INL/CON-23-73727-Revision-0

Bayesian Inference with Latent Hamiltonian Neural Networks (L-HNNs)

July 2023

Changing the World's Energy Future

Som LakshmiNarasimha Dhulipala, Yifeng Che, Michael Shields

Idaho National Laboratory

INL is a U.S. Department of Energy National Laboratory operated by Battelle Energy Alliance, LLC

DISCLAIMER

This information was prepared as an account of work sponsored by an agency of the U.S. Government. Neither the U.S. Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness, of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof.

INL/CON-23-73727-Revision-0

Bayesian Inference with Latent Hamiltonian Neural Networks (L-HNNs)

Som LakshmiNarasimha Dhulipala, Yifeng Che, Michael Shields

July 2023

Idaho National Laboratory
Idaho Falls, Idaho 83415

http://www.inl.gov

Prepared for the
U.S. Department of Energy
Under DOE Idaho Operations Office
Contract DE-AC07-05ID14517

Bayesian Inference with Latent Hamiltonian Neural Networks (L-HNNs)

Som L. Dhulipala, Yifeng Che
Idaho National Laboratory

Michael D. Shields
Johns Hopkins University

17th US National Congress on Computational Mechanics

Johns Hopkins University
Idaho National Laboratory

Motivation

Variational inference

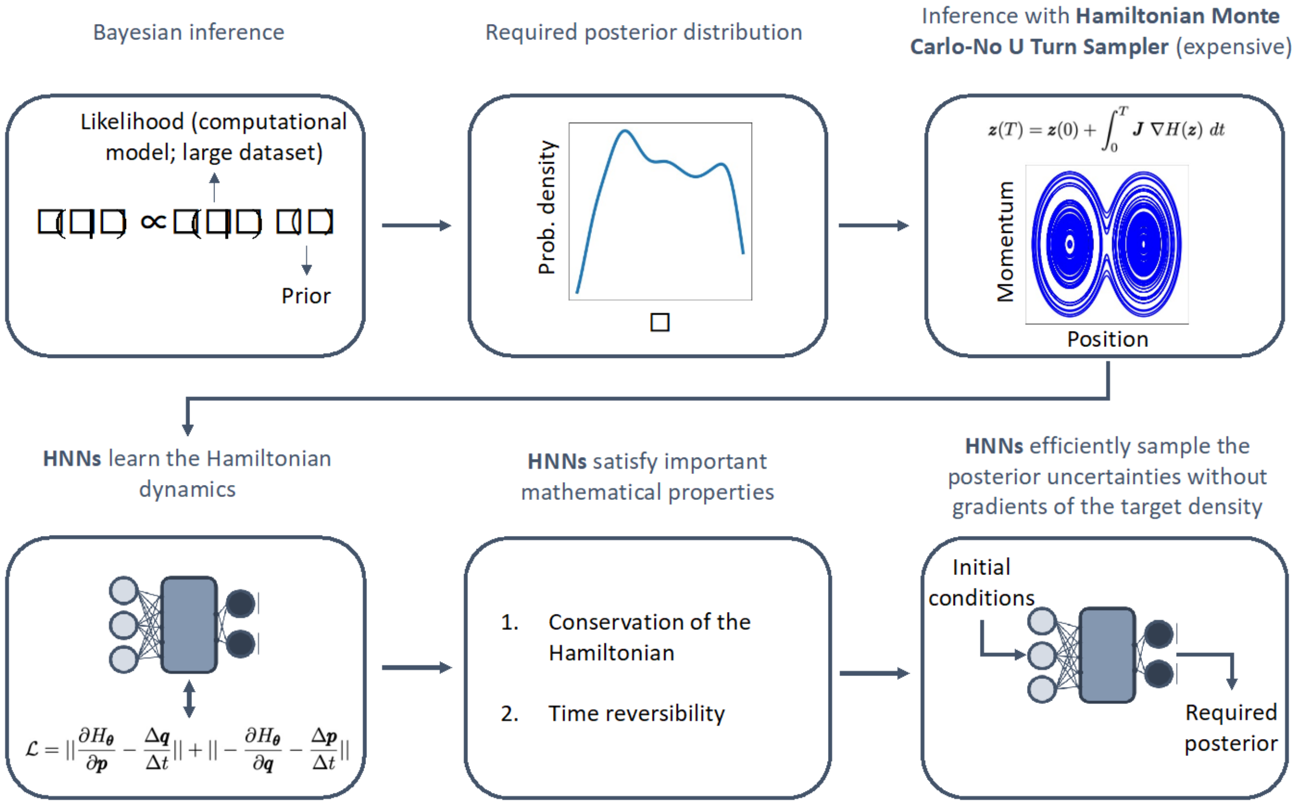
Markov-chain Monte Carlo

Particle-based inference (Stein variational gradient descent)

Normalizing flows

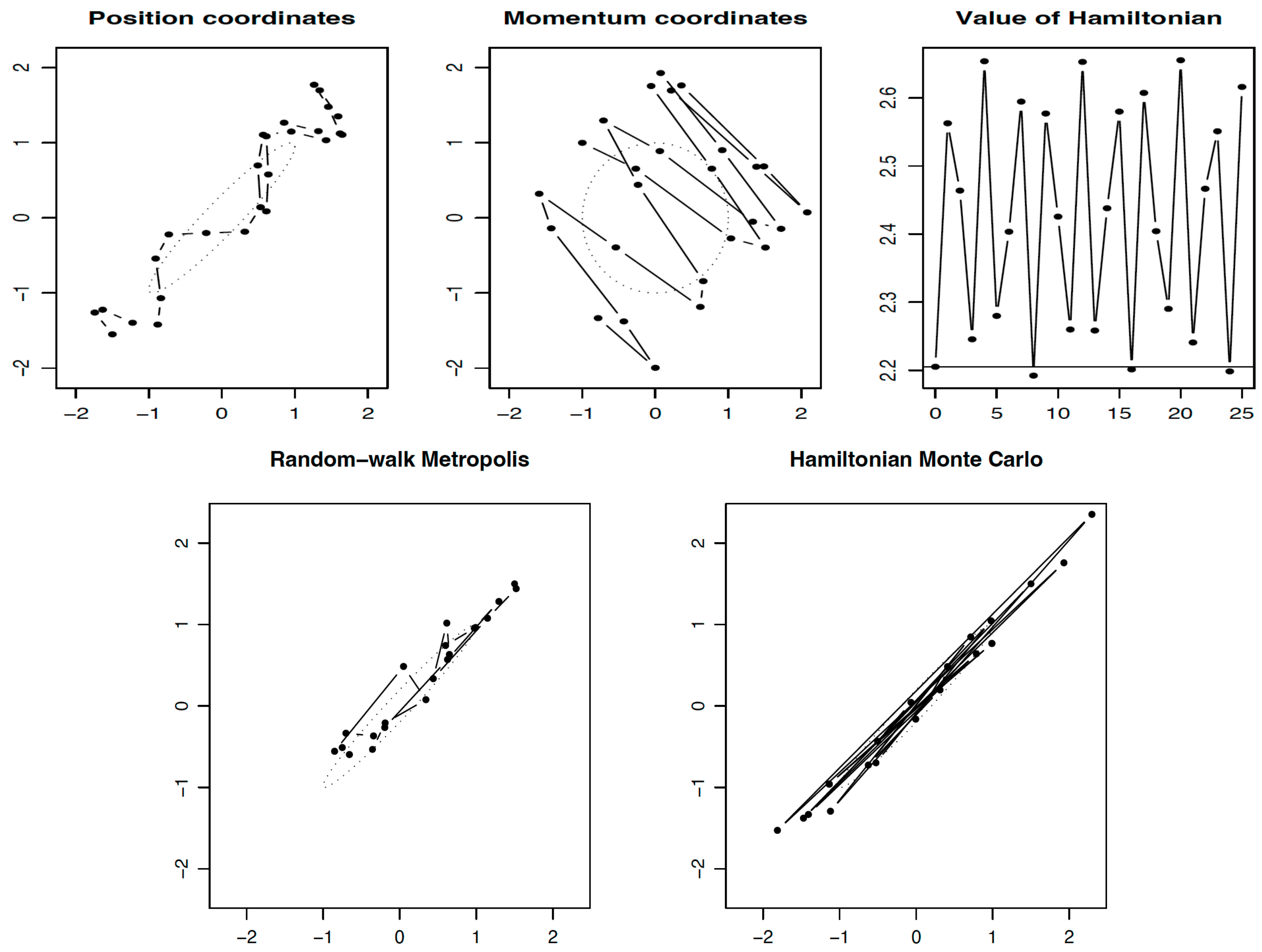
Least expensive	Moderately expensive	Very expensive
Random walk	**Langevin-based**	**Hamiltonian-based**
• No gradient information required • Poor scalability with dimensionality • Large correlations b/w samples	• Uses gradient information • Better scalability with dimensionality • Fairly large correlations b/w samples	• Uses gradient information • Best scalability with dimensionality • Small correlations b/w samples

