Normalizing flows for Bayesian Posteriors

Yukari Yamauchi

Based on

- arXiv:2310.04635 with Landon Buskirk, Pablo Giuliani, Kyle Godbey (FRIB)
- Ongoing work with Lindsey Schneider (INT)

Codes: https://gitlab.com/yyamauchi/uq_nf

August 23rd, 2024, INT workshop "Heavy Ion Physics in the Era of EIC"

INSTITUTE for NUCLEAR THEORY

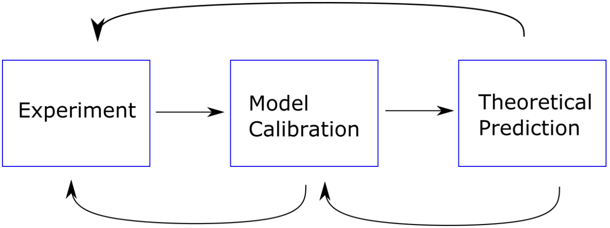
Bayesian analysis

Bayes theorem provides a posterior distribution of model parameters $\boldsymbol{\alpha}$

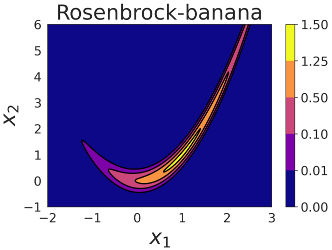
$$P(\boldsymbol{\alpha}|\boldsymbol{Y}) = \frac{P(\boldsymbol{Y}|\boldsymbol{\alpha})P(\boldsymbol{\alpha})}{P(\boldsymbol{Y})}$$

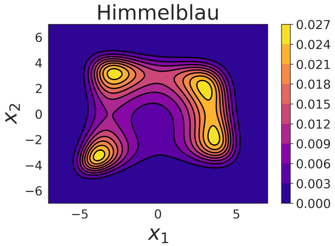
given observation $\boldsymbol{Y}$.

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

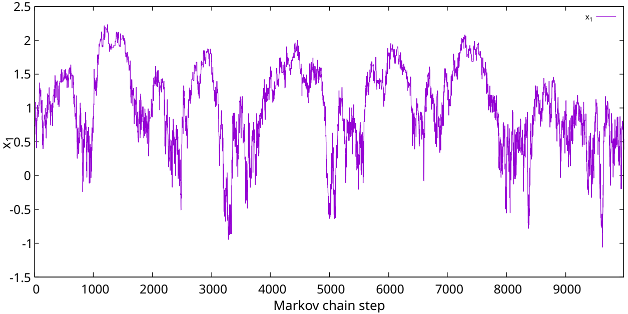


Challenges in Markov-chain Monte Carlo simulation





Auto-correlations



Sampling problems in nuclear physics

Lattice QCD

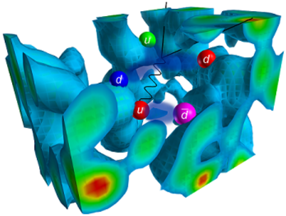


Image credit: Derek Leinweber

Many-body wavefunction

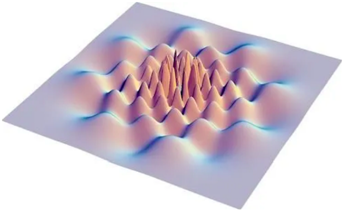


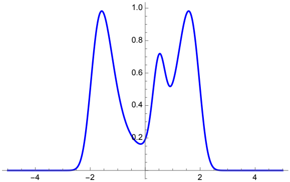
Image credit: David Tong

$$\langle \mathcal{O} \rangle = \frac{\int d\vec{x}\ P(\vec{x})\ \mathcal{O}(\vec{x})}{\int d\vec{x}\ P(x)}$$

