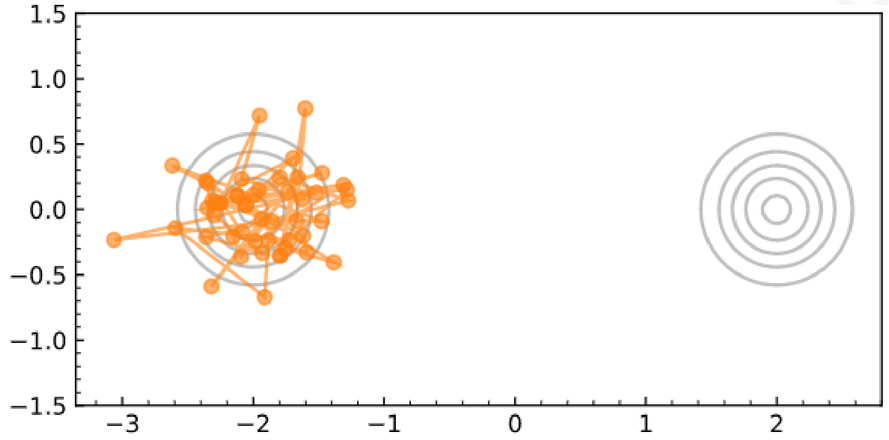
Normalizing Flow model (Flow in th figure) has correctly estimated uncertainty even at “small” sample size



arxiv.org/abs/2202.11712

(Toy) Prevent Mode Collapse

Example of mode collapse for two Gaussians – *MCMC algorithm generates sample only for one mode of target density*



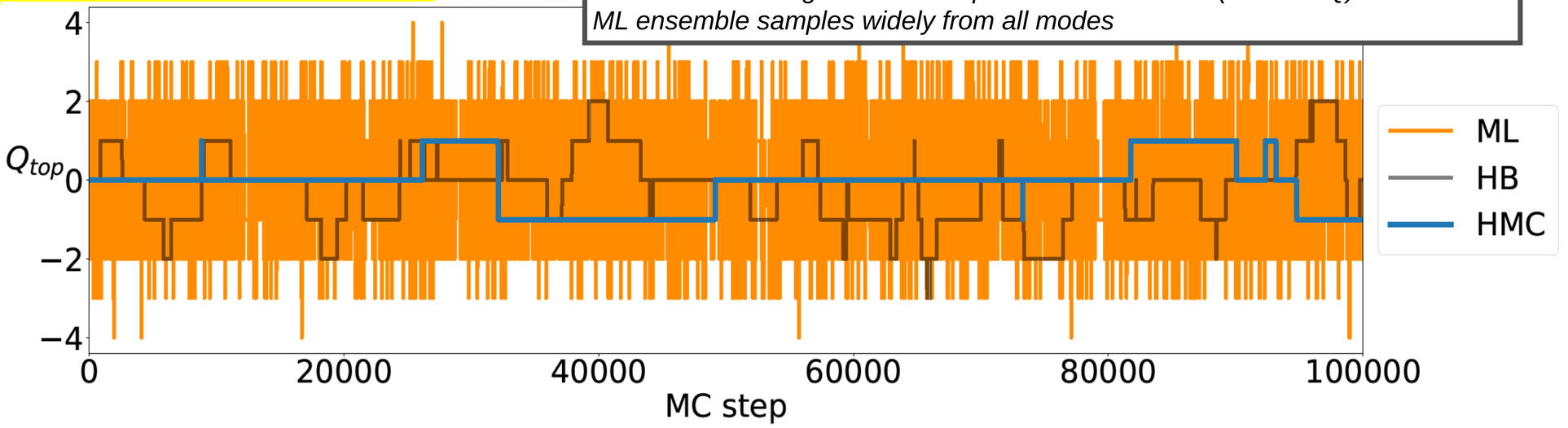
Example of sampling topological charge in U(1) lattice gauge theory (arXiv:2003.06413)

MCMC ensembles of configurations sampled with different algorithms

- HMC – Hybrid Monte Carlo, SOTA in LQCD
- HB – Heat Bath, SOTA on gluodynamics
- ML – Normalizing Flow model

HMC/HB have long chain of samples from one model (value of Q)
ML ensemble samples widely from all modes

Think of it as of index of the topological mode



(Toy) QFT Thermodynamics

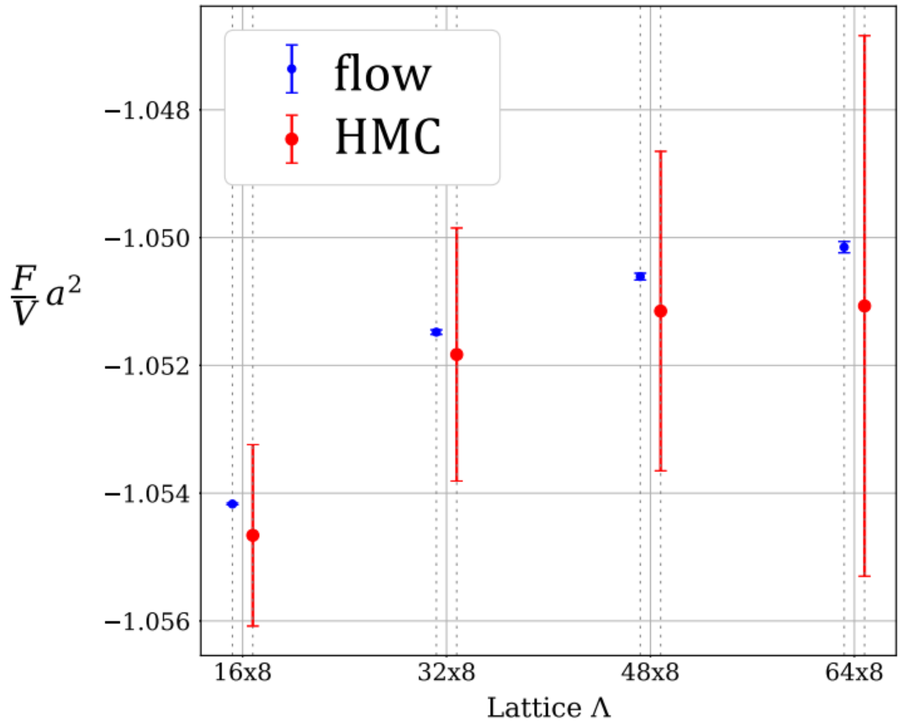
Lattice scalar field theory

The fundamental difficulty is that MCMC is not able to directly estimate the partition function of the lattice field theory.