Proposed contributions



- Use of Hamiltonian Neural Networks (HNNs) for Bayesian inference without needing numerous gradients of target posterior
- Introduction of latent variable outputs to HNNs (L-HNNs) for improved expressivity
- L-HNNs in No-U-Turn Sampling (NUTS) with an online error monitoring scheme

Background: Hamiltonian Monte Carlo



Credit: Neal (2013)

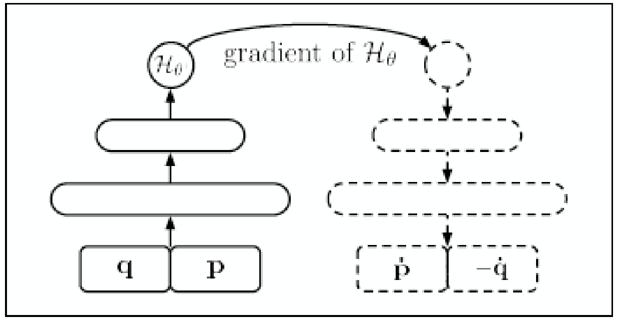
- Sampling is "similar" to MCMC: Random momenta, compute particle trajectory by solving the governing equation, Metropolis acceptance step
- positions **q** and momenta **p**:

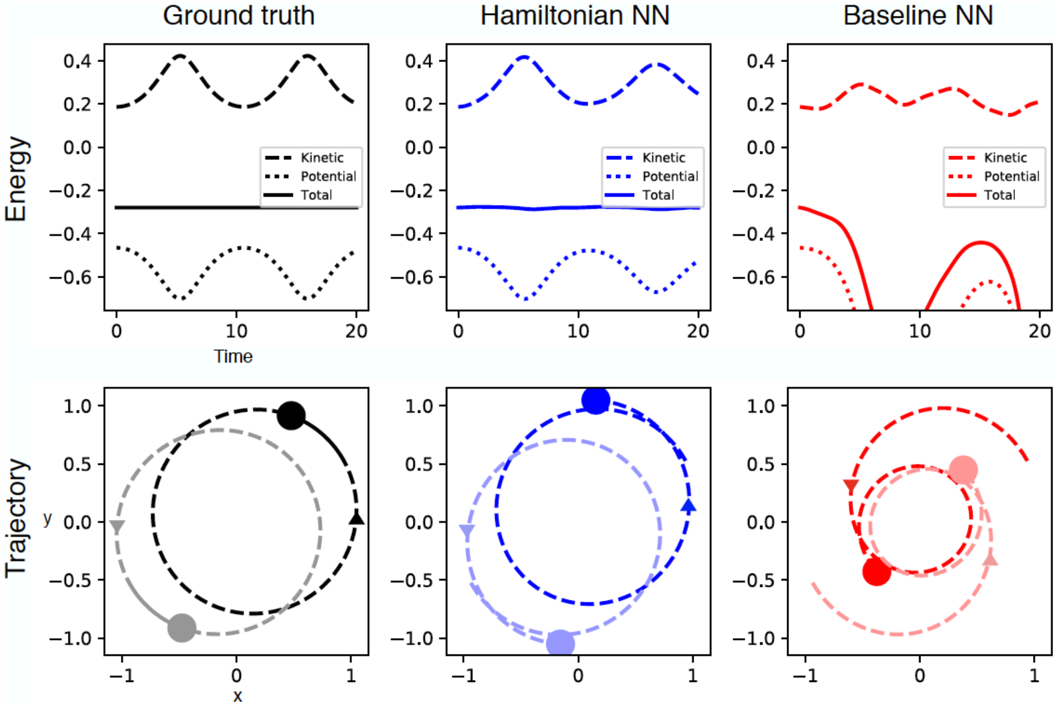
$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \qquad \forall \quad i \in [1, \ldots, d].$$
$$\frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

- Hamiltonian (H) definition: Potential energy U(**q**) is dependent on the target posterior and Kinetic energy K(**p**) is based on "Canonical distribution"
- Exact numerical integration: all proposed samples accepted
- Better scalability than random-walk due to use of gradient information

Background: Hamiltonian Neural Networks (HNNs)

Hamiltonian NN





Credit: Greydanus et al. (2019)

- In forward pass, HNNs predict the Hamiltonian (H) and gradients of it are computed
- HNNs minimize the loss function:

$$\underset{\theta}{\operatorname{argmin}} \left\| \frac{d\mathbf{q}}{dt} - \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{p}} \right\|^2 + \left\| \frac{d\mathbf{p}}{dt} + \frac{\partial \mathcal{H}_\theta}{\partial \mathbf{q}} \right\|^2$$

- Hamilton's equations:

$$\begin{aligned} \frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i} \end{aligned} \qquad \forall \quad i \in [1, \ldots, d].$$

- Better performance than Neural Nets
- HNNs applied to classical mechanics problems
- Not used for Bayesian inference (to our knowledge)

Bayesian inference with HNNs

Leap frog integration with HNNs for solving: $\begin{aligned} \frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i} \end{aligned} \quad \forall \quad i \in [1, \ldots, d].$