Normalizing flows

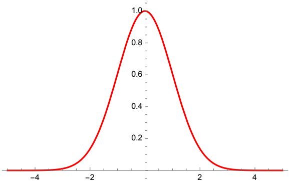
is a map $\boldsymbol{\alpha} \leftrightarrow \boldsymbol{x}$:

Some distribution



Map
$\leftrightarrow$
$\boldsymbol{\alpha}(\boldsymbol{x})$

Gaussian distribution



$$\det\left(\frac{\partial\vec{\boldsymbol{\alpha}}}{\partial\vec{\boldsymbol{x}}}\right) P(\boldsymbol{\alpha}) = \mathcal{N}\prod_{i=1}^{N}\frac{1}{\sqrt{2\pi}}e^{-x_i^2/2} = \mathcal{N}\, G_N(\boldsymbol{x})$$

if such a map exists, then

$$\langle\mathcal{O}\rangle = \frac{\int d\boldsymbol{\alpha}\; P(\boldsymbol{\alpha})\mathcal{O}(\boldsymbol{\alpha})}{\int d\boldsymbol{\alpha}\; P(\boldsymbol{\alpha})} = \frac{\int d\boldsymbol{x}\; G_N(\boldsymbol{x})\mathcal{O}(\boldsymbol{x}(\boldsymbol{\alpha}))}{\int d\boldsymbol{x}\; G_N(\boldsymbol{x})}$$

Machine-learned normalizing flow

Find an approximate normalizing flow

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

by optimizing parameters in the neural network.

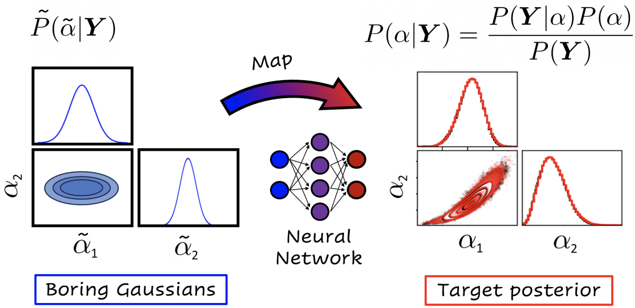


Image credit: Pablu Giuliani

Normalizing flow is approximate

$P(\boldsymbol{\alpha})$: True posterior distribution
$Q(\boldsymbol{\alpha})$: Distribution induced from normalizing flow

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

To sample from $P(\boldsymbol{\alpha})$, Reweight

for samples drawn from $Q(\boldsymbol{\alpha})$, assign the weight $\frac{P(\boldsymbol{\alpha})}{Q(\boldsymbol{\alpha})}$

To measure $\langle \mathcal{O} \rangle$

$$\langle \mathcal{O} \rangle = \frac{\int d\boldsymbol{\alpha} \; P(\boldsymbol{\alpha}) \; \mathcal{O}(\boldsymbol{\alpha})}{\int d\boldsymbol{\alpha} \; P(\boldsymbol{\alpha})} = \frac{\langle \frac{P(\boldsymbol{\alpha})}{Q(\boldsymbol{\alpha})} \; \mathcal{O} \rangle_Q}{\langle \frac{P(\boldsymbol{\alpha})}{Q(\boldsymbol{\alpha})} \rangle_Q}$$

Algorithm

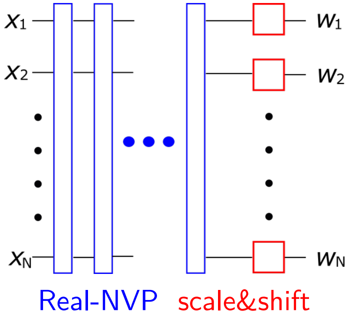
Goal:

Construct a normalizing flow that yields $\approx P(\alpha|\boldsymbol{Y})$

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

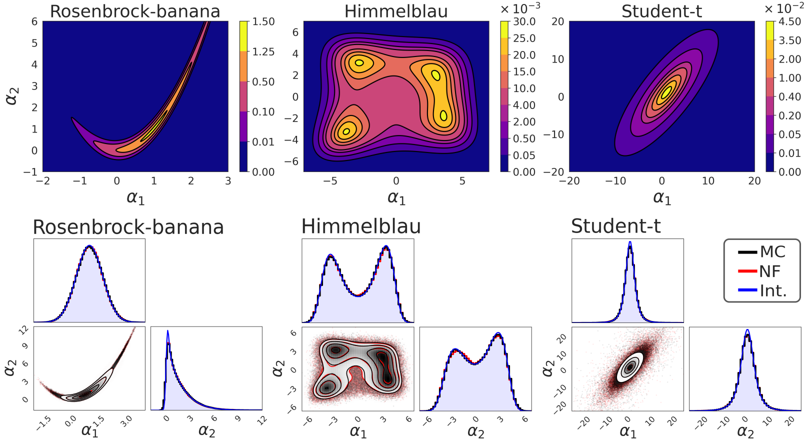
$$D(P|Q) = \int d\alpha \left(\tilde{P} \log \frac{P}{Q} + \tilde{Q} \log \frac{Q}{P}\right)$$

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



2-dimensional examples

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



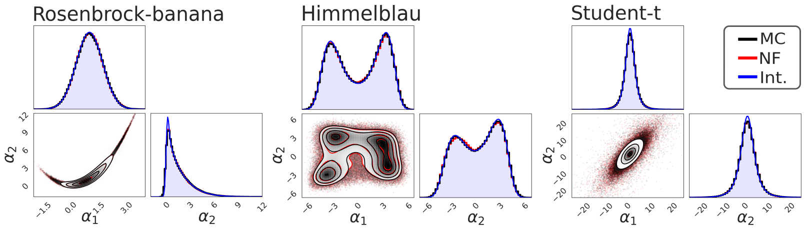
$D =$ 1.0×10^{-5} 3.8×10^{-3} 6.0×10^{-5}

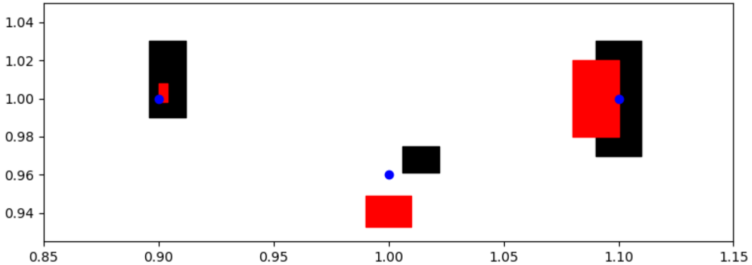
How correct are the normalizing flows?

MCMC samples: 10^5 samples with 100-step skip

NF samples: 10^5 samples

Correlation ~ 1000 ~ 100 ~ 100





More demonstration: relativistic mean field model[2]

A model of nucleons, mesons (σ, ω, ρ) and photon

$$\begin{aligned}\mathcal{L}_{\rm int} &= \bar{\psi}\left[g_s\phi - \left(g_v V_\mu + \frac{g_\rho}{2}\tau\cdot b_\mu + \frac{e}{2}(1+\tau_3)A_\mu\right)\gamma^\mu\right]\psi \\ &- \frac{\kappa}{3!}(g_s\phi)^3 - \frac{\lambda}{4!}(g_s\phi)^4 + \frac{\zeta}{4!}{g_v}^4(V_\mu V^\mu)^2 + \Lambda_v\left({g_\rho}^2 b_\mu\cdot b^\mu\right)\left({g_v}^2 V_\nu V^\nu\right)\end{aligned}$$