Normalizing flows have direct access to partition function

$$Z = \int D\phi q_\theta(\phi) \frac{e^{-S(\phi)}}{q_{\theta(\phi)}} = \left\langle \frac{e^{-S(\phi)}}{q_{\theta(\phi)}} \right\rangle_{q_{\theta(\phi)}}$$

Example free energy computation



K. A. Nicoli, C. J. Anders, L.Funcke, T. Hartung, K. Jansen, P. Kessel, S. Nakajima, P. Stornati, Phys. Rev. Lett. 126, 032001 (2021)

(Toy) QFT phase diagram in (mu, T)

Direct MCMC simulations of QCD at nonzero chemical potential is not tractable due to Sign Problem

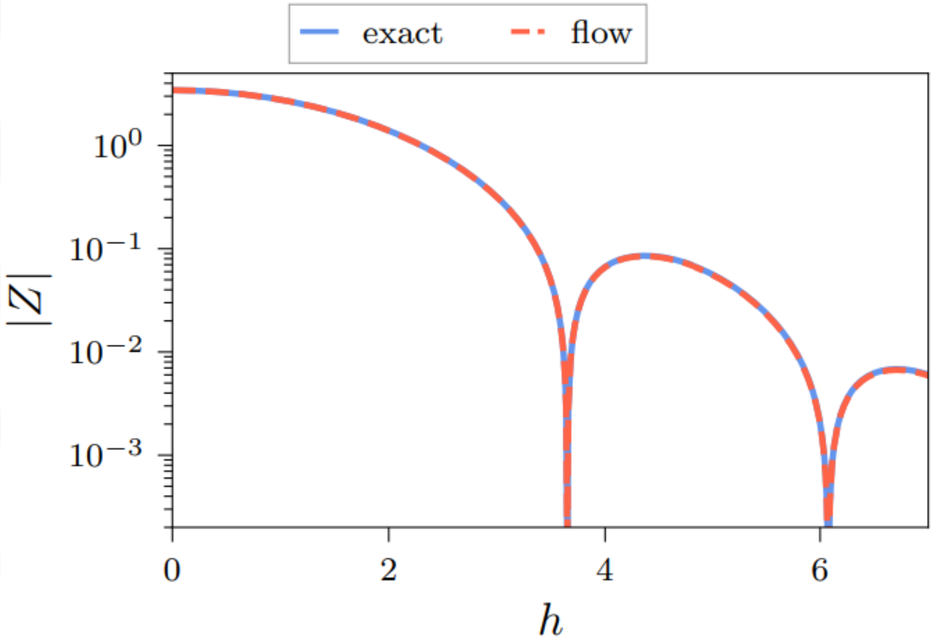
Several approaches use MCMC simulations at zero and/or imaginary chemical potential

Simulations at several values of imaginary chemical potential required in order to do extrapolation to real region

<u>After training Normalizing flow model gives access to “all” values of imaginary chemical potential</u>

Demonstration of flow-based Density of State

$$S(\phi) = \frac{m^2}{2}\left(\phi_1^2 + \phi_2^2\right) + \lambda\left(\phi_1^2 + \phi_2^2\right)^2 + ih\phi$$



Jan M. Pawlowski1 and Julian M. Urban,
https://arxiv.org/pdf/2203.01243.pdf

Examples in Quenched QCD

App 1: Pion $\langle x \rangle_g$ w/ flowed Feynman-Hellmann

[QCDSF-UKQCD 1205.6410]

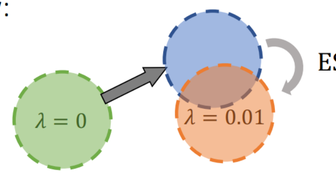
$$\delta S = -\lambda \frac{\beta}{N_c}\left[\sum_i P_{ti} - \sum_{i<j} P_{ij}\right]$$

$$\langle x \rangle_g^{\text{lat}} = -\frac{2}{3m}\frac{\partial m}{\partial \lambda}\bigg|_{\lambda=0} \approx -\frac{2}{3m}\frac{1}{\lambda}[m(\lambda) - m(0)]$$

Parameters:

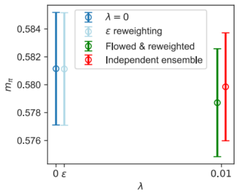
$8^3 \times 16 \quad \beta = 6 \quad \kappa = 0.132$ (quenched)

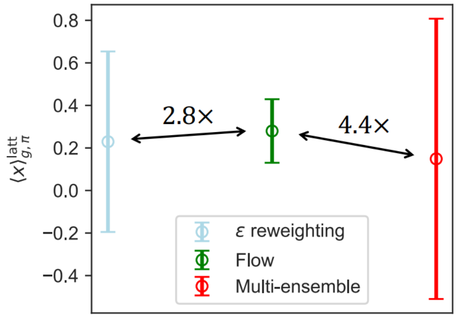
Flow:



ESS ≈ 0.84

Compute $m(\lambda)$ from $\langle w\, C^{2\text{pt}} \rangle_q$



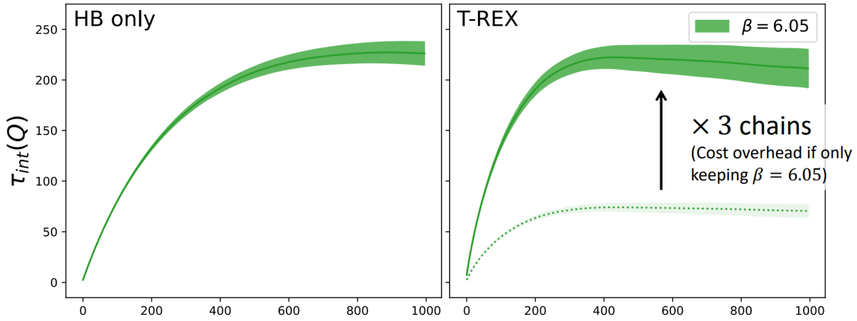


Dan Hackett - Aug 3 - Lattice 2023 8

Slide credit: Dan Hackett
See his talk at Lattice23

App 2: T-REX Results

Speed-up for multi-ensemble calculation
T-REX streams correlated (useful!)
Break-even for sampling $\beta = 6.05$
} Neglecting flow costs!