Hamiltonian Monte Carlo

Algorithm 3.3 HNNs in Hamiltonian Monte Carlo

```
Hamiltonian: H = U(q) + K(p); Samples: M; Starting sample: {q^0, p^0}; End time for trajectory: T; Steps: N; Dimensions: d
for i = 1 : M do
    p(0) ~ N(0, I_d)
    q(0) = q^{i-1}
    Compute {q*, p*} = {q(T), p(T)} with Algorithm 3.2
    α = min{1, exp(H({q^{i-1}, p^{i-1}}) − H({q*, p*}))}
    With probability α, set {q^i, p^i} ← {q*, p*}
end for
```

$$U(\boldsymbol{q}) \propto -\log\left[\mathcal{L}(\boldsymbol{q}|D)\ g(\boldsymbol{q})\right]$$

$$K(\boldsymbol{p}) \propto -\log[N(\boldsymbol{0}, \boldsymbol{I})]$$

Algorithm 3.2 Hamiltonian neural networks evaluation in leapfrog integration

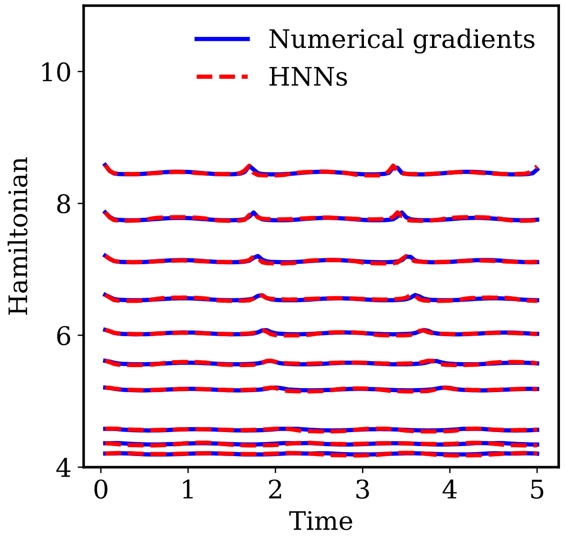
```
Hamiltonian: H; Initial conditions: z(0) = {q(0), p(0)}; Dimensions: d; Steps: N; End time: T
Δt = T/N
for j = 0 : N − 1 do
    t = j Δt
    Compute HNN output gradient ∂H_θ/∂q(t)
    for i = 1 : d do
        q_i(t + Δt) = q_i(t) + (Δt/m_i) p_i(t) − (Δt^2/2m_i) ∂H_θ/∂q_i(t)
    end for
    Compute HNN output gradient ∂H_θ/∂q(t+Δt)
    for i = 1 : d do
        p_i(t + Δt) = p_i(t) − (Δt/2) (∂H_θ/∂q_i(t) + ∂H_θ/∂q_i(t+Δt))
    end for
end for
```

Using HNNs (traditional: numerical gradients)

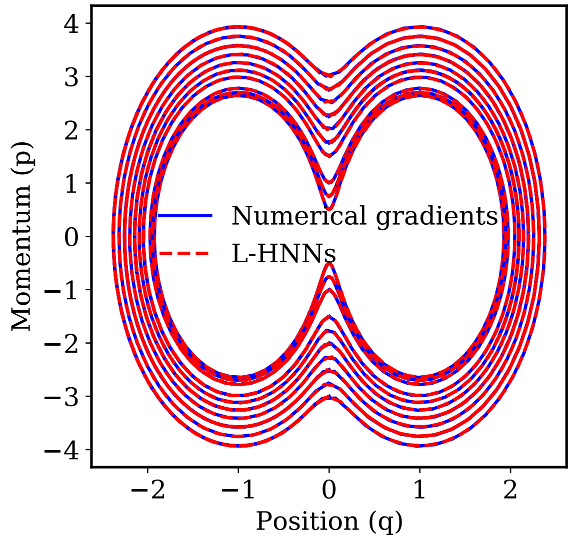
Bayesian inference with HNNs

1-D Gaussian mixture density

$$f(q) \propto 0.5 \ \exp\left(\frac{(q-1)^2}{2 \times 0.35^2}\right) + 0.5 \ \exp\left(\frac{(q+1)^2}{2 \times 0.35^2}\right)$$



Hamiltonian conservation for some initial ***q*-*p*** values

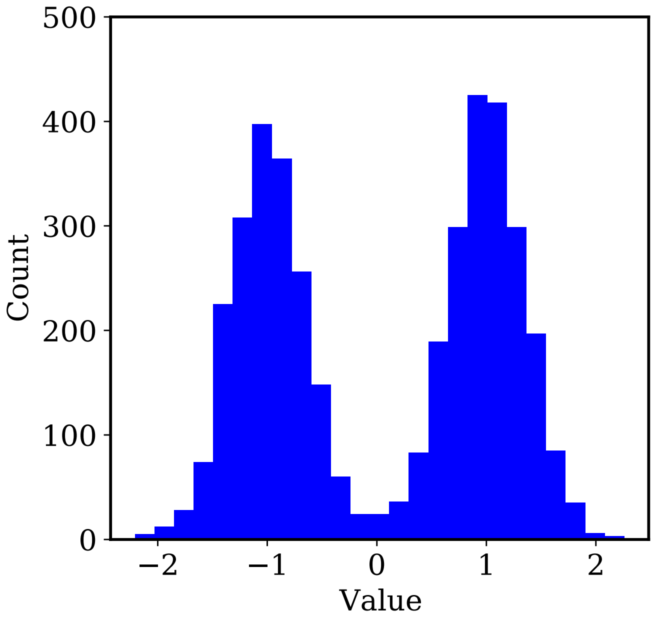


q*-*p phase space plots for some initial values

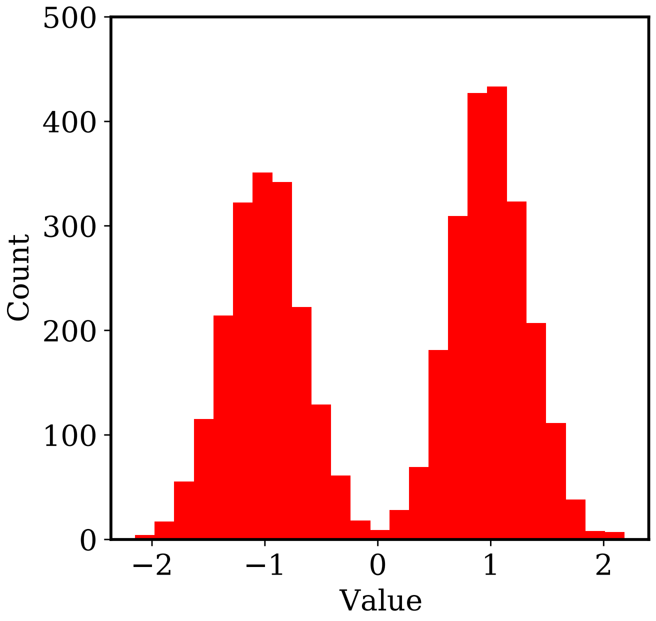
Bayesian inference with HNNs

1-D Gaussian mixture density

$$f(q) \propto 0.5 \ \exp\left(\frac{(q-1)^2}{2 \times 0.35^2}\right) + 0.5 \ \exp\left(\frac{(q+1)^2}{2 \times 0.35^2}\right)$$



