Hamiltonian Monte Carlo: Algorithm, Theory, and Experiments

Genghis Luo, Alex Ni

NYU Shanghai, NYU

2025/04/30

Outline

Outline

Problem Setup I

Suppose we wish to evaluate:

$$u = \int_{\mathcal{X} \subseteq \mathbb{R}^{\widehat{d}}} f(x)\pi(x)dx = \mathbb{E}_{\pi}[f] \tag{1}$$

with an oracle accessing evaluation of $f(x) \in C_0^b(\mathcal{X}), \pi(x) \in \mathcal{P}(\mathcal{X})$ for $\forall x \in \mathcal{X}$.

Problem Setup II

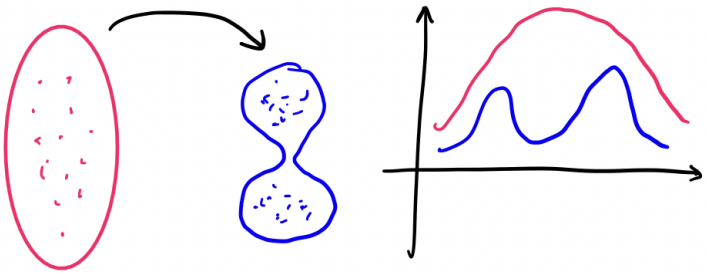


Figure 1: Importance Sampling

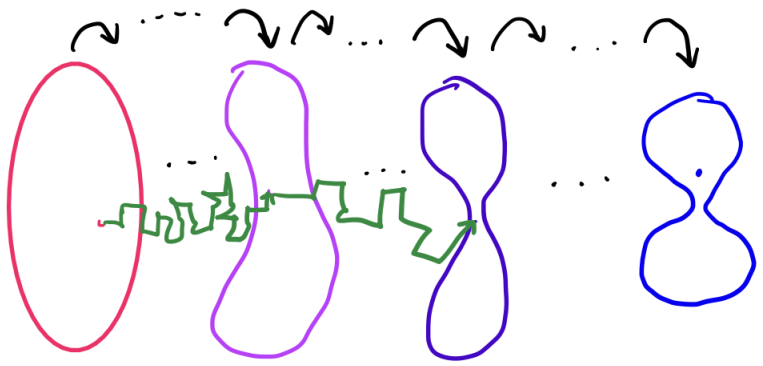


Figure 2: Markov Chain Monte Carlo

Problem Setup III

In principle, we need the chain to satisfy three key properties [1] [2]:

I. The desired distribution π is an **invariant** [3] distribution of the Markov chain, i.e.

$$\int_x \pi(x)P(x,y)dx = \pi(y)$$

II. The chain is **irreducible** if for $\forall x, y \in \mathcal{X}$:

$$\Pr(X_t = x | X_0 = y) > 0 \text{ for some } t > 0$$

III. The chain is **aperiodic** if $\exists t_0 > 0$ s.t.:

$$\Pr(X_{t_0} = x | X_0 = x) > 0 \text{ for } \forall t > t_0$$

---

[1] Indeed, (I)+(II)+(III) implies **ergodicity**, saying $\forall x \in \mathcal{X}$, $\lim_{t\to\infty} \left\| P^t(x,\cdot) - \pi(\cdot) \right\|_{\mathrm{TV}} = 0$

[2] Indeed, (II) and (III) imply **primitivity**, saying $\exists t_0 > 0$ s.t. for $\forall x, y \in \mathcal{X}$: $\Pr(X_t = x | X_0 = y) > 0$ for $\forall t > t_0$

[3] A stricter condition is the detailed balance condition(**reversibility**): for $\forall x, y \in \mathcal{X}$, $\pi(x)P(x,y) = \pi(y)P(y,x)$

Langevin Dynamics I

Langevin dynamics can produce samples from a probability density $\pi(x)$ using only the score function $\nabla_x \log \pi(x) = -\nabla_x U(x)$. Given a fixed step size $\epsilon > 0$, and an initial value $X_0 \sim \pi_0(x)$ with $\pi_0(x)$ being any prior distribution, the Langevin dynamics recursively compute the following noisy gradient descent step for $t \in [T]$:

$$X_t = X_{t-1} + \frac{\epsilon}{2} \nabla_x \log p\left(X_{t-1}\right) + \sqrt{\epsilon} Z_t \tag{2}$$

where $Z_t \sim \mathcal{N}(0, I_d)$. The distribution of X_T equals $\pi(x)$ when $\epsilon \to 0$ and $T \to \infty$, in which case X_T becomes an exact sample from $\pi(x)$ under some regularity conditions [WT11]. When $\epsilon > 0$ and $T < \infty$, a Metropolis-Hastings update is needed to correct the error of (2), but it can often be ignored in practice [CFG14] [DM19] [NHH$^+$19]. The error is negligible when ϵ is small and T is large.

Langevin Dynamics II

Algorithm 1 (Unadjusted) Langevin Dynamics

Require: Target density $\pi(x) = e^{-U(x)}$; initial state $X^{(0)}$; step size ϵ; number of iterations N.

1: **for** $i = 1, \ldots, N$ **do**

2: Compute the gradient $\nabla U(X^{(i-1)})$.

3: Sample $Z^{(i)} \sim \mathcal{N}(0, I_d)$

4: Update the state:

$$X^{(i)} \leftarrow X^{(i-1)} - \frac{\epsilon}{2}\nabla U(X^{(i-1)}) + \sqrt{\epsilon}\, Z^{(i)}.$$

5: **end for**

6: **return** $\{X^{(i)}\}_{i=0}^{N}$

Slow!