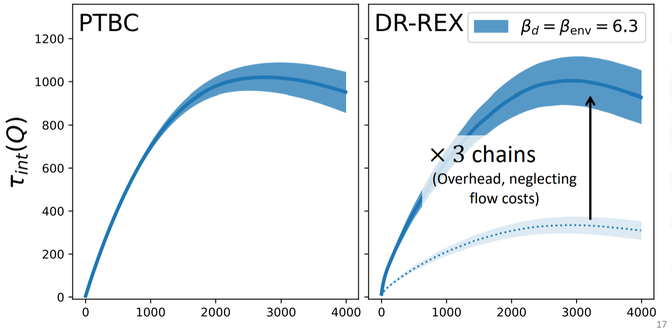
Dan Hackett - Aug 3 - Lattice 2023 12

App 3: DR-REX Results

Target: $\beta = 6.3$ on 16^4

Two flows to repair a 2^3 OBC defect $\beta_d = 0 \to 3 \to 6.3$

Flows act on 8^4 subvolume



Dan Hackett - Aug 3 - Lattice 2023 17

We need better models!

Normalizing flows

Flow-based models learn a change-of-variables that transforms a known distribution to the desired one
[Rezende & Mohamed 1505.05880]

<u>Idea:</u> change of variables

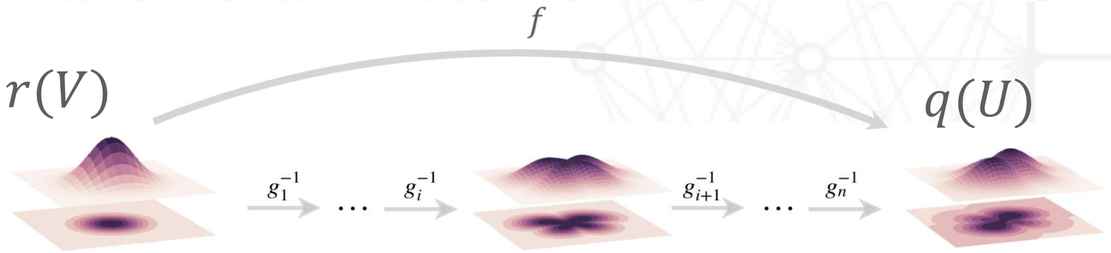


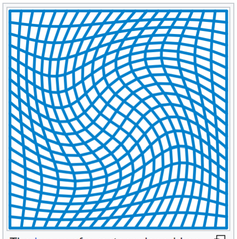
Image credit arXiv:1904.12072

$$\int r(V)dV \overset{U=f(V)}{=} \underbrace{\int r(U)\left|\det \frac{\partial [f(V)]^{-1}}{\partial U}\right| dU}_{=q(U)}$$

Posterior density $q(U)$ is **well-defined** only when flow transformation is diffeomorphism!

In mathematics, a **diffeomorphism** is an isomorphism of smooth manifolds. It is an invertible function that maps one differentiable manifold to another such that both the function and its inverse are differentiable.

WIKIPEDIA
The Free Encyclopedia



The image of a rectangular grid on a square under a diffeomorphism from the square onto itself.

Image credit to Wikipedia

Normalizing flows

Flow-based models learn a change-of-variables that transforms a known distribution to the desired one
[Rezende & Mohamed 1505.05880]

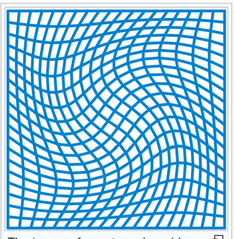
Idea: change of variables

Essentially, we need to build expressive diffeomorphism and train it as NF

Posterior density $q(U)$ is **well-defined** only when flow transformation is diffeomorphism!

In mathematics, a **diffeomorphism** is an isomorphism of smooth manifolds. It is an invertible function that maps one differentiable manifold to another such that both the function and its inverse are differentiable.