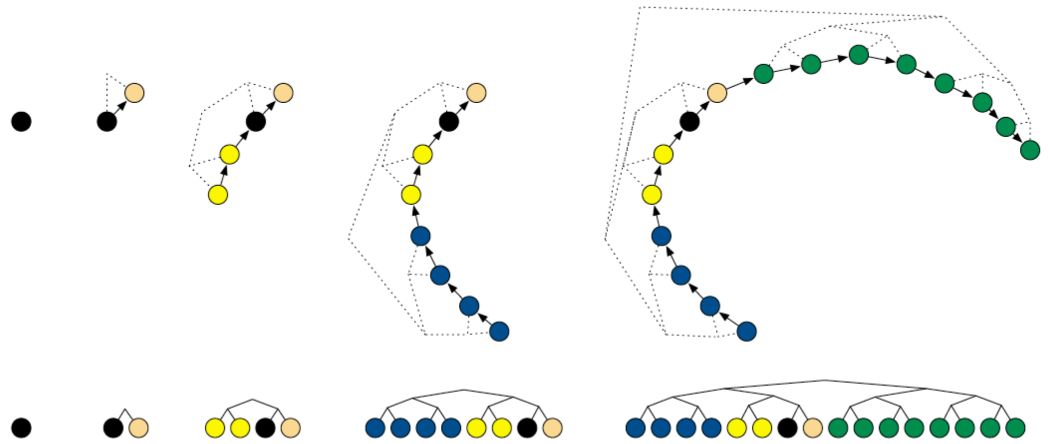
Samples using traditional HMC. 500,000 posterior gradients



Samples using HNNs in HMC. 8,000 posterior gradients

Background: No-U-Turn Sampling (NUTS)

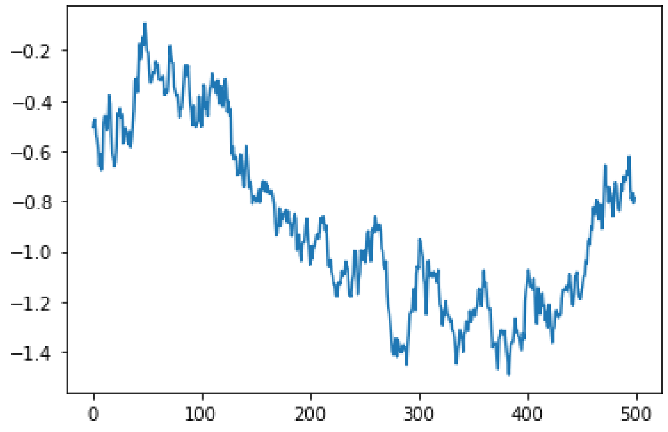
Theory



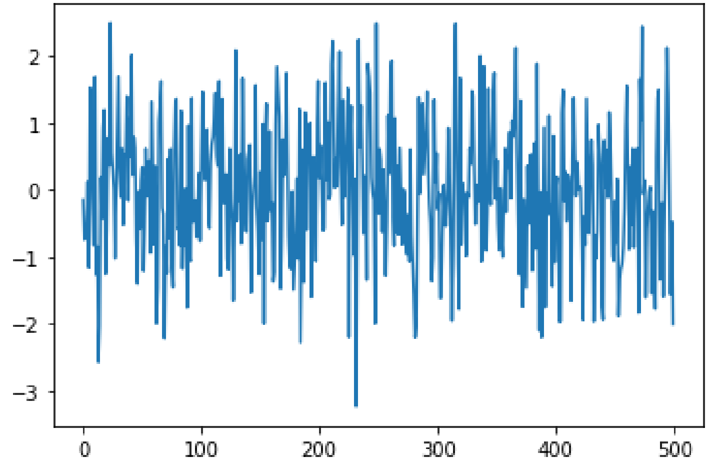
Credit: Hoffman and Gelman (2013)

- Successive binary trees with doubling length either in positive or negative direction
- "Slice sampling" to select **q-p** states that satisfy stationarity
- Tree building terminated when a u-turn is made (correlations between samples are small)
- Also terminated when integration errors are large when simulating the Hamiltonian dynamics

Practice (a high-dimensional posterior)



HMC with an end time of 5 units

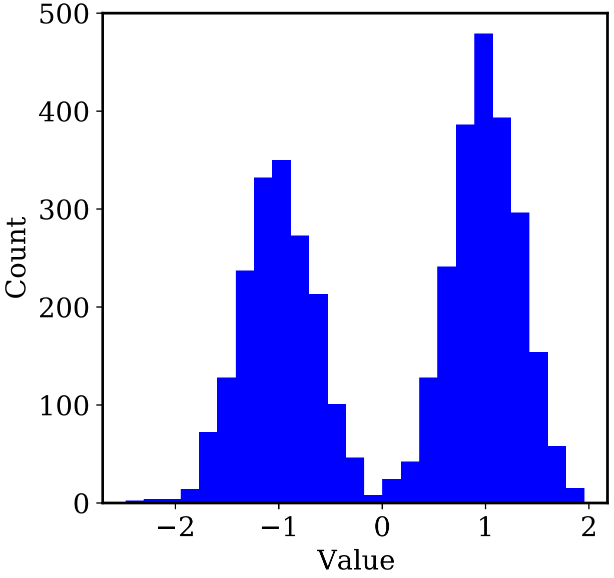


NUTS

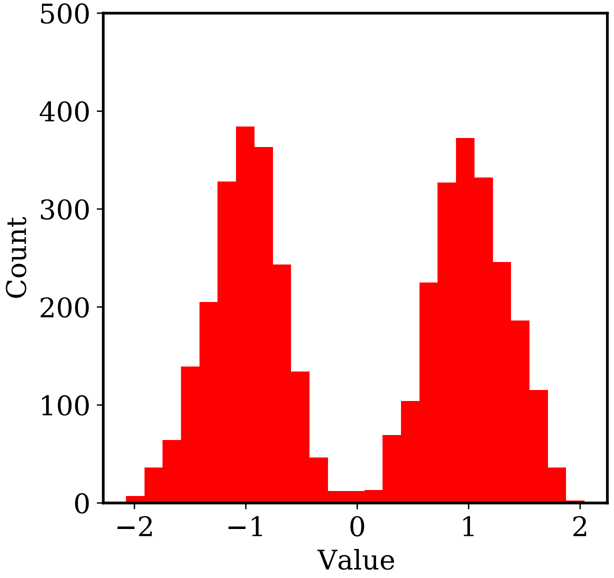
HNNs in NUTS

1-D Gaussian mixture density

$$f(q) \propto 0.5\ \exp\left(\frac{(q-1)^2}{2\times 0.35^2}\right) + 0.5\ \exp\left(\frac{(q+1)^2}{2\times 0.35^2}\right)$$