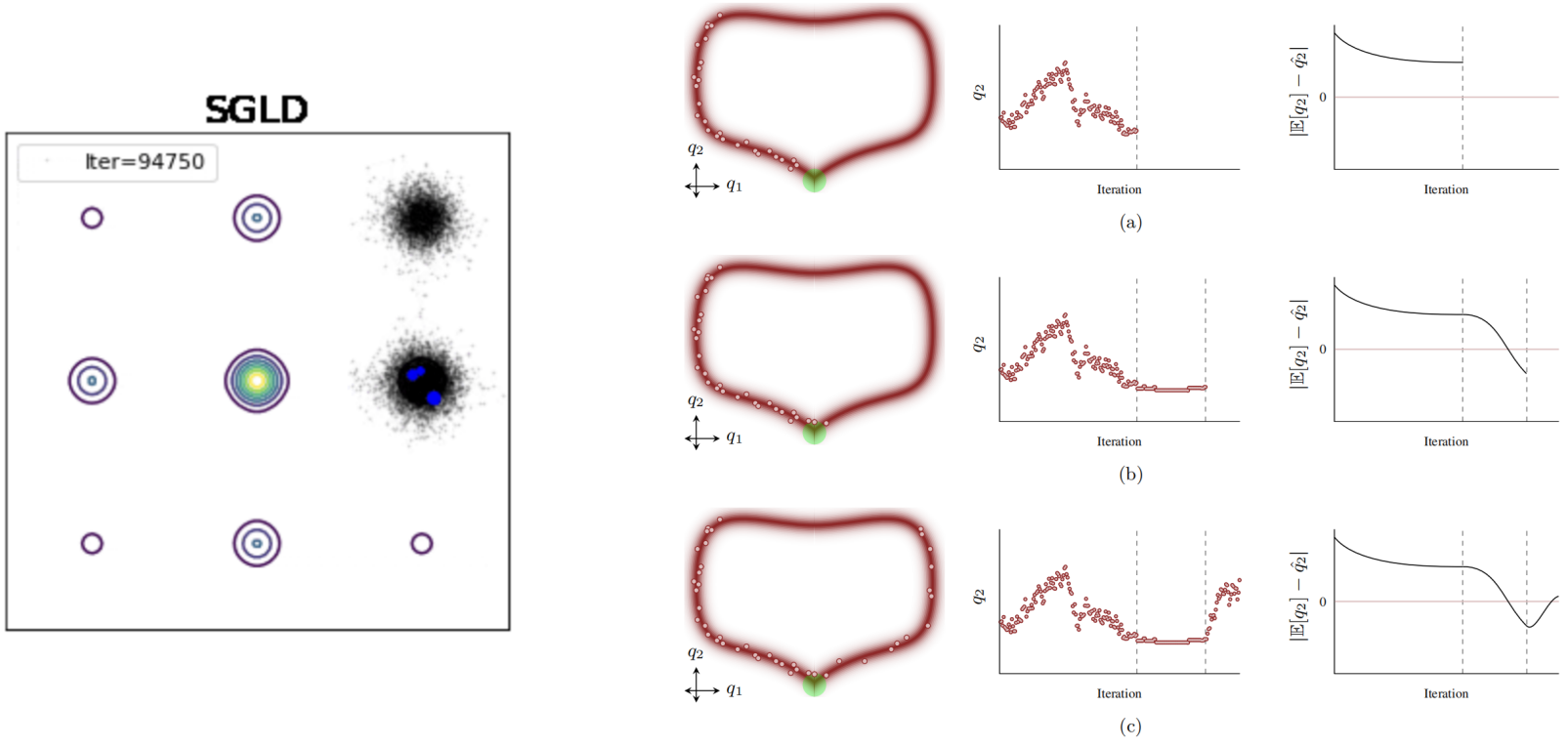


Figure 3: (Stochastic Gradient) Langevin Dynamics [GitHub] [Bet18]

Challenge: Slow!
Tentative solution: Extend local behavior to global behavior

Faster!

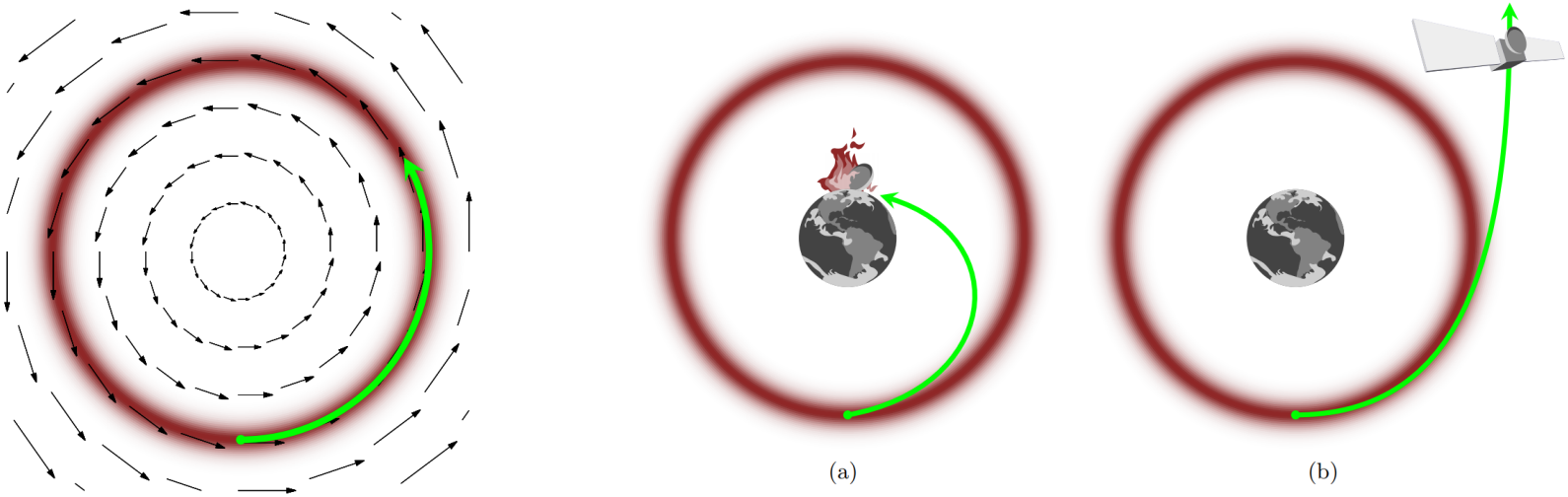


Figure 4: A vector field is the assignment of a direction at every point in parameter space. When those directions are aligned with the typical set we can follow them like guide posts, generating coherent exploration of the target distribution. We need to launch a satellite that orbits correctly, not wanting it to crash. [Bet18]

Outline

Hamiltonian dynamics I

The Hamiltonian Monte Carlo algorithm augments the original position variable with an auxiliary momentum variable. If $x \in \mathbb{R}^{\widehat{d}}$ denotes the **position** (with target distribution $\pi(x)$), HMC introduces a **momentum** $p \in \mathbb{R}^{\widetilde{d}}$. Let $d = \widehat{d} + \widetilde{d}$. Typically, a **Hamiltonian** $H(x, p) : \mathbb{R}^d \to \mathbb{R}$ is then defined as the sum of a **potential energy** $U(x) : \mathbb{R}^{\widehat{d}} \to \mathbb{R}$ and **kinetic energy** $K(p) : \mathbb{R}^{\widetilde{d}} \to \mathbb{R}$:

$$H(x, p) = U(x) + K(p).$$

Canonically, $U(x)$ is deterministically chosen as minus the log target density, so that $U(x) = -\log \pi(x)$. The kinetic energy $K(p)$ is flexible in theory while usually taken to be quadratic practically as $K(p) = \frac{1}{2} p^T M^{-1} p$, where M is a mass matrix (often set to the identity or diagonal). This choice corresponds to p being assigned a Gaussian distribution $\mathcal{N}(0, M)$ as its marginal. The introduction of p thus defines an extended target distribution $\widetilde{\pi}(x, p) \propto \exp[-H(x, p)] = \exp[-U(x) - K(p)]$ on $\mathbb{R}^d$, whose x-marginal is the original $\pi(x)$ by construction.

Hamiltonian dynamics II

Hamilton's equations are given by the coupled ordinary differential equations:

$$\frac{dx}{dt} = \nabla_p H(x,p) = \nabla_p K(p), \quad \frac{dp}{dt} = -\nabla_x H(x,p) = -\nabla_x U(x), \tag{3}$$

In particular for our choice, we have $\nabla_p K(p) = M^{-1}p$, and $\nabla_x U(x) = -\nabla_x \log \pi(x)$. These equations describe how (x,p) would evolve in an artificial "physics" system with Hamiltonian H. [4]

Some good properties the generator of this Hamiltonian ODE system (3) has:

- **invariant** w.r.t. $\widetilde{\pi}_H \propto e^{-H}$ (further, **skew-reversible**)
- **time-symmetrical**
- **Hamiltonian-conserving** (CAVEAT: not irreducible!)

[4] See a more delicate interpretation in our report!