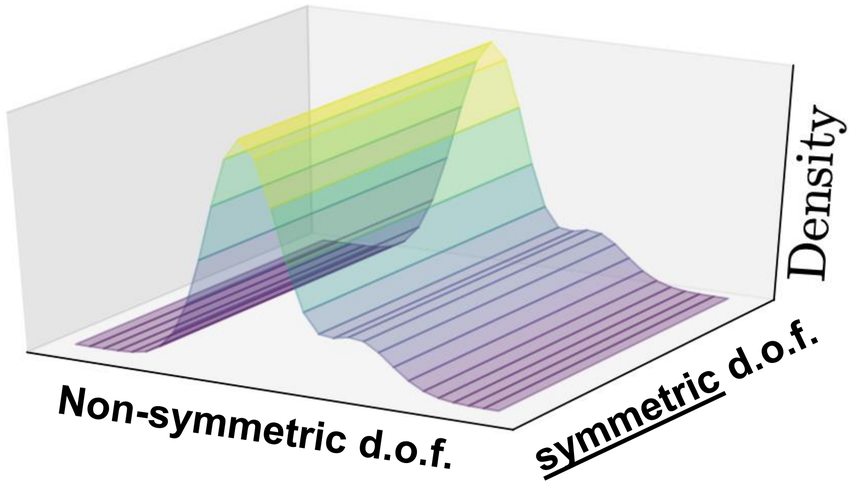
WIKIPEDIA
The Free Encyclopedia



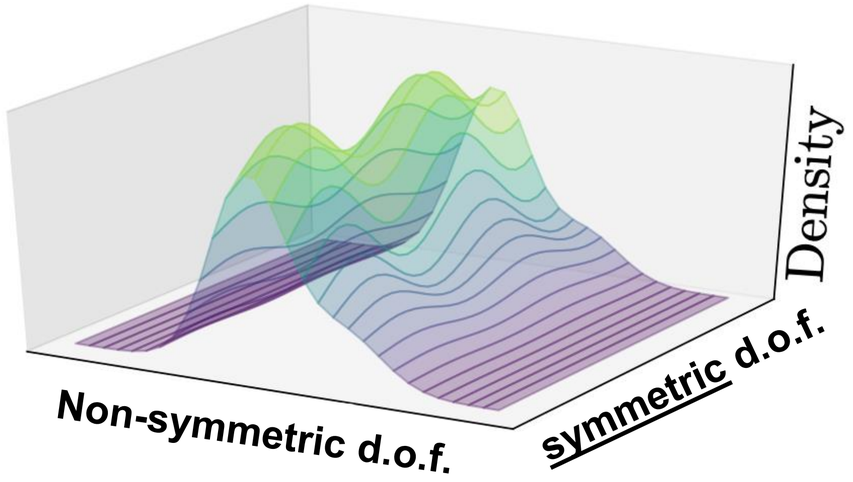
The image of a rectangular grid on a square under a diffeomorphism from the square onto itself.

Image credit to Wikipedia

Incorporating symmetries



True symmetric distribution



approximately learned

Normalizing flows produce invariant posterior distribution if

- Prior distribution is invariant

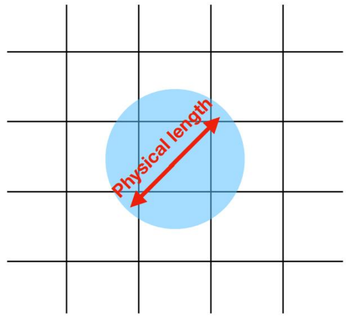
$$r(U) = r(\Omega \circ U)$$

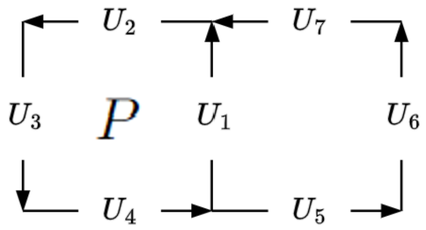
- Flow transformation is equivariant

$$f(U) \rightarrow f(\Omega \circ U) = \Omega \circ f(U)$$

For more information see works of Taco S. Cohen and Max Welling
And
Geometric Deep Learning

Lattice QCD intro





$$Z = \int \mathrm{d}U_{1,2,3,4,5,6,7} \exp\left(\frac{\beta}{3}\mathrm{Re}\,\mathrm{Tr}\left[U_1U_2U_3U_4 + U_5U_6U_7U_1^\dagger\right]\right)$$

Plaquette

$$P = U_1 \cdot U_2U_3U_4\,,$$

$$Z = \int \mathrm{d}P\mathrm{d}U_{2,3,4,5,6,7} \exp\left(\frac{\beta}{3}\mathrm{Re}\,\mathrm{Tr}\left[P + U_5U_6U_7 \cdot U_2U_3U_4P^\dagger\right]\right)$$

We build diffeomorphism in terms of plaquette(s) P but not a links U_i because gauge transformation of links is not local. Gauge transformation of plaquette is conjugacy.

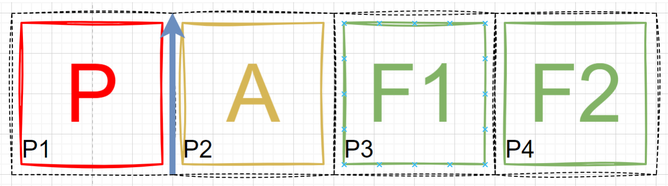
$$U_i \rightarrow \Omega_{a(i)} U_i\, \Omega^\dagger_{b(i)}$$

$$P \rightarrow \Omega P \Omega^\dagger$$

Spectral Flow intro

NF: $\int dV\, e^{S(V)} = \int dU\, e^{S(U)+logJ} = \int dU$

Target action $S = ReTr\, P_1 + ReTr\, P_2 + ReTr\, P_3 + ReTr\, P_4$



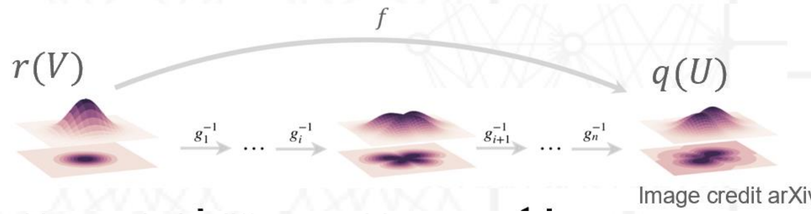


Image credit arXiv:1904.12072

This is just a layer in a sequence making a whole diffeomorphism

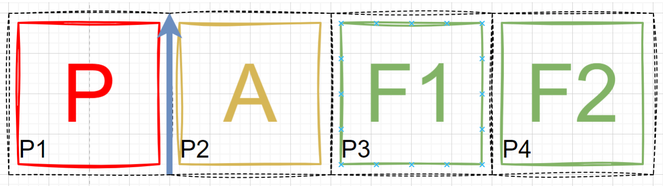
1) Mask plaquettes
2) Transform blue (active) link $U \rightarrow U'$
   a) Compute plaquette A