Samples using traditional NUTS



Samples using HNNs in NUTS

ESS: effective sample size to measure sampling quality

	ESS per gradient of target density
HNNs in HMC	$4.59E-3$
Traditional HMC	$8.42E-5$
HNNs in NUTS	$4.83E-3$
Traditional NUTS	$4.4E-4$

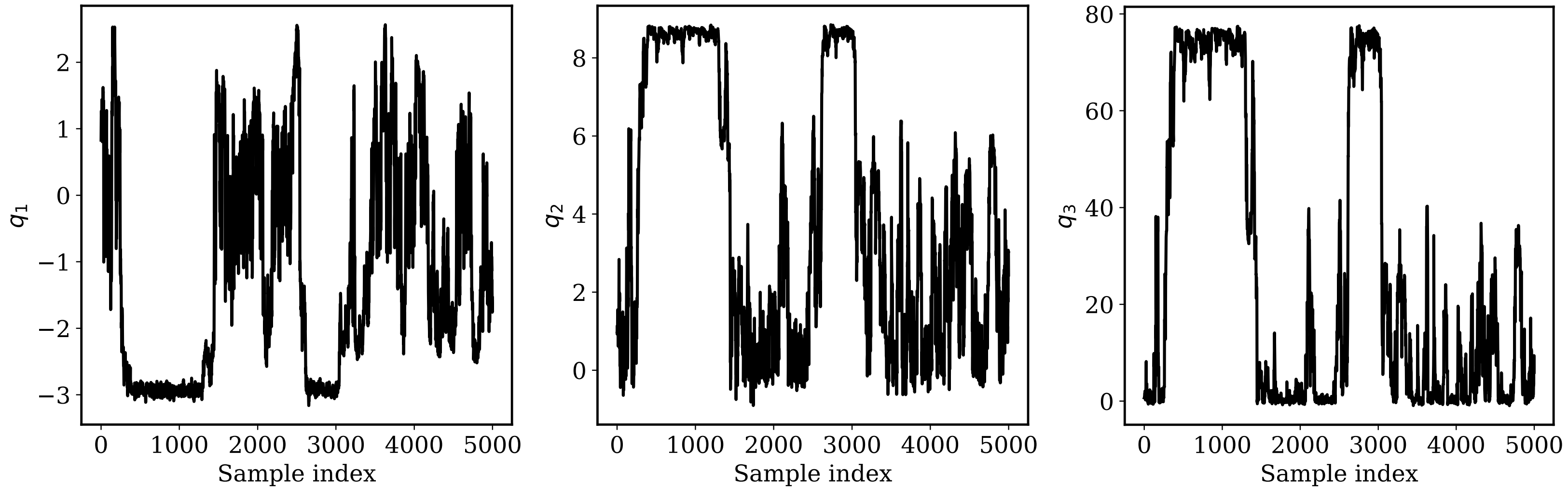
490,000 HNN gradients

91,000 HNN gradients

HNNs in NUTS: sampling degeneracy problem

3-D Rosenbrock density

$$f(\boldsymbol{q}) \propto \exp\left(-\sum_{i=1}^{N-1}\left[100(q_{i+1}-q_i^2)^2+(1-q_i)^2\right]/20\right)$$



Dimension 1 Dimension 2 Dimension 3

- In the tails, HNNs may have little to no training data
- Therefore, integration errors when simulating Hamiltonian dynamics using HNNs can be large
- As a result, NUTS prematurely terminates the tree building procedure
- Many sample clusters in regions of low density: sampling degeneracy problem

IDAHO NATIONAL LABORATORY

HNNs in NUTS with online error monitoring

Integration error in the original NUTS (H: Hamiltonian; u: slice value; Δ_{max}: threshold [1000])

$$\varepsilon \equiv H(\boldsymbol{z}) + \ln u > \Delta_{max}$$

Proposed online error monitoring

$$\varepsilon^{hnn} \leq \Delta^{hnn}_{max} \rightarrow \text{use leap frog with L-HNN gradients}$$

$$\varepsilon^{hnn} > \Delta^{hnn}_{max},\ \varepsilon^{lf} \leq \Delta^{lf}_{max} \rightarrow \text{use leap frog with posterior gradients for } N^{lf} \text{ samples}$$

$$\varepsilon^{hnn} > \Delta^{hnn}_{max},\ \varepsilon^{lf} > \Delta^{lf}_{max} \rightarrow \text{terminate tree building; move to next sample}$$

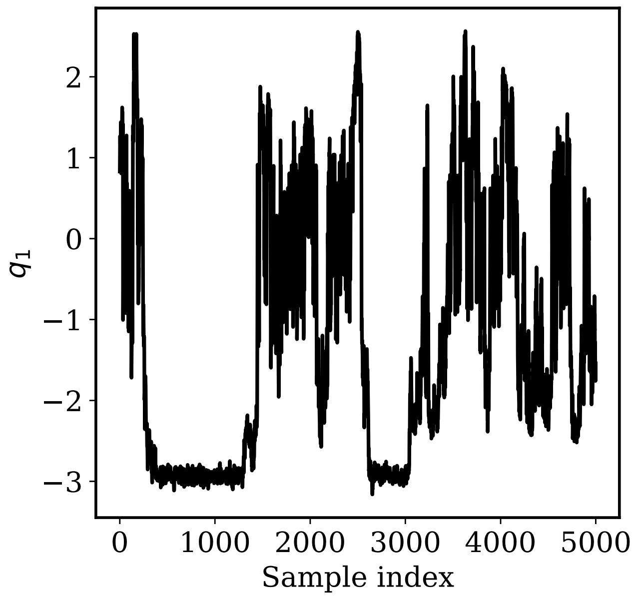
- Original NUTS terminates tree building when $\varepsilon > \Delta_{max}$
- Causes sampling degeneracy in the tails when using NUTS with HNNs
- Proposed online error monitoring:
  - $\Delta^{hnn}{}_{max}$: when using HNNs and $\Delta^{lf}{}_{max}$: when using posterior gradients
  - $\Delta^{hnn}{}_{max} \ll \Delta^{lf}{}_{max}$ (e.g., 10 and 1000)
  - Default use HNN gradients
  - Switch to posterior gradients for N_{lf} samples (e.g., 5-20) to bring sampler to high density regions
  - Then, HNN gradients take over