Safe and sound

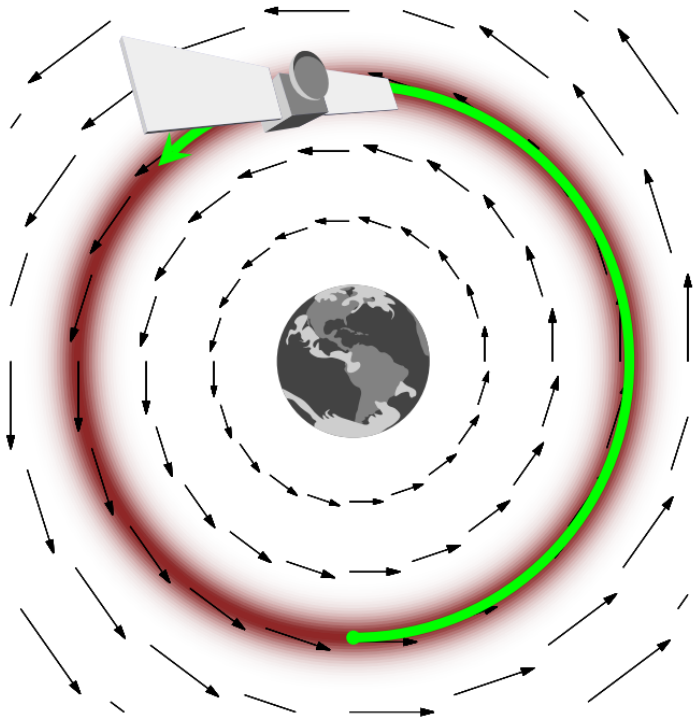


Figure 5: When we introduce exactly the right amount of momentum to the physical system, the equations describing the evolution of the satellite define a vector field aligned with the orbit. The subsequent evolution of the system will then trace out orbital trajectories. [Bet18]

Hamiltonian dynamics III

Question: How to impose the irreducibility?
Answer: Let's explore different level sets!

We sample a new momentum p from its Gaussian distribution independent of the current state. For example, one can draw $p \sim \mathcal{N}(0, M)$, so each component p_i is drawn from $\mathcal{N}(0, m_i)$ if $M = \text{diag}(m_1, \ldots, m_d)$.

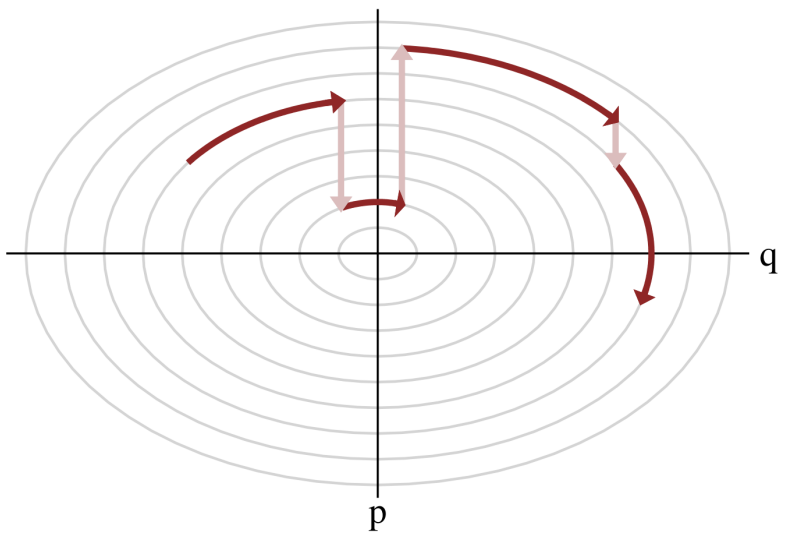


Figure 6: Explore different level sets (q is x) [Bet18]

Hamiltonian dynamics IV

Question: How to run the Hamiltonian system (3) in practice?
Answer: Let's use an integrator!

Euler's method

Choice 1: Euler's method

$$p_i(t+\varepsilon) = p_i(t) + \varepsilon \frac{dp_i}{dt}(t) = p_i(t) - \varepsilon \frac{\partial U}{\partial x_i}(x(t)),$$

$$x_i(t+\varepsilon) = x_i(t) + \varepsilon \frac{dx_i}{dt}(t) = x_i(t) + \varepsilon \frac{p_i(t)}{m_i}.$$

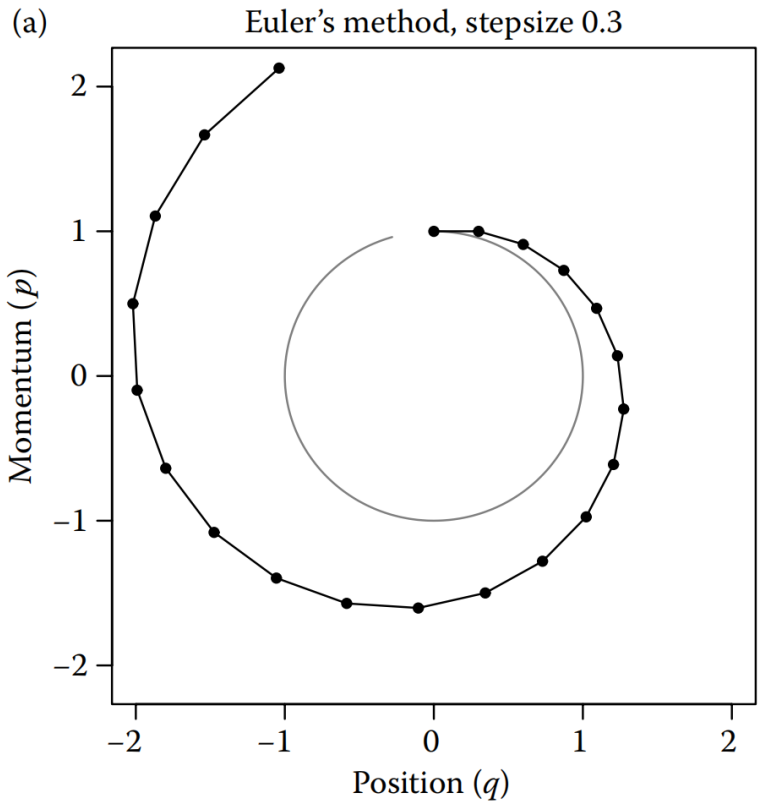


Figure 7: Euler's method initialized at $(x, p) = (0, 1)$, with $H(x, p) = x^2/2 + p^2/2$ and step size $\varepsilon = 0.3$ [Nea12]

Modified Euler's method

Choice 2: Modified Euler's method

$$p_i(t+\varepsilon) = p_i(t) - \varepsilon \frac{\partial U}{\partial x_i}(x(t)),$$

$$x_i(t+\varepsilon) = x_i(t) + \varepsilon \frac{p_i(t+\varepsilon)}{m_i}.$$

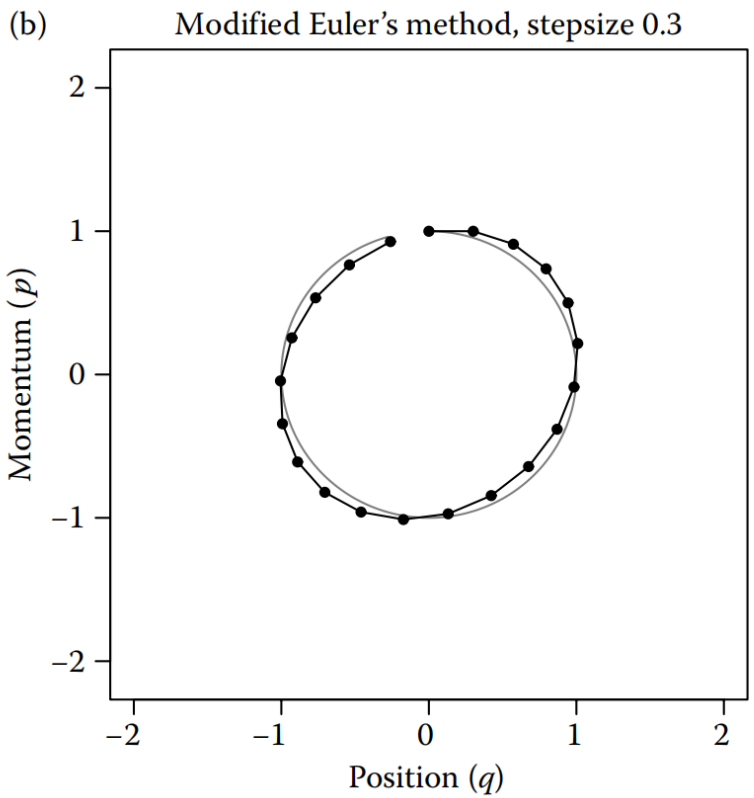


Figure 8: Modified Euler's method initialized at $(x, p) = (0, 1)$, with $H(x, p) = x^2/2 + p^2/2$ and step size $\varepsilon = 0.3$ [Nea12]