   b) Push update to the link $U \rightarrow U' = A' A^{\dagger} U$

Spectral Flow intro

NF: $\int dV\, e^{S(V)} = \int dU\, e^{S(U)+logJ} = \int dU$

Target action $S = ReTr\, P_1 + ReTr\, P_2 + ReTr\, P_3 + ReTr\, P_4$



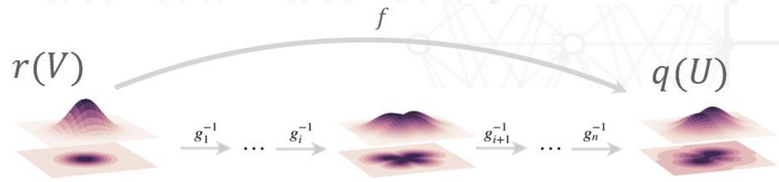


Image credit arXiv:1904.12072

This is just a layer in a sequence mak

1) Mask plaquettes
2) Transform blue (active) link $U \rightarrow U'$
   a) Compute plaquette A
   b) Transform active plaquette $A \rightarrow A' = f(A|\, TrF1,\, TrF2)$
      - Diagonalize $A = V^\dagger\, L\, V$
      - Transform diagonals $L \rightarrow L' = g(L|TrF1, TrF2)$
      - Undiagonalize $A = V^\dagger\, L'\, V$
   b) Push update to the link $U \rightarrow U' = A' A^\dagger U$

How to build conjugacy-equivariant flow

$$f(\Omega\, U\, \Omega) = \Omega\, f(U) \Omega^\dagger$$

- Use function on a matrix:

$$f(U) = f\left(V\, D\, V^\dagger\right) = V f(D) V^\dagger = V \begin{pmatrix} f(d_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & f(d_n) \end{pmatrix} V^\dagger$$

- Under symmetry transformation:

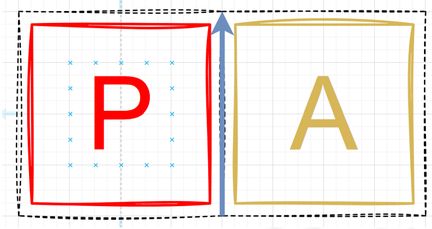
$U \rightarrow \Omega U \Omega^\dagger$
$V \rightarrow \Omega V, V^\dagger \rightarrow \Omega^\dagger V^\dagger, D \rightarrow D$
$f(U) \rightarrow f\left(\Omega\, U\, \Omega^\dagger\right) = \Omega V f(D) V^\dagger \Omega^\dagger = \Omega\, f(U) \Omega^\dagger$

Conjugacy-equivariant transformation is transformation on a Maximal Torus!

Improving Expressivity

Target action

$S = Re\ Tr\ (A + P)$



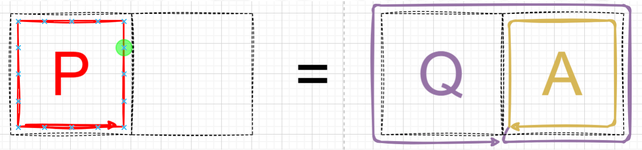
What would happen if we used different active loops (W2x1, W2x2,…)?
It would only change loop(s) Q!

$U' = f(U)$

We need take into dependence of passive loop on active. Both are transformed though an active link, let's use it.
$A = S_A U^\dagger$, $P = U\ S_P$ => $P = A^\dagger S_A S_P$

Aim: $\int dU' \exp[S(U) + logJ] = \int dU$

$\Rightarrow -logJ = S(U) = ReTr(A + P) = ReTr(A\ (1 + Q))$



Recall for spectral flow

a) Transform active plaquette $A \rightarrow A' = f(A|\ TrF1,\ TrF2)$
- Diagonalize $A = V^\dagger L\ V$
- Transform diagonals $L \rightarrow L' = g(L|TrF1, TrF2)$
- Undiagonalize $A = V^\dagger L'\ V$

b) Push update to the link $U \rightarrow U' = A' A^\dagger U$

And don't forget Haar measure

$\Rightarrow \frac{dL'}{dL} = Haar(L)\ \exp\left[-Re\ Tr\ \left(L\ V(1 + Q)V^\dagger\right)\right]$

L is a diagonal matrix of eigen values

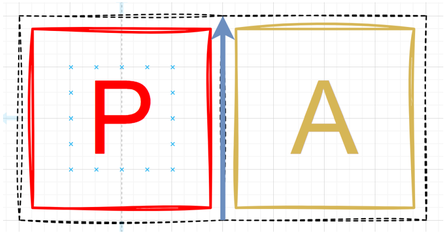
This transformation trivialized target action!

$\Rightarrow L' = \int\ \exp\left[-Re\ Tr\ \left(L\ (V(1 + Q)V^\dagger\right)\right] Haar(L) dL$

If we could take this integral, we would not need NN for parametrization of diffeomorphism

Improving Expressivity 2

Target action $S = Re\ Tr\ (A + P)$



We can add $diag$ because L is diagonal!

$$L' = \int \exp\left[-Re\ Tr\left(L\ V(1+Q)V^{\dagger}\right)\right] Haar(L)dL$$

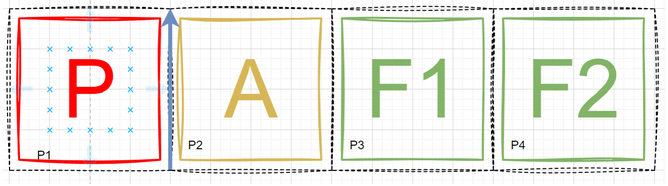
L is a diagonal matrix of eigen values

Expressive transformation $L' = f(L|features)$ must be conditioned on features

$$\boldsymbol{diag(V(1+Q)V^{\dagger})}$$

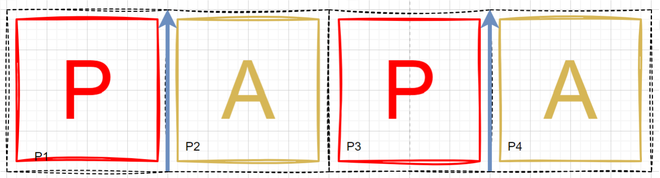
which are gauge-invariant and eigen decomposition invariant.

How would transformation change if we had frozen loops $S = Re\ Tr\ (A + P + F1 + F2)$?