- Similar in concept to multifidelity modeling

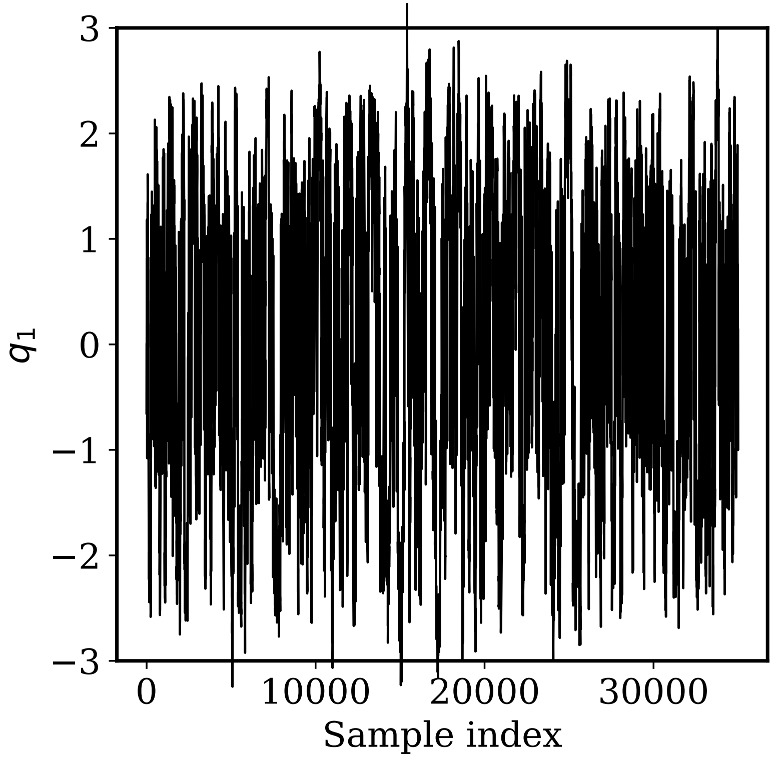
HNNs in NUTS with online error monitoring

3-D Rosenbrock density

$$f(\boldsymbol{q}) \propto \exp\left(-\sum_{i=1}^{N-1}\left[100(q_{i+1}-q_i^2)^2+(1-q_i)^2\right]/20\right)$$



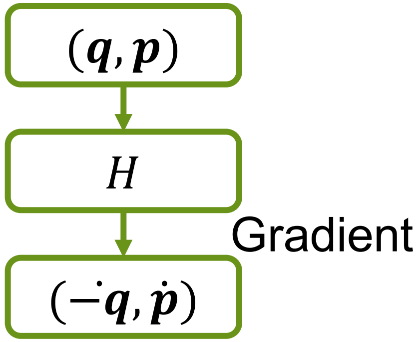
Naïve HNNs in NUTS



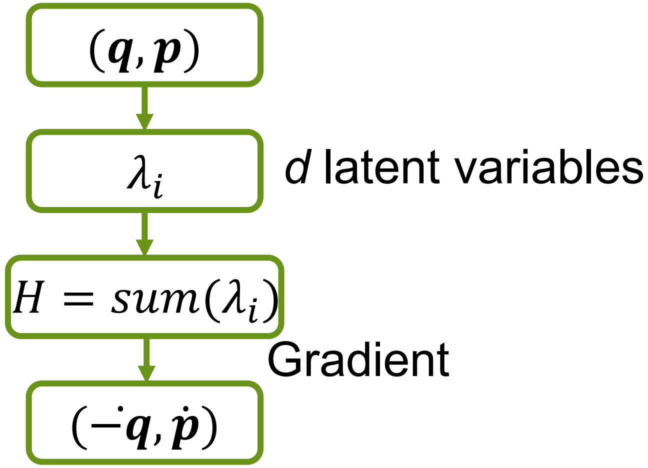
HNNs in NUTS with error monitoring (calls a few posterior gradients during the sampling)

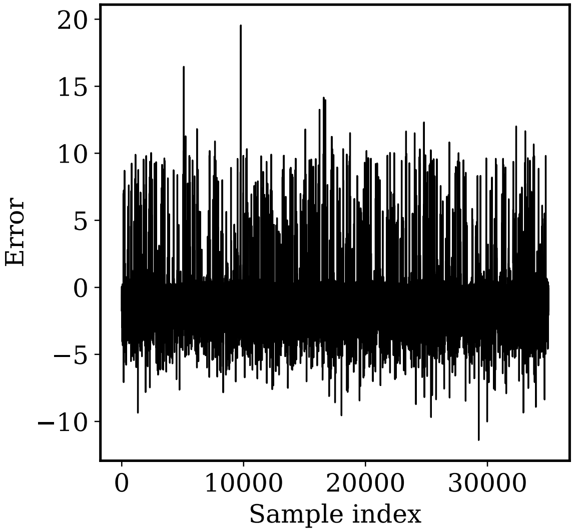
Latent output HNNs (L-HNNs)

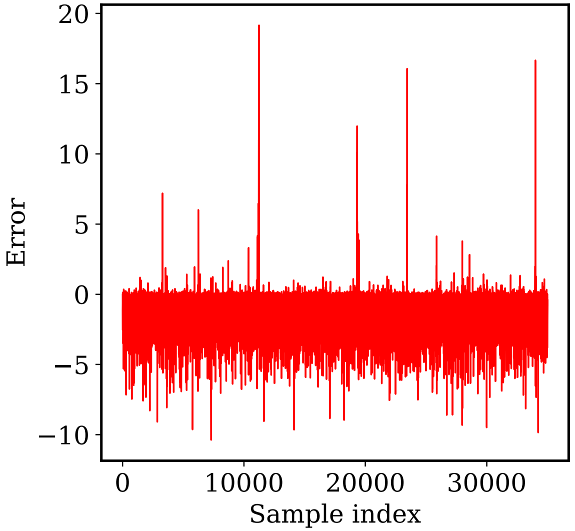
HNNs



L-HNNs





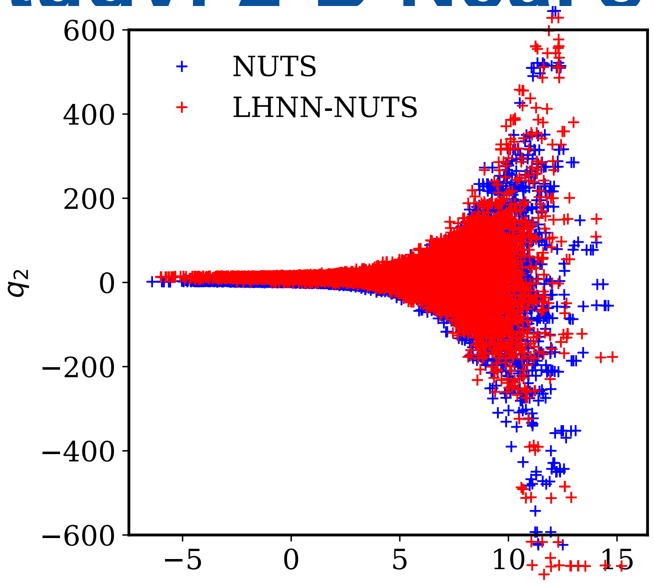


$$\varepsilon \equiv H(\boldsymbol{z}) + \ln u > \Delta_{max}$$

- 3-D Rosenbrock density
- HNNs and L-HNNs individually used in NUTS with online error monitoring
- HNNs and L-HNNs trained with the same training data
- HNNs: 4,706 samples used posterior gradients with ~1.5 Million posterior gradients during error monitoring
- L-HNNs: 180 samples used posterior gradients with ~0.064 Million posterior gradients during error monitoring