Leapfrog/Velocity Verlet method

Choice 3: Leapfrog/Velocity Verlet method

$$p_i(t+\varepsilon/2) = p_i(t) - (\varepsilon/2)\frac{\partial U}{\partial x_i}(x(t)),$$

$$x_i(t+\varepsilon) = x_i(t) + \varepsilon\frac{p_i(t+\varepsilon/2)}{m_i},$$

$$p_i(t+\varepsilon) = p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial x_i}(x(t+\varepsilon)).$$

In practice, we like leapfrog method a lot, we have tested different integrator's performance in the Numerical Experiments section.

Symplectic integrator

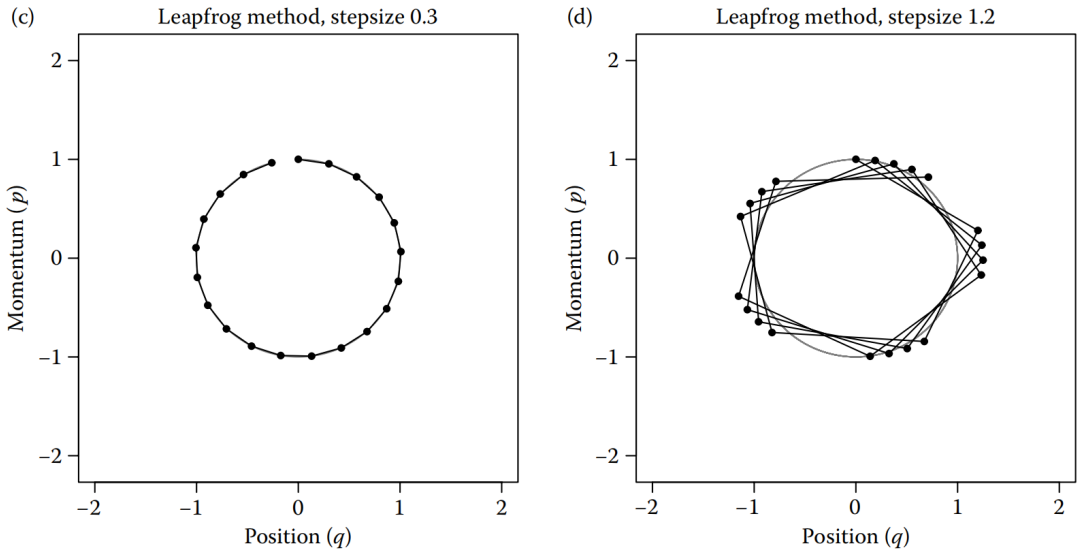


Figure 9: Leapfrog method initialized at $(x, p) = (0, 1)$, with $H(x, p) = x^2/2 + p^2/2$ and step size $\varepsilon = 0.3$ for (c), step size $\varepsilon = 1.2$ for (d) [Nea12]

In fact, **time-reversibility** and **volume-preserving** property of symplectic integrators are all we need. See in our report showing that Leapfrog is one of them. There are also other alternatives that are discussed in [HLW06].

Hamiltonian dynamics V

Question: (Optionally) How do we mitigate the discretization error?
Answer: Let's use a Metropolis acceptance step!

Metropolis step

We accept the proposed state with probability

$$\alpha = \min\{1, \exp[-H(x^*, p^*) + H(x, p)]\} = \min\{1, \exp(-\Delta H)\},$$

where $\Delta H = H(x^*, p^*) - H(x, p)$ is the change in the Hamiltonian along the numerically simulated trajectory. If energy were exactly conserved ($\Delta H = 0$) as in the continuous limit, the acceptance probability would be always 1. If the move is rejected, the state remains at (x, p). If the proposal is accepted, we take the new position x^* as the next sample and set $p^* \leftarrow -p^*$ as a negation step. This step is crucial in the theoretical sense to keep the proposal distribution symmetric in HMC (thus does not appear in the acceptance ratio), while it can be simply neglected because the kinetic energy we use is often even and we will resample the momentum next round.
See more details in our report!

Pseudocode: All in one

Algorithm 2 Hamiltonian Monte Carlo with Metropolis

Require: initial position $x^{(1)}$, step size ϵ, mass matrix M, # of leapfrog steps L, sample size N

for $t = 1, 2, \ldots, N$ **do**
  Sample momentum: $p^{(t)} \sim \mathcal{N}(0, M)$
  Set $(x_0, p_0) \leftarrow (x^{(t)}, p^{(t)})$
  $p_0 \leftarrow p_0 - \frac{\epsilon}{2} \nabla U(x_0)$
  for $i = 1$ **to** s **do**
    $x_i \leftarrow x_{i-1} + \epsilon M^{-1} p_{i-1}$
    $p_i \leftarrow p_{i-1} - \epsilon \nabla U(x_i)$
  end for
  $p_L \leftarrow p_L - \frac{\epsilon}{2} \nabla U(x_L)$
  Set proposal $(x^*, p^*) \leftarrow (x_L, -p_L)$ {A negation step that can be often neglected}
  Draw $u \sim \mathrm{Uniform}(0, 1)$
  Compute $\rho = \exp\big(H(x^*, p^*) - H(x^{(t)}, p^{(t)})\big)$
  if $u < \min(1, \rho)$ **then**
    $x^{(t+1)} \leftarrow x^*$
  else
    $x^{(t+1)} \leftarrow x^{(t)}$
  end if
end for
return $\{x^{(t)}\}_{t=1}^{N}$

Last conceptual sight

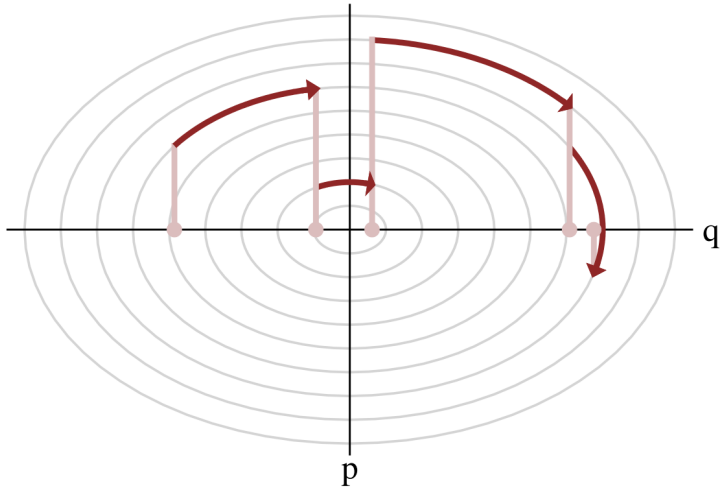
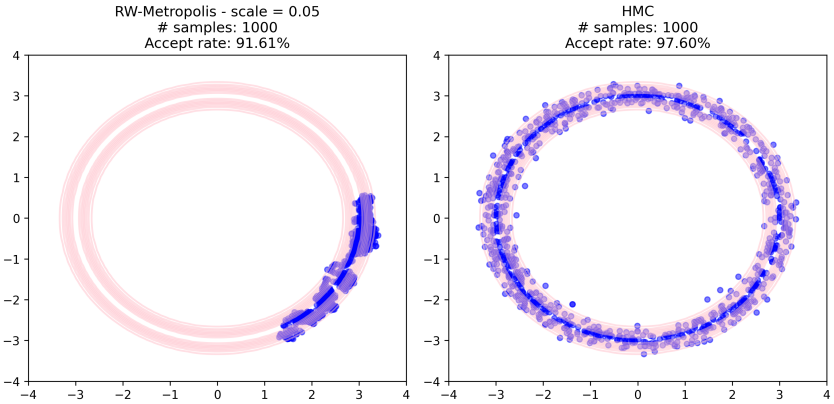


