Accelerated Monte Carlo methods from machine learning

Yukari Yamauchi

- arXiv:2310.xxxxx with Landon Buskirk, Pablo Giuliani, Kyle Godbey
- arXiv:2310.xxxxx with Scott Lawrence

September 26th, 2023

Workshop: Probing the Frontiers of Nuclear Physics with AI at the EIC

INSTITUTE for NUCLEAR THEORY

“Accelerating” Monte Carlo sampling for EIC physics

1. Uncertainty quantification via Bayesian analysis

$$P(\omega|\boldsymbol{Y}) = \frac{P(\boldsymbol{Y}|\omega)P(\omega)}{P(\boldsymbol{Y})}$$

   Assisting continuous calibration of model parameters $\boldsymbol{\omega}$ via **normalizing flows**

2. Lattice calculations

$$\langle\mathcal{O}\rangle = \frac{\int \phi \exp(-S_0(\phi))\ \mathcal{O}(\phi)}{\int_\phi \exp(-S_0(\phi))}$$

   Increasing effective sampling size via **control variates**

Bayesian analysis with normalizing flows

Bayes theorem provides posterior distribution

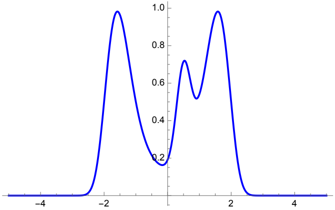
$$P(\omega|\boldsymbol{Y}) = \frac{P(\boldsymbol{Y}|\omega)P(\omega)}{P(\boldsymbol{Y})}$$

How do we sample from $P(\omega|\boldsymbol{Y})$? Normally MCMC

Normalizing flow $\omega = f(\boldsymbol{x})$ maps

$$d\omega\ P(\omega|\boldsymbol{Y}) = \mathcal{N} \prod_{i=1}^{N} dx_i\ e^{-x_i^2/2}$$

Some distribution

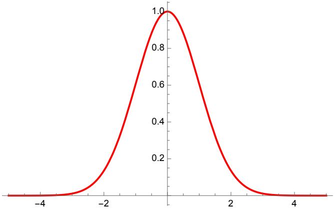


Map

$$\leftrightarrow$$

$$\omega = f(\boldsymbol{x})$$

Gaussian distribution



Mahcine-learned normalising flow

Find an approximate normalizing flow

$$\mathcal{N} \prod_{i=1}^{N} dx_i \; e^{-x_i^2/2} = d\omega \; Q(\omega) \approx d\omega \; P(\omega|\boldsymbol{Y})$$

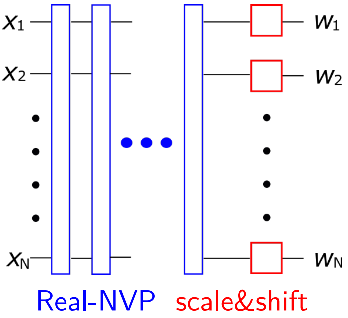
by optimizing parameters in the neural network

Real NVP (6 layers) + Scale&Shift

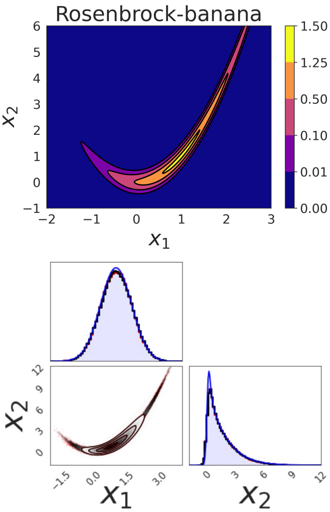
1. Prepare train data via MCMC
2. Initialize NN as a Gaussian fit
3. Train via Jeffreys' divergence

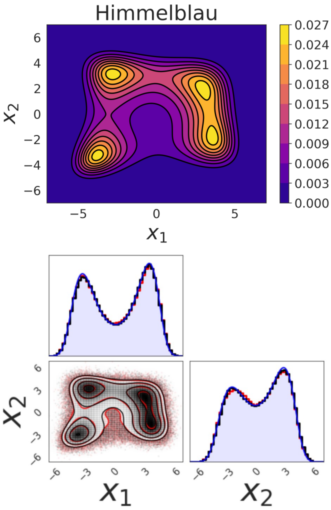
$$D(P|Q) = \int d\omega \left(\tilde{P} \log \frac{P}{Q} + \tilde{Q} \log \frac{Q}{P} \right)$$