Method	ESS/gradient
Traditional NUTS	0.000128
HNNs in NUTS with online error monitoring	0.000824
L-HNNs in NUTS with online error monitoring	**0.00236**

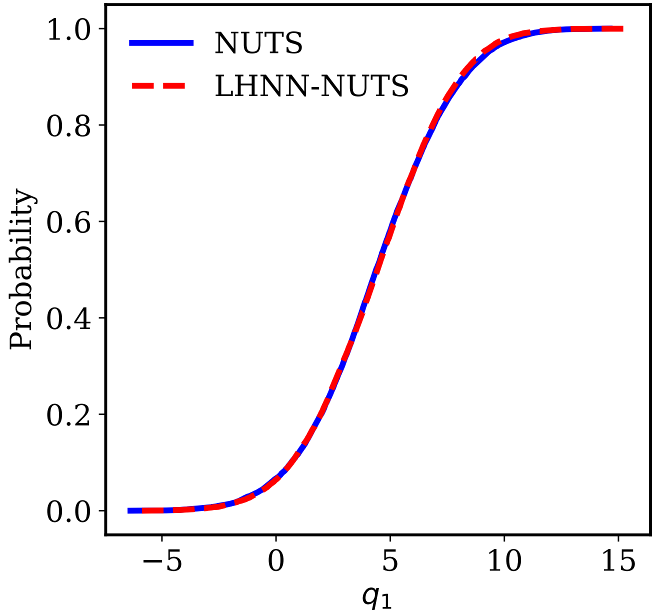
Case study: 2-D Neal's funnel density

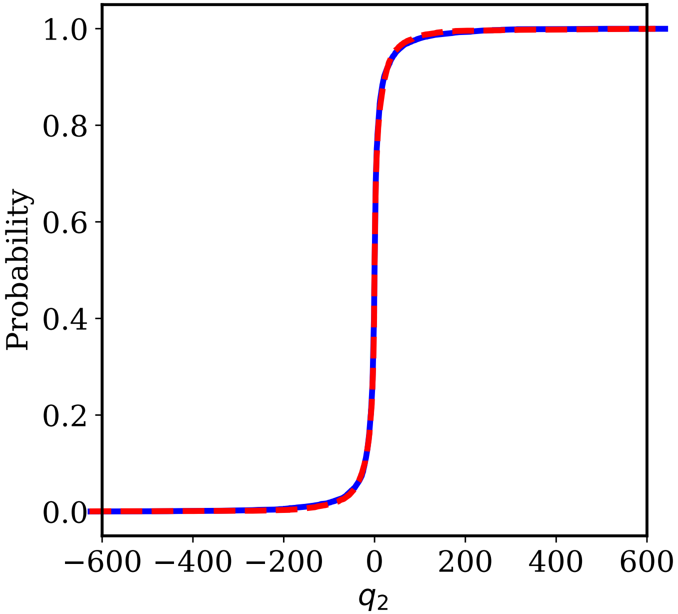


$$f(\boldsymbol{q}) \propto \begin{cases} q_1 = \mathcal{N}(0,\ 3) \\ q_2 = \mathcal{N}(0,\ \exp^{q_1}) \end{cases}$$

25,000 samples

Method	# gradients	ESS/gradient
Traditional NUTS	16.6 Mil	0.000052
L-HNNs in NUTS with online error monitoring (proposed)	**3.59 Mil**	**0.000234**





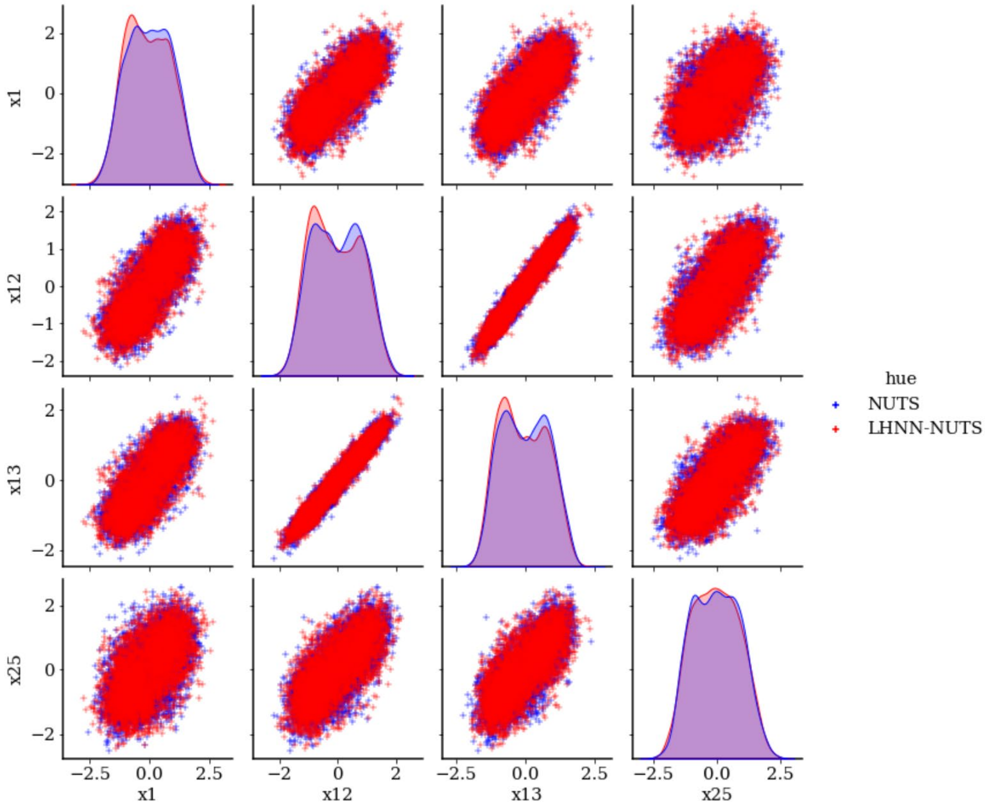
Case study: 25-D Allen-Cahn Equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} - V'(u) + \sqrt{2}\eta$$

where, $V(u) = (1-u^2)^2$

$$\pi(\boldsymbol{u}) \propto \exp\left(-\int_0^1 \left(\frac{1}{2}\left(\frac{\partial u}{\partial x}\right)^2 + V(u(x))\right)dx\right)$$

$$\pi(\boldsymbol{u}) = \exp\left(-\sum_{i=0}^{d-1}\left(\frac{1}{2\Delta x}\big(u(i\Delta x+\Delta x) - u(i\Delta x)\big)^2 + \frac{\Delta x}{2}\Big(V\big(u(i\Delta x+\Delta x)\big) - V\big(u(i\Delta x)\big)\Big)\right)\right)$$



Method	# gradients	ESS/gradient
Traditional NUTS	0.681 Mil	0.00032
L-HNNs in NUTS with online error monitoring (proposed)	**0.097 Mil**	**0.00423**

Goodman and Weare (2010)

Case study: 50-D elliptic PDE

$$\frac{\partial}{\partial x}\left(k(x,y)\,\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(k(x,y)\,\frac{\partial u}{\partial y}\right) = f(x,y)$$