with

ψ: nucleons, ϕ: σ meson, V: ω meson, b: ρ meson

In the mean-field limit

quantum fields $\rightarrow$ expectation values

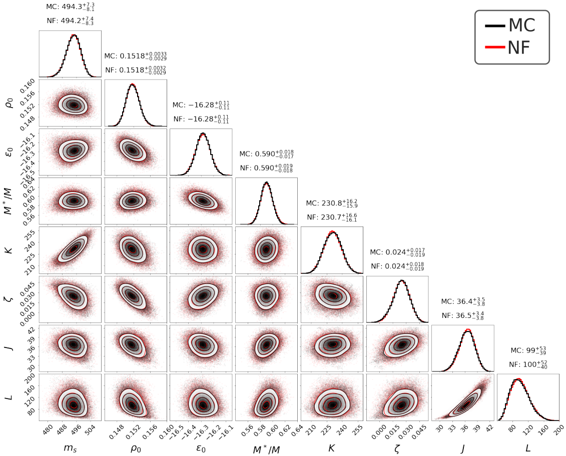
The parameters to calibrate[1]

m_s, g_s, g_v, g_ρ, κ, λ, Λ_v, ζ

($m_\omega = 783$ MeV, $m_\rho = 763$ MeV)
Calibrated by experimental binding energies and charge radii of nuclei
(^{16}O, ^{40}Ca, ^{48}Ca, ^{68}Ni, ^{90}Zr, ^{100}Sn, ^{116}Sn, ^{132}Sn, ^{144}Sm, ^{208}Pb)

[1] P. Giuliani, K. Godbey, et al., arXiv:2209.13039[nucl-th]
[2] W. Chen and J. Piekarewicz, arXiv:1408.4149

Normalizing flows for the relativistic mean field model[3]



- 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

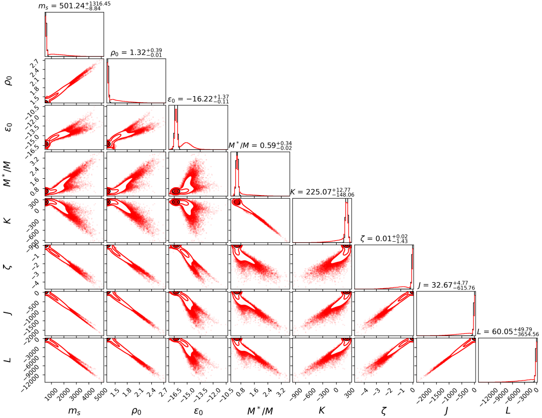
$$D : 10 \to 2 \times 10^{-3}$$

All codes developed in this work are available at https://gitlab.com/yyamauchi/uq_nf

[3] **YY**, L. Buskirk, P. Giuliani, and K. Godbey, arXiv:2310.04635 [nucl-th]

An issue with supervised training

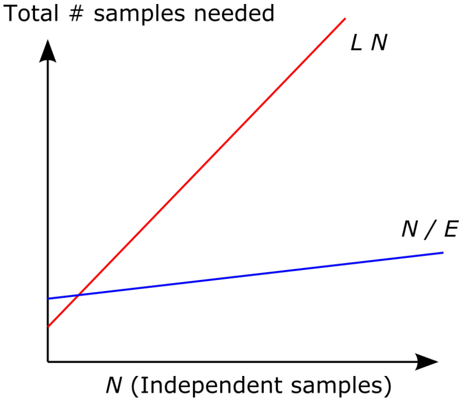
Outcome (NF) depends on your data (MCMC samples) quality



Self learning. No MCMC. Always sample from Gaussian

For N **independent** samples

- MCMC : $N_{\text{therm}} + L \times N$
- NF : $N_{\text{train}} + N \ / \ E$ with $E = \frac{1}{\langle (P/Q)^2 \rangle_Q}$ (effective sampling size)



Summary and outlook

Why NF?

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

Further algorithmic development:

- A more efficient and universal algorithm.
- Rigorous quantification of the correctness of NF.
- Establishing an online platform to share the resources.

Applications:

- BSM physics with Vincenzo Cirigliano, Wouter Dekens, Sebastián Urrutia-Quiroga
- Color Glass Condensate physics with Farid Salazar
- $\cdots$

Thank you!