- Adam with learning rate of 10^{-3}
- 1000 samples/train step



2-dimentional examples

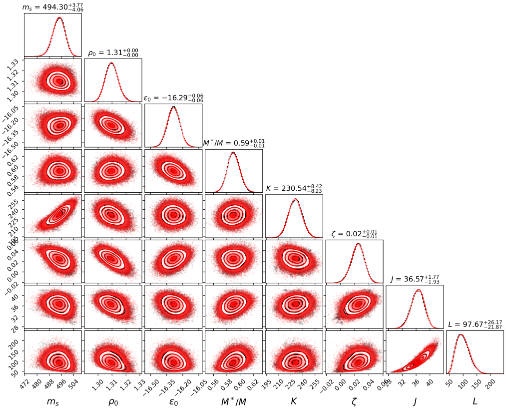




blue: exact, **black**: train data, **red**: from normalizing flow

Relativistic mean field model[3]

Calibrated by experimental binding energies and charge radii of nuclei[1]
(^{16}O, ^{40}Ca, ^{48}Ca, ^{68}Ni, ^{90}Zr, ^{100}Sn, ^{116}Sn, ^{132}Sn, ^{144}Sm, ^{208}Pb)



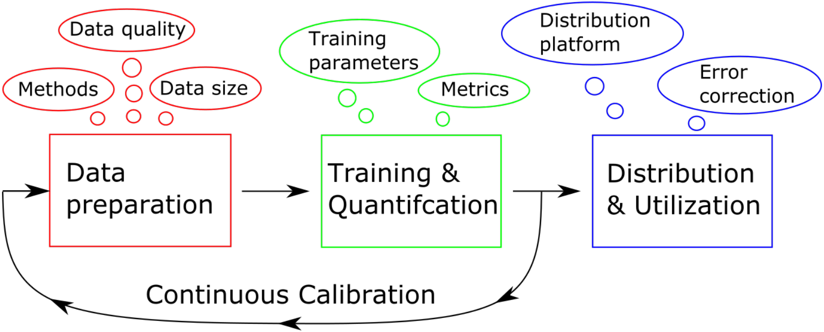
- m_s: σ meson mass
- ρ_0: saturation density
- ϵ_0: binding energy at ρ_0
- M^*: effective nucleon mass at ρ_0
- K: incompressibility at ρ_0
- J: value of symmetric energy at ρ_0
- L: slope of symemtric energy at ρ_0
- ζ: ω meson quartic coupling

Black: MCMC, Red: NF

All codes used in this work will become available at publication[2]

[1] P. Giuliani, K. Godbey, et al., arXiv:2209.13039[nucl-th]
[2] **YY**, L. Buskirk, P. Giuliani, and K. Godbey, in preparation
[3] W. Chen and J. Piekarewicz, arXiv:1408.4149

Why normalizing flow? Why with machine learning?



- Neural network can parametrize a large family of functions efficiently.
- Normalizing flows serve as a compression tool for $P(\boldsymbol{\omega}|\boldsymbol{Y})$.
- One can generate more samples easily, quickly, in parallel (including reweighting).

Next!

Improving the statistics in lattice calculations via **control variates**

For a given Monte Carlo sampling task in lattice simulations

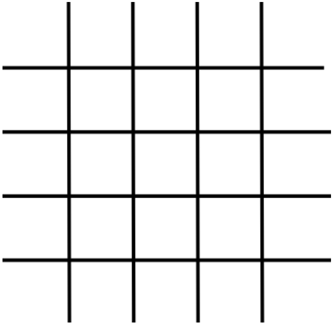
$$\langle \mathcal{O} \rangle = \frac{\int \mathcal{D}[\phi]\ e^{-S(\phi)}\ \mathcal{O}(\phi)}{\int \mathcal{D}[\phi]\ e^{-S(\phi)}}$$

with bad

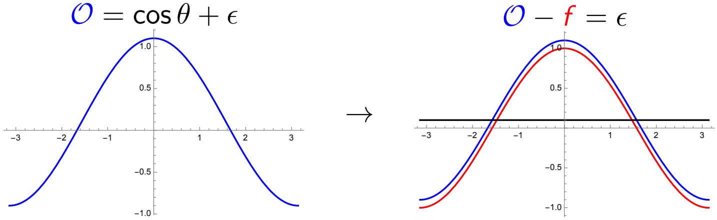
Signal-to-noise problem

How do we reduce the "noise"

Variance $\mathrm{Var}(\mathcal{O})$



Complex control variates



The idea is very simple...