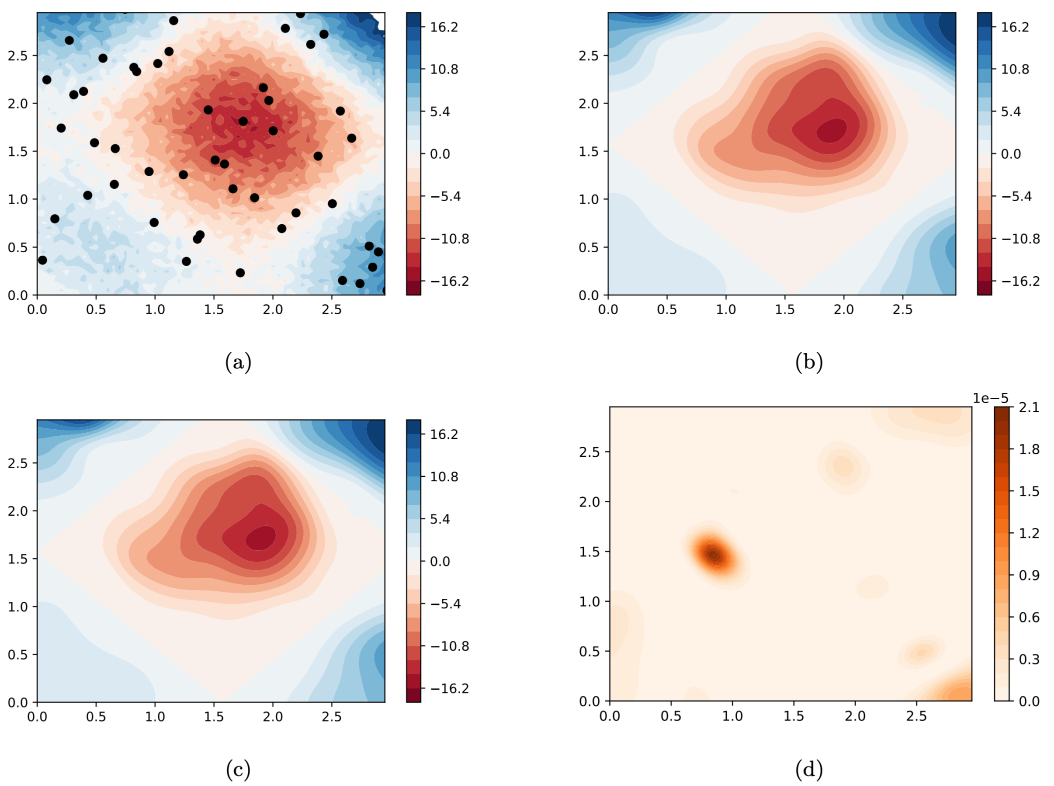
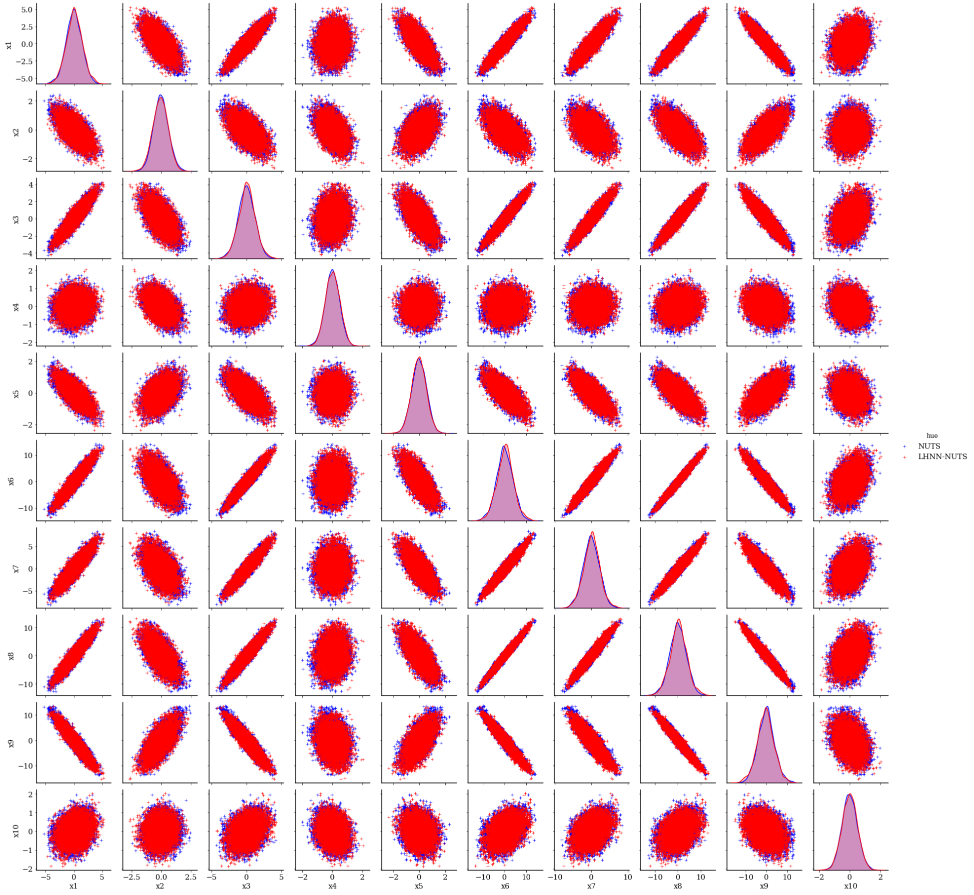


Figure 12: (a) The $f(x,y)$ field corrupted by a Gaussian noise and the points at which $f(x,y)$ is recorded for treating as experimental data. (b) and (c) The estimated mean $f(x,y)$ field after performing Bayesian inference using LHNN-NUTS and NUTS, respectively. (d) Mean squared error of the mean $f(x,y)$ field between LHNN-NUTS and NUTS.

Method	# gradients	ESS/gradient
Traditional NUTS	0.621 Mil	0.0065
L-HNNs in NUTS with online error monitoring (proposed)	**0.217 Mil**	**0.015**

Future work



100-D Gaussian with inverse Wishart covariance (NUTS: 5.61 Mil gradients; LHNN-NUTS: 1.46 Mil gradients)

- **Newer HNN-type architectures:** Physics-informed HNNs and SympNets
- **Sampling with physical constraints:** nuclear fuel model calibration. Spherical HMC with L-HNNs (Yifeng Che is leading this)
- **L-HNNs trained on Manifolds:** Riemannian HMC with L-HNNs; better ESS due to neutralizing bad posterior geometry
- **Active (or online) learning:** Proposed online error monitoring similar in concept; but no re-training of L-HNNs.

Thank you
(Som.Dhulipala@inl.gov)

Resources:

"Bayesian Inference with Latent Hamiltonian Neural Networks" *arXiV:2208.06120*

"Physics-Informed Machine Learning of Dynamical Systems for Efficient Bayesian Inference" *arXiv:2209.09349*