Figure 10: Each Hamiltonian Markov transition lifts the initial state onto a random level set of the Hamiltonian, which can then be explored with a Hamiltonian trajectory before projecting back down to the target parameter space. [Bet18]

Outline

1. RWM vs HMC on 2D Donut



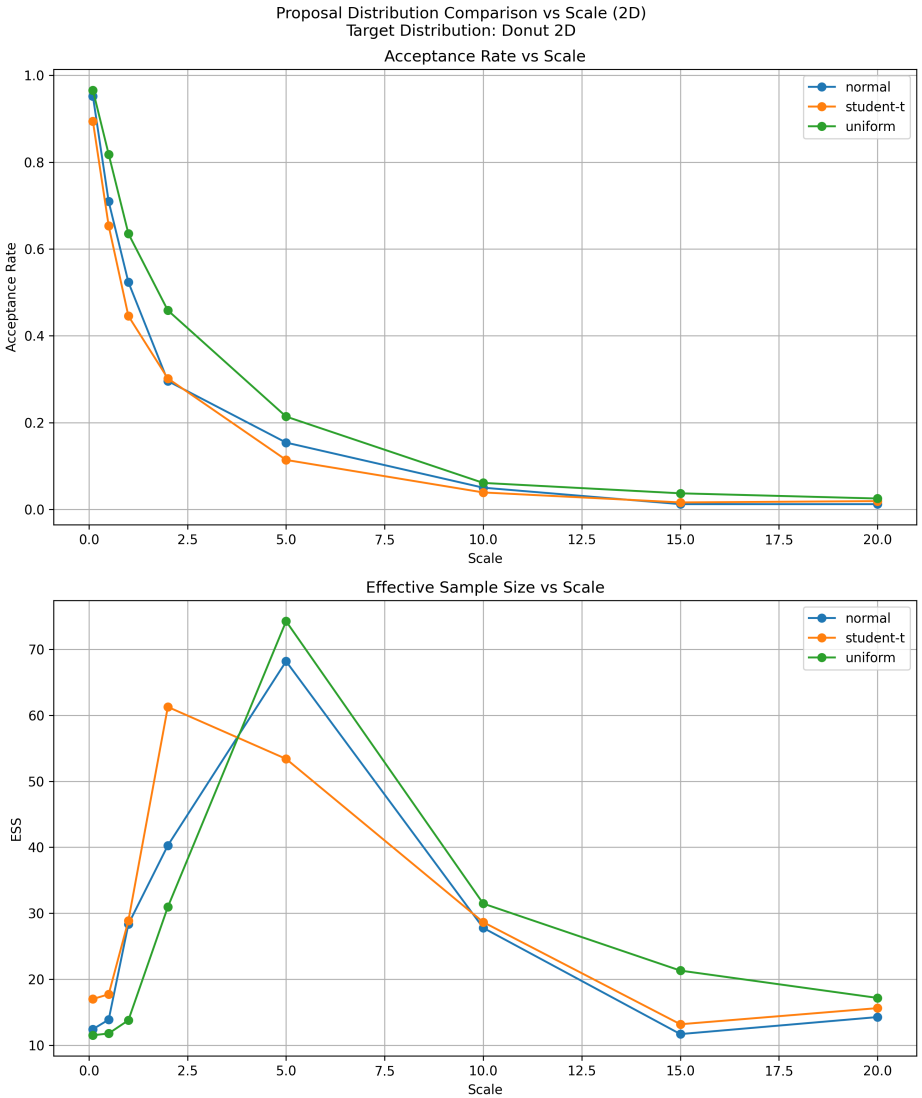
HMC vs RWM: 2D Donut Samples

Parameter	Value / Description
Dimension (d)	Configurable, default: 2
Number of Samples	1000
Donut Radius (r)	3.0
Shell Thickness (σ^2)	0.05
Initial State	$[r, 0, \ldots, 0]$
Proposal Type (Metropolis)	Gaussian (NormalProposal)
Metropolis Scales	0.05 (small), 1.0 (large)
HMC Step Size (ϵ)	0.1
HMC Leapfrog Steps (L)	50
Output File	`donut_comparison_{dim}d.png`
Visualization	2D scatter plots / pairwise projections
Target Distribution	Donut: $\exp\left(-\frac{(\|x\|-r)^2}{\sigma^2}\right)$

Experiment Summary:

- **Objective:** Show that HMC performs better than MCMC on donut distribution, which motivates our further experiments.
- **Conclusion**: HMC captures the ring structure significantly better and is more efficient at high curvature regions.