Subtract a function f from $\mathcal{O}$!!

Without changing physics, so

$$\langle \mathcal{O} \rangle = \langle \mathcal{O} - f \rangle \;\text{ but }\; \mathrm{Var}(\mathcal{O}) > \mathrm{Var}(\mathcal{O} - f)$$

So we strictly impose

$$\langle f \rangle = \int \mathcal{D}[\phi]\, e^{-S(\phi)}\, f(\phi) = 0$$

Notes on control variates

Strength of control variates

- Can be applied to any (including discrete) lattice theories
- Good (or even perfect) control variates always exist

How do we find good control variates?

1. Analytical (perturbative) approaches
   - S. Lawrence, arXiv:2009.10901[hep-lat]
   - S. Lawrence and **YY**, arXiv:2212.14606 [hep-lat]

2. Numerical approaches
   - Start with ansatz and optimize
   - Machine learning

Demonstration: Classical Ising model (Lee-Yang zeros)

Classical Ising model: $S(\vec{s}) = -J\sum_{\langle i,j\rangle} s_i s_j - h\sum_i s_i$

Lee-Yang Theorem: the partition function is **0** only on with imaginary h.

Goal: Compute $Z = \sum_s e^{-S}$ at **purely imaginary magnetic field**

Measure

$$\frac{Z(h)}{Z(h=0)} = \frac{\sum_s \exp\Big(J\sum_{\langle i,j\rangle} s_i s_j\Big)\ \exp\big(h\sum_i s_i\big)}{\sum_s \exp\Big(J\sum_{\langle i,j\rangle} s_i s_j\Big)} = \left\langle e^{h\sum_i s_i}\right\rangle_Q$$

By replacing

$$e^{h\sum_i s_i} \to e^{h\sum_i s_i} - \textbf{CV}$$

and optimize **CV** to minimize

$$\mathrm{Var}\left(e^{h\sum_i s_i} - \textbf{CV}\right)$$

Extreme learning machine

1. Prepare basis functions

$$\left\{\sum s, \cos(\sum s), \sin(\sum s)\right\} \times s \times S(h=0)^n$$

$(0 \leq n \leq 3)$

2. Input basis functions to ELM

3. Take "divergence"

$$F_i = f(s_i) - f(-s_i)$$

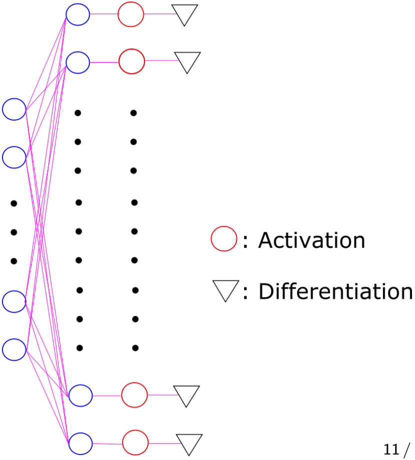
4. The $CV = \sum_i c_i F_i$

The coefficients c_i are optimized by estimating

$$M_{ij} = \langle F_i F_j \rangle, v_j = \langle \mathcal{O} F_i \rangle$$

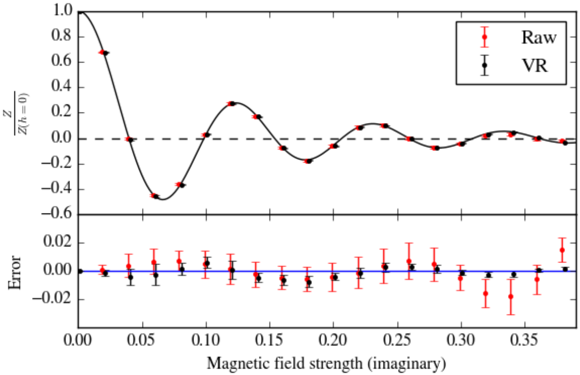
and

$$c = M^{-1}\ v$$



Classical Ising model[4]

At purely imaginary h, $J = 0.4 < J_c \approx 0.441$, 8×8 lattice:



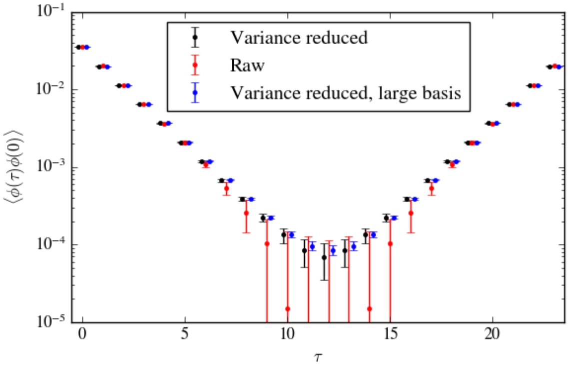
- Raw: 5k samples for Z
- VR: 5k samples to optimize, 5k samples for Z

[4]S. Lawrence and **YY**, in preparation

Scaler field theory[5]

Lattice scalar ϕ^4 theory in Euclidean

$$S = \sum_{\langle r,r'\rangle} \frac{(\phi(r)-\phi(r'))^2}{2} + \sum_r \left[\frac{m^2}{2}\phi^2(r) + \frac{\lambda}{24}\phi^4(r)\right]$$



24×24 lattice, $m^2 = 0.0, \lambda = 2.0$

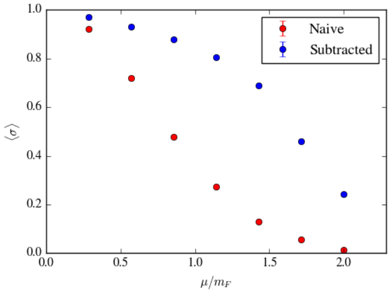
[5] T. Bhattacharya, S. Lawrence, and J. Yoo, arXiv:2307.14950 [hep-lat]

Thirring model in $1+1$-dimension at finite density [6]

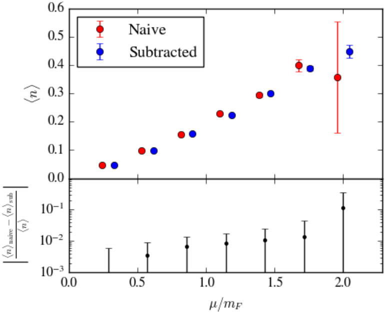
$$S = \sum_{x,\nu} \frac{2}{g^2} \left(1 - \cos A_\nu(x)\right) - \log\det K,\, A_\nu \in [0, 2\pi)$$

with the Dirac matrix ($\eta_0 = (-1)^{\delta_{0,x_0}}$ and $\eta_1 = (-1)^{x_0}$)

$$K[A]_{xy} = m\delta_{xy} + \frac{1}{2}\sum_{\nu=0,1} \eta_\nu e^{iA_\nu(x)+\mu\delta_{\nu,0}}\delta_{x+\nu,y} - \eta_\nu e^{-iA_\nu(y)-\mu\delta_{\nu,0}}\delta_{y+\nu,x}$$



Average sign



Density

4×4 lattice, $m = 0.05, g = 1.0 \to m_B = 0.33(1), m_F = 0.35(2)$

[6] S. Lawrence and **YY**, arXiv:2212.14606 [hep-lat]

Future

Bayesian analysis via NF

- Efficient training algorithms
- Continuous calibration

Lattice control variates

- Application to LQCD
- Towards sign problems

Collaborators (?)

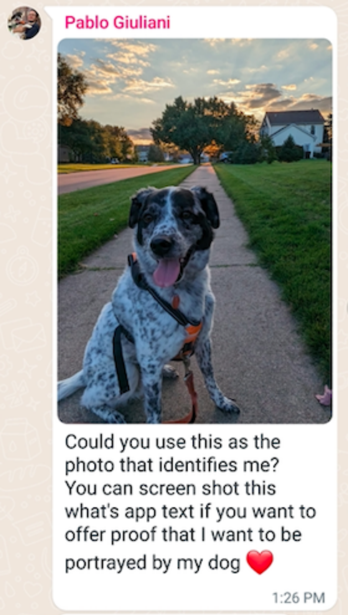
Landon Buskirk

Pablo Giuliani

Kyle Godbey

Scott Lawrence

BACK UP


Pablo Giuliani
Could you use this as the photo that identifies me? You can screen shot this what's app text if you want to offer proof that I want to be portrayed by my dog
1:26 PM

by