- With one iteration we can trivialize only plaquettes (active and passive) which contain active link
- In this case there is no useful information in frozen plaquettes

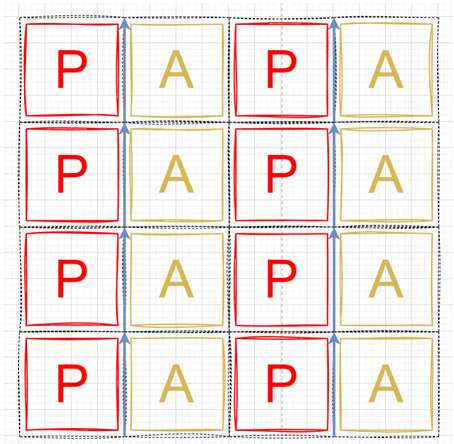
=> With dense mask we can(speaking only about 2D) trivialize all plaquettes in the action



Expressive Spectral flow

Algorithm

1) Use dense mask as frozen loops contain no useful information



2) Build proper features

$$diag(V(1+Q)V^{\dagger})$$

3) Use expressive transformation (splines)

4) Not all links need to be transformed

Spectral flow Coupling Layer:

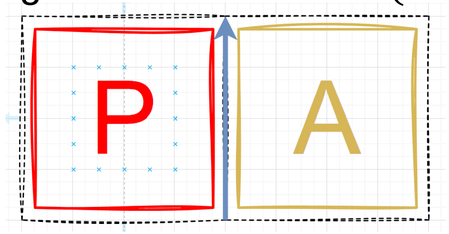
- Apply mask
- Transform active links $U \rightarrow U'$
- Transform active plaquette $A \rightarrow A' = f(A|features)$
  - Diagonalize $A = V^{\dagger} L\, V$
  - Compute loops Q *(more on this later)*
  - Build “diagonal” features $diag(V(1+Q)V^{\dagger})$
  - Transform $L \rightarrow L' = g(L|features)$
  - Undiagonilize $A = V^{\dagger} L'\, V$
- Push update to the link $U \rightarrow U' = A' A^{\dagger} U$

With expressive transformation $g(L|features)$
logJ should trivialize/compensate plaquettes in the action.

Only one coupling layer is needed!

A Deeper Look

Target action $S = Re\ Tr\ (A + P)$



$$L' = \int \exp\left[-Re\ Tr\left(L\ (V(1+Q)V^{\dagger}\right)\right] Haar(L)\ dL\ * Constant$$

In fact, there is a family of solutions. Which should we choose?

Remember Coupling transformation $L' = f(L|features)$ must be <u>diffeomorphism on circle</u>!

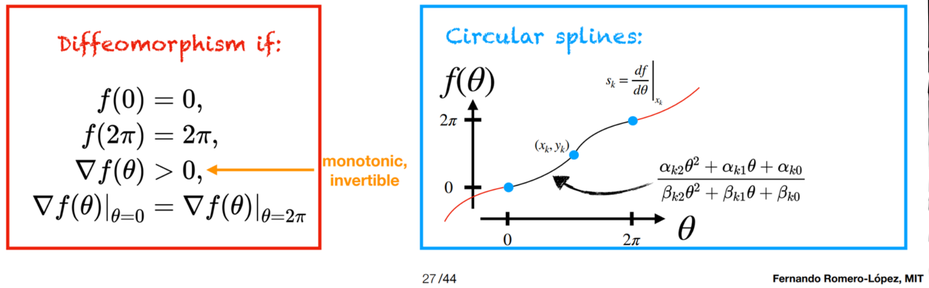


Image credit: F. Romero-López

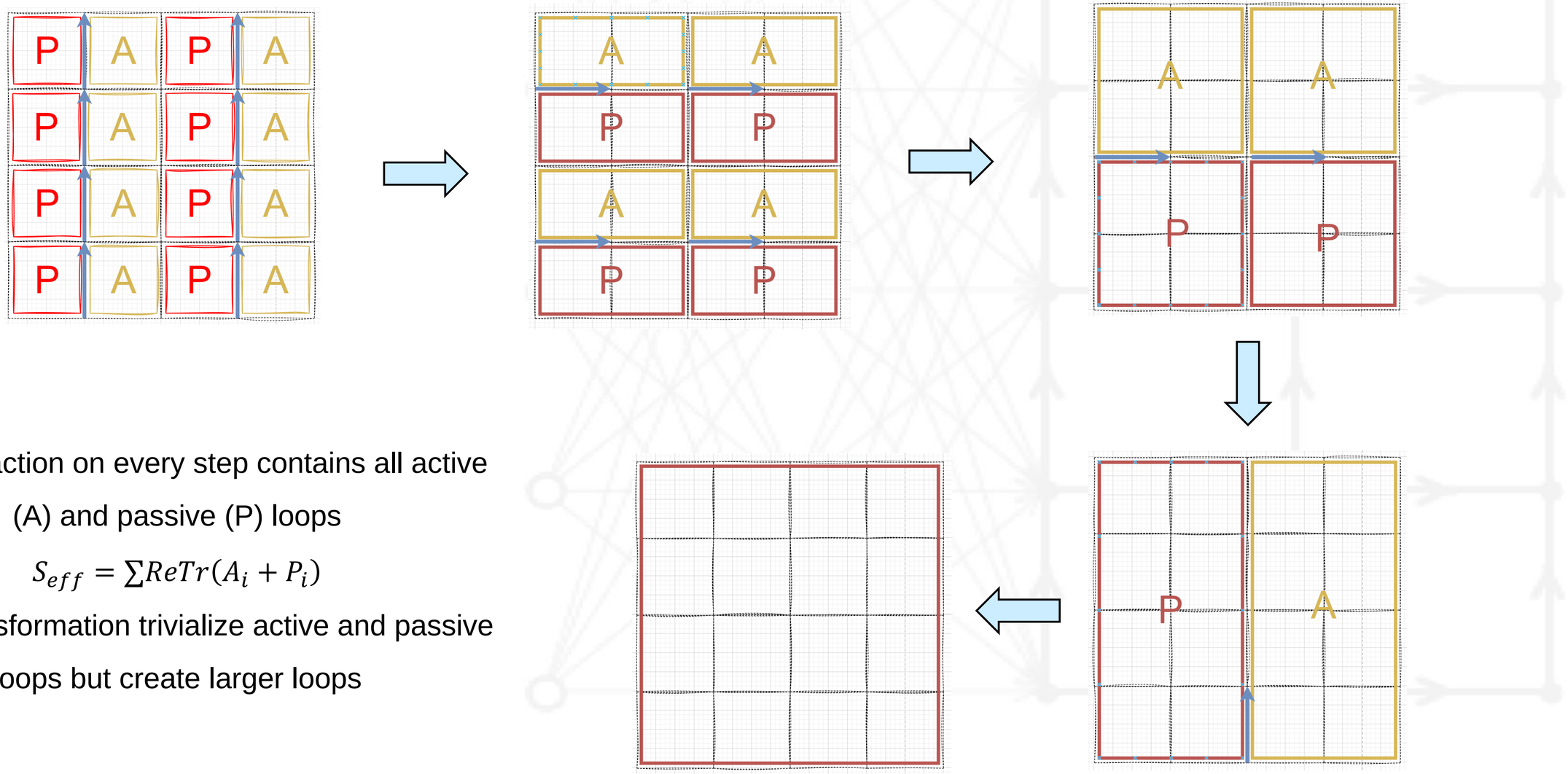
This uniquely determined a solution from this family! Coupling transformation should satisfy:

- $L' = e^{\Theta'=0} = f(L = e^{\Theta=0}|features)$ => $f(I|features) = I$

$$L' = \frac{1}{Normalizer(Q)} \int \exp\left[-Re\ Tr\left(L\ (V(1+Q)V^{\dagger}\right)\right] Haar(L) dL$$

This normalizer will appear in logJ!
As a result after trivializing plaquettes we will have larger loops (Q) in the effective action

Example trivializing 2D LGT



Effective action on every step contains all active (A) and passive (P) loops

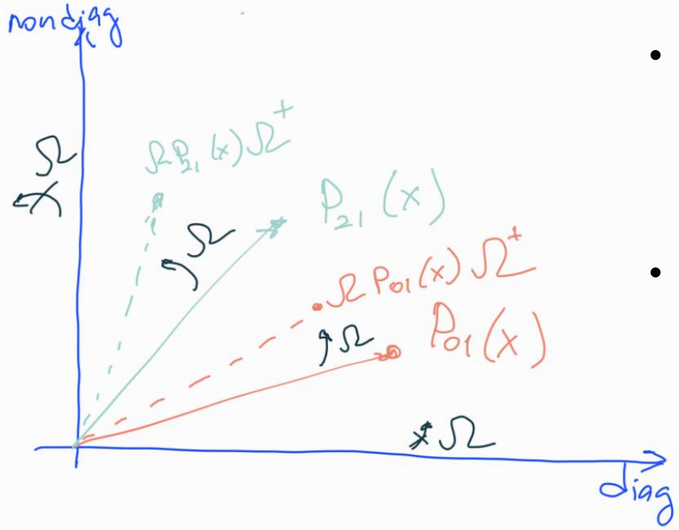
$$S_{eff} = \sum ReTr(A_i + P_i)$$

Every transformation trivialize active and passive loops but create larger loops

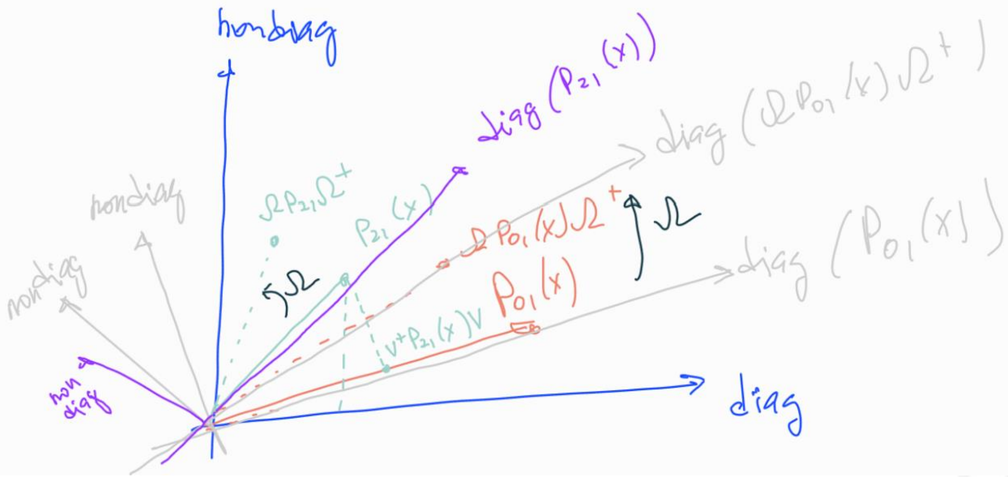
Interpretation

$$W = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \Rightarrow \big(nondiag(W), diag(W)\big)$$

$$nondiag(W) = (b, c, d, f, g, h), \qquad diag(W) = (a, e, i)$$



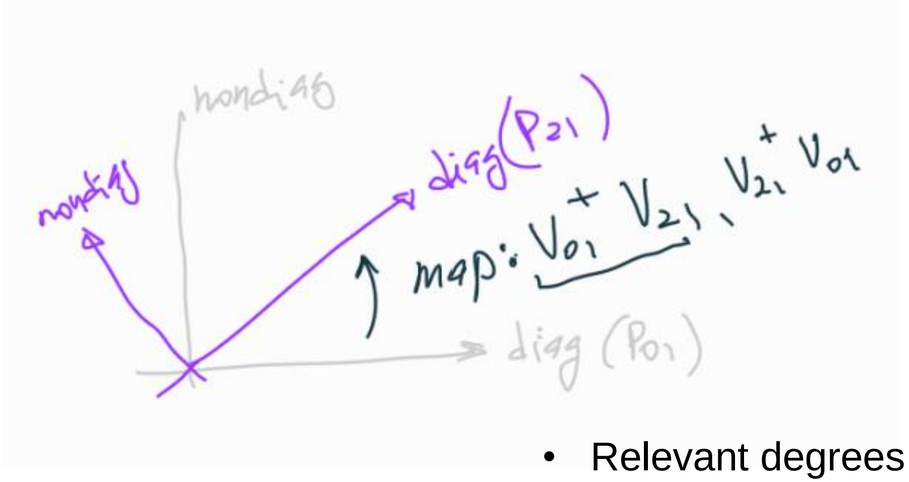
- In “static” (conventional) coordinate system every loop is rotated with a gauge
- Target action is “scalar” and does not depend on choice of coordinate system