2.Metropolis Proposal Comparison vs Scale



Target Distribution: 2D Donut Distribution

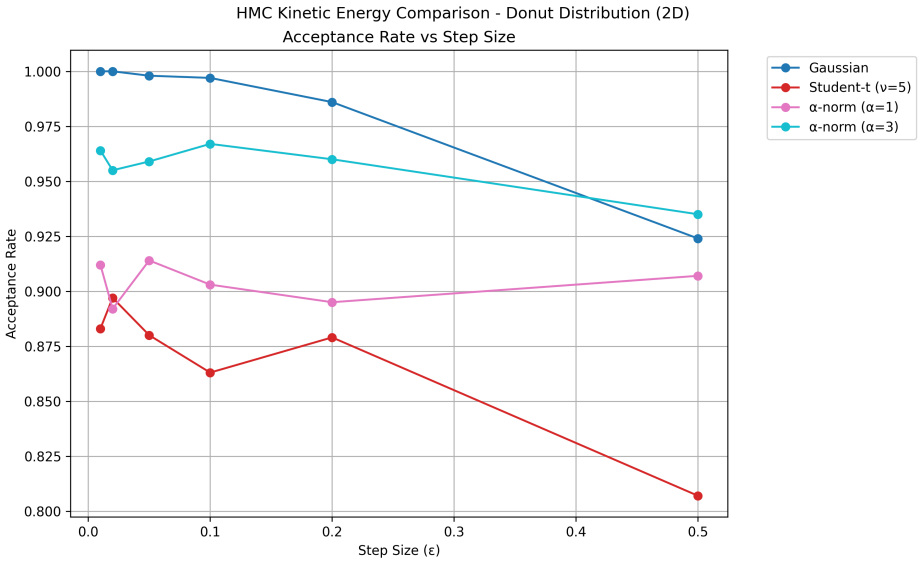
Experiment Settings

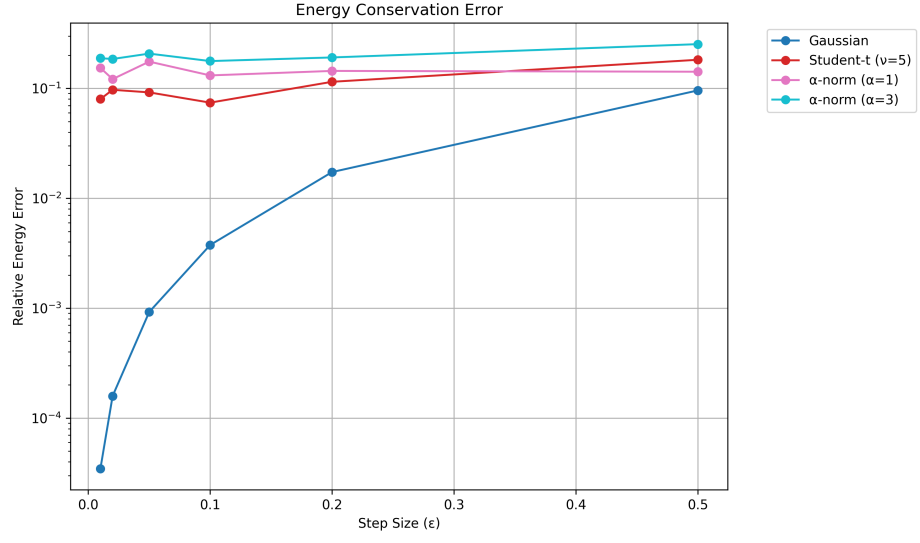
Parameter	Value
Target Distribution	Donut Distribution
Target Dimension	2D
Donut Radius	3.0
Donut Thickness Variance (σ^2)	0.5
Number of Samples	1,000
Initial State	$(3.0, 0.0)$
Proposal Types	Gaussian, Student-t (df=3), Uniform
Proposal Scales	0.1, 0.5, 1.0, 2.0, 5.0, 10, 15, 20
Sampler	MCMC with different proposals
Acceptance Rate Estimation	Accepted samples / Total samples(1000)
Evaluation Metrics	Acceptance Rate, Effective Sample Size (ESS)

Experiment Summary

- **Objective:** Show that larger scale (variance) in proposal distribution leads to lower acceptance rates. And proper scale could improve ESS results..
- **Conclusion**: Larger scales or smaller scales easily fall outside of the high-density region (ring) of the donut distribution. Generally MCMC performs bad on donut distribution.

3. HMC Kinetic Energy Comparison vs Step Size





Target Distribution: 2D Donut Distribution

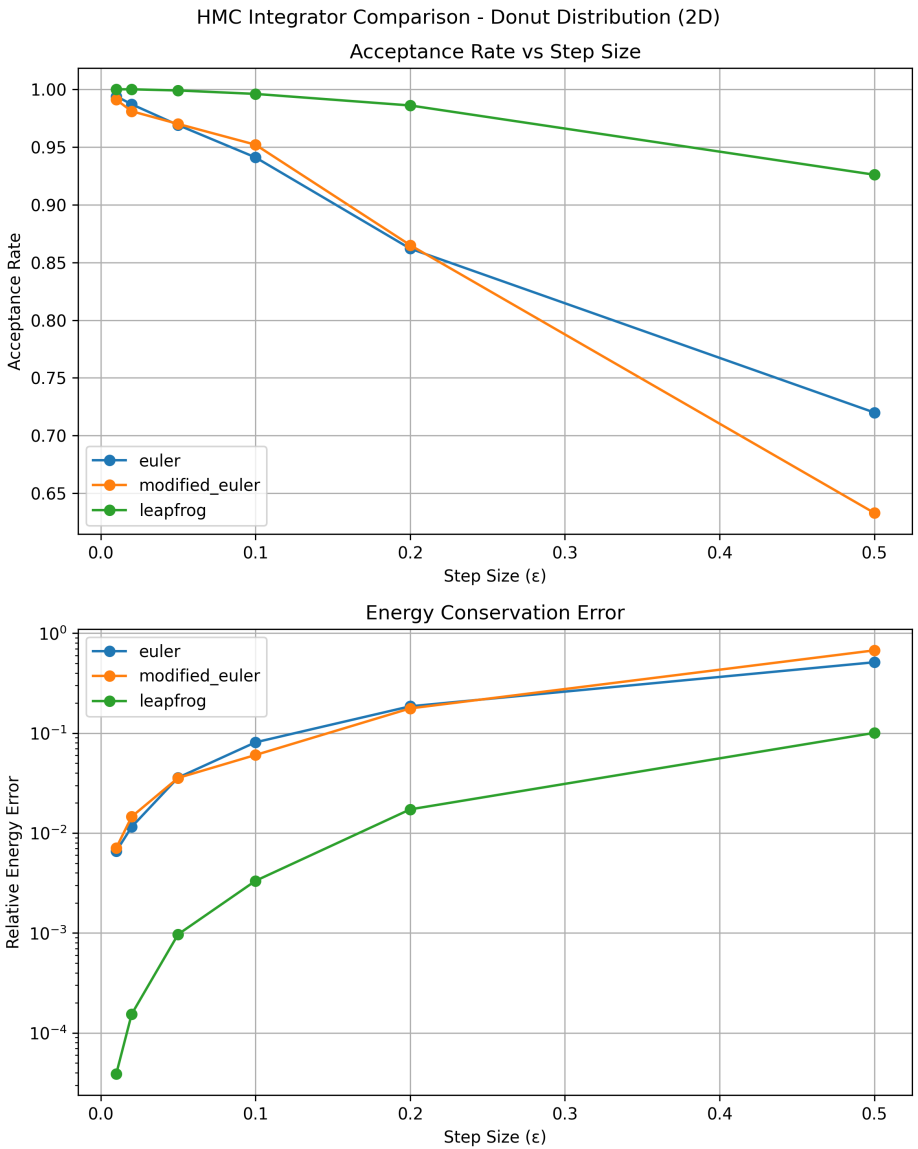
General Settings

Parameter	Value
Target Distribution	Donut Distribution
Target Dimension	2D
Number of Samples	1000
Donut Thickness Variance (σ^2)	0.5
Initial State	$(3.0, 0.0)$
Trajectory Length ($s = \epsilon \times L$)	0.5
Step Sizes (ϵ)	0.01, 0.02, 0.05, 0.1, 0.2, 0.5
Kinetic Energy Types	Gaussian, Student-t ($\nu = 5$), ℓ_1-norm ($\alpha = 1$), ℓ_3-norm ($\alpha = 3$)
Integrator	Leapfrog (fixed)
Sampler	HMC with varying kinetic energy
Acceptance Rate Estimation	Accepted samples / Total samples
Evaluation Metrics	Acceptance Rate, Relative Energy Error

Experiment Summary

- **Objective:** Compare HMC's performance on different kinetic energy vs. different step size(ϵ)
- **Conclusion**: Gaussian kinetic is the most used and most robust energy function for 2D donut distribution, and all of them perform better than the MCMC sampler.

4. HMC Integrator Comparison vs Step Size



Target Distribution: 2D Donut Distribution

General Settings

Parameter	Value
Target Distribution	Donut Distribution
Target Dimension	2D
Donut Radius	3.0
Donut Thickness Variance (σ^2)	0.5
Number of Samples	1000
Initial State	$(3.0, 0.0)$
Trajectory Length ($s = \epsilon \times L$)	0.5
Step Sizes (ϵ)	0.01, 0.02, 0.05, 0.1, 0.2, 0.5
Integrators	Euler, Modified Euler, Leapfrog
Sampler	HMC with different integrators
Acceptance Rate Estimation	Accepted samples / Total samples
Evaluation Metrics	Acceptance Rate, Relative Energy Error

Experiment Summary

- **Objective:** Compare HMC's performance on different integrators vs. different step size(ϵ)
- **Conclusion**: Leapfrog integrator performs the better, the acceptance rate is consistently higher than MCMC with different proposals.

5. MCMC vs HMC w.r.t. Dimension Experiment Settings

Category	Configuration
General Settings	
Number of Points to Sample	1000
Warmup Samples	100
Dimensions Tested	[2, 5, 10, 20, 50, 100, 200, 500]
Target Distributions	
Standard Gaussian	
Type	Multivariate Normal
Mean	Zero vector
Covariance	Identity matrix
Donut Distribution	
Type	N-dimensional shell
Radius	3.0
Donut Thickness Variance(σ^2)	0.5
MCMC Settings — Proposal Distributions	
Gaussian	Covariance = Identity matrix
Student-t	Degrees of freedom = 3
Uniform	Range = [-1, 1]
HMC Settings	
Leapfrog Integrator	
Trajectory Length ($s = \epsilon$ x L)	0.5
Step Size (ϵ)	0.1
Kinetic Energy Distributions	
Gaussian	Standard normal momentum
Student-t	Degrees of freedom (ν) = 3.0

Compare HMC variants with MCMC variants under different dimensions

Objective: Compare performance of different samplers under varying dimensions.

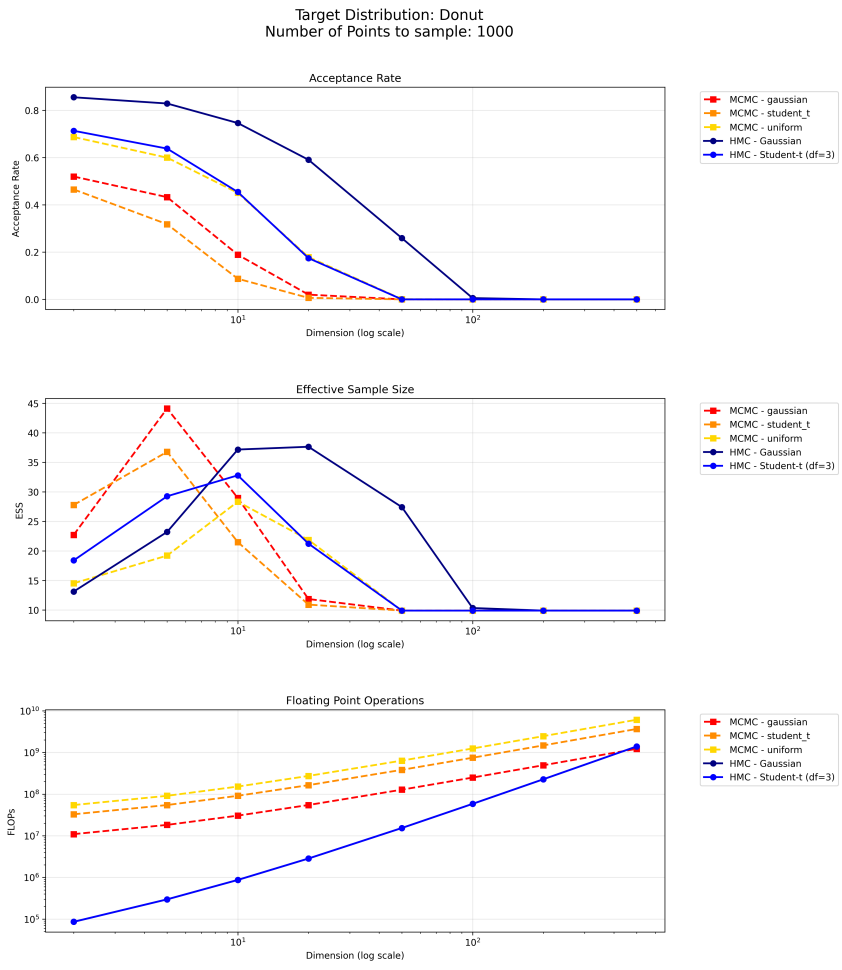
Metrics Evaluated:

- Acceptance Rate
- Effective Sample Size (ESS)
- Number of float point operations

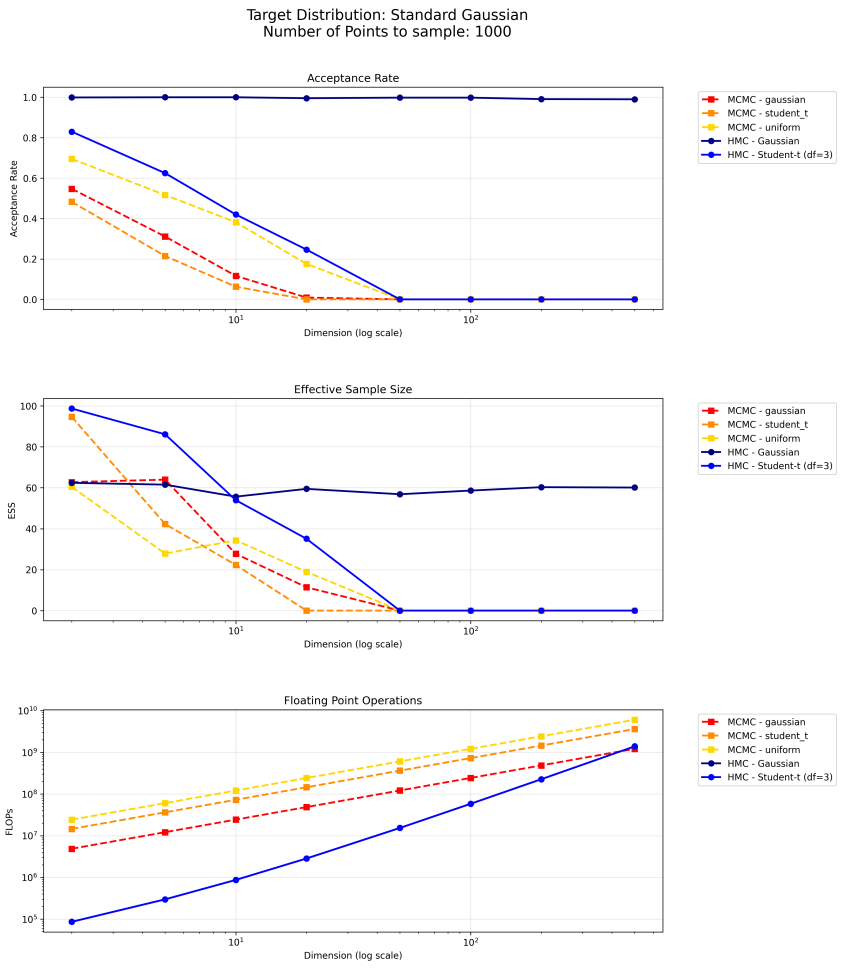
Comparisons:

- **MCMC**: different proposal distributions
- **HMC**: different kinetic energy formulations

5. MCMC vs HMC Across Dimensions Experiment Results



Donut Distribution



Gaussian Distribution

6. High-Dimension HMC Variants Experiment Settings

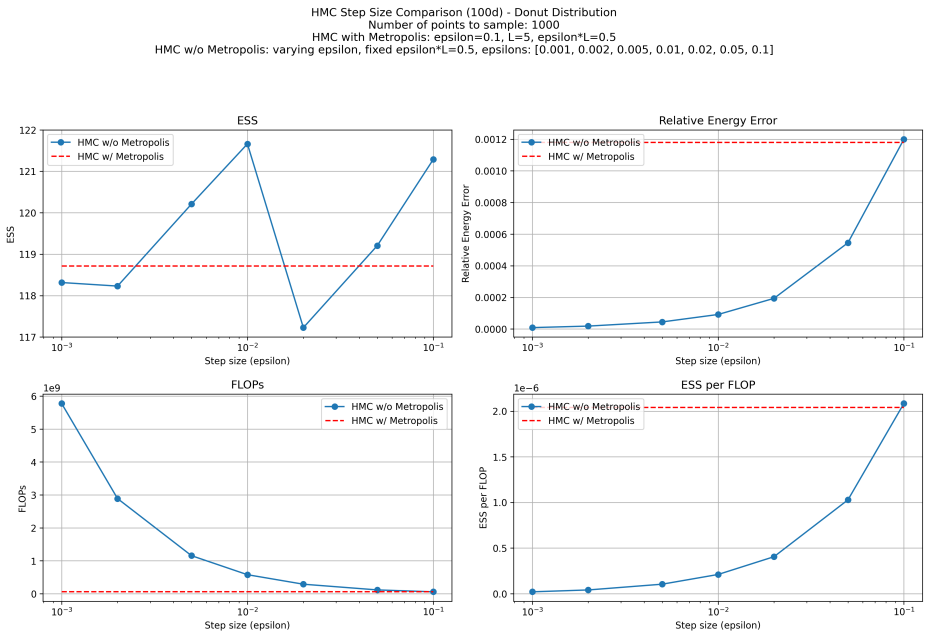
Objective:
In this experiment, we aim to verify the hypothesis that: The Metropolis-Hastings correction step may not be necessary in high-dimension settings, and decreasing step size may be more helpful. But we cannot ignore the fact that with Metropolis correction we can significantly decrease the FLOPs and reach similar level of ESS as without Metropolis correction.