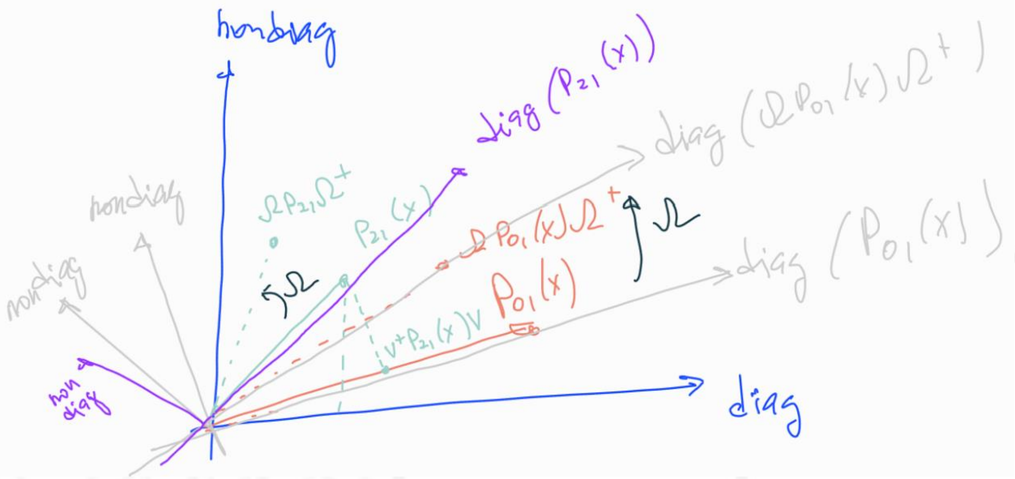


- There is a local coordinate system in which loop is diagonal
- These coordinate systems are not unique and transformed under symmetry (gauge) transformation
- They are transformed in a same way as loops in “static” coordinate system

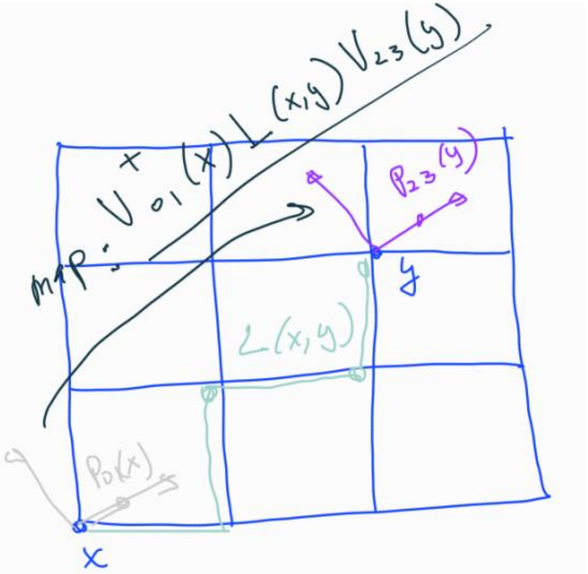
- Every loop is evaluated in local coordinate system (Gauge symmetry)
- Relevant degrees of freedom are transformed in a local coordinate system when loop is diagonal

Interpretation





- Relevant degrees of freedom are transformed in a local coordinate system when loop is diagonal
- Relevant features are loops evaluated **in the same coordinate system**
- Map between different coordinate systems is defined by eigen vectors V

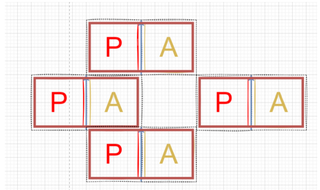


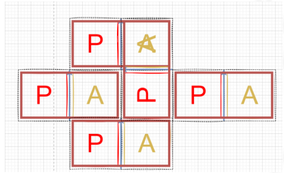
Example

- Active loop $P_{01}(x) = V_{01}(x)^{\dagger} D(x) V_{01}(x)$ is transformed in local C.S. - $D(x)' = f(D(x)|\,.)$
- Conditional information could be a loop $P_{23}(y) = V_{23}(y)^{\dagger} D(y) V_{23}(y)$
- Map is defined by <u>parallel transport</u> and <u>eigenvectors</u> $V_{01}(x)^{\dagger} L(x,y) V_{23}(y)^{\dagger}$
- Transformation $\mathrm{L}'(\mathrm{x}) = \mathrm{f}(\mathrm{L}(\mathrm{x})|\, V(x) U_{\mu} V(x+\mu), \mathrm{L}(\mathrm{x}+\mu))$

Uncovered but Important topics

- **Defects**. 2D LGT can be tilled and trivialized in hierarchical way. However, 3/4D LGT can not be tiled. In 3/4D there are limited number of links/loops which transformed more/less then others, I call them **defects**.





- In **3/4D** to deal with defects more expressive transformation must be used. It results in more complicated features and **autoregressive coupling**.
- We can think of a algorithm which automatically builds necessary featured loops from active links/loops
- In LGT **volume scaling** is determined by residual of interpolation. Keeping residual the same as volume growth will result in a same model quality. Naive analysis predicts increasing number of interpolation intervals as $\boldsymbol{V\sqrt{V}}$

Take away message

Blind application of ML techniques delivers not the best results. Approach should be adapted with math and knowledge of the physical domain

Thank you for attention!

Let’s build stronger our AI4Science community together