Experiment Configuration:

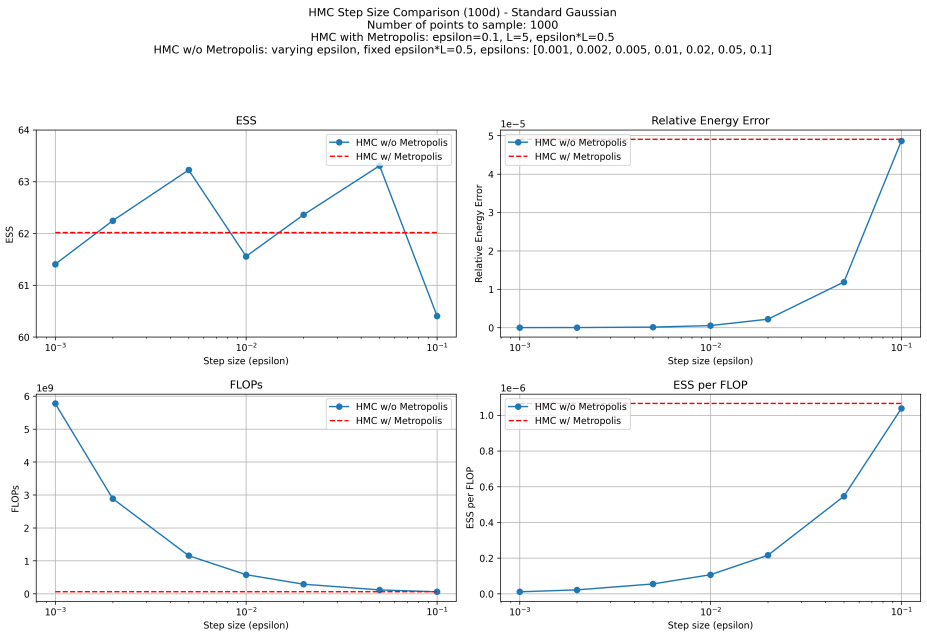
Parameter	Value
Target Distributions	Standard Gaussian, Donut
Donut Parameters	Radius = 3.0, Variance = 0.5
Dimensionality	100, 200, 300
Samples per Run	1000
Warmup Samples	100
Trajectory Length ($s=\epsilon$ x L)	0.5
Kinetic Energy Form	Gaussian
Step Size(ϵ) (with Metropolis)	0.1
Step Sizes(ϵ) (no Metropolis)	[0.002, 0.005, 0.01, 0.025, 0.04, 0.055, 0.07, 0.085, 0.1]
Metrics Measured	ESS, Computation Time, Relative Energy Error, ESS per second

6.High dimension HMC variants Experiment Results - 100d

Under 100d distribution we have:



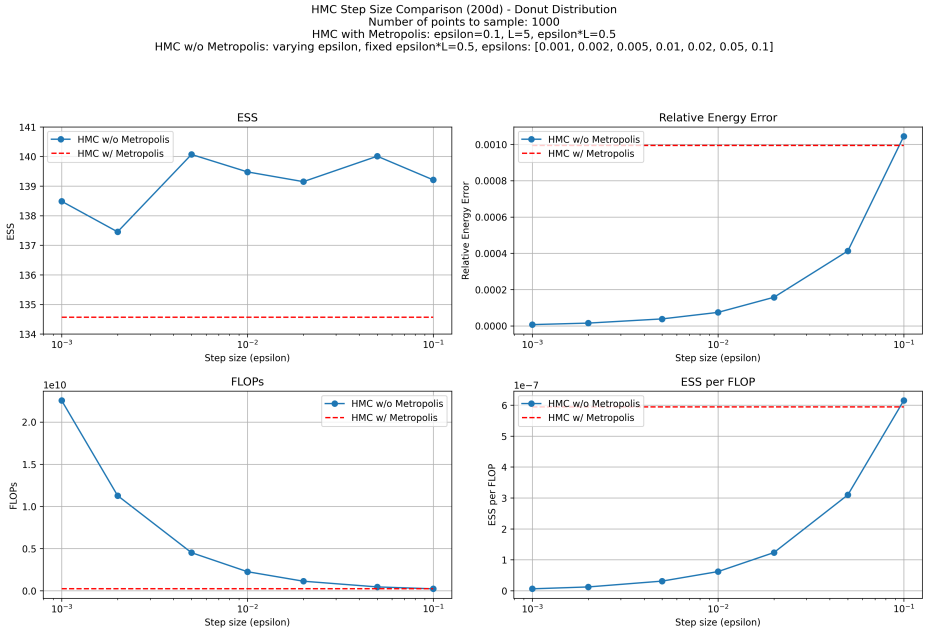
Donut Distribution



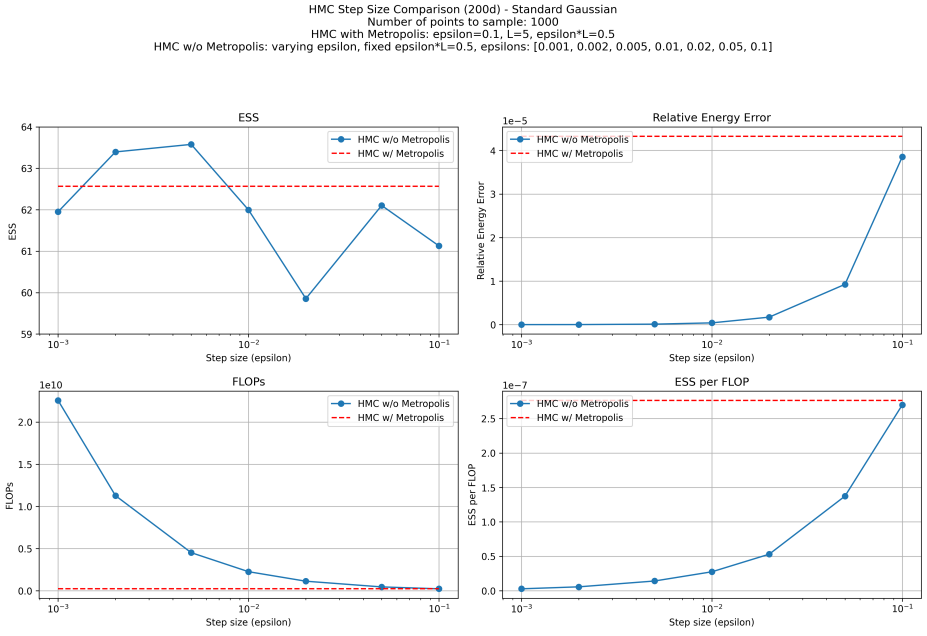
Gaussian Distribution

6.High dimension HMC variants Experiment Results - 200d

Under 200d distribution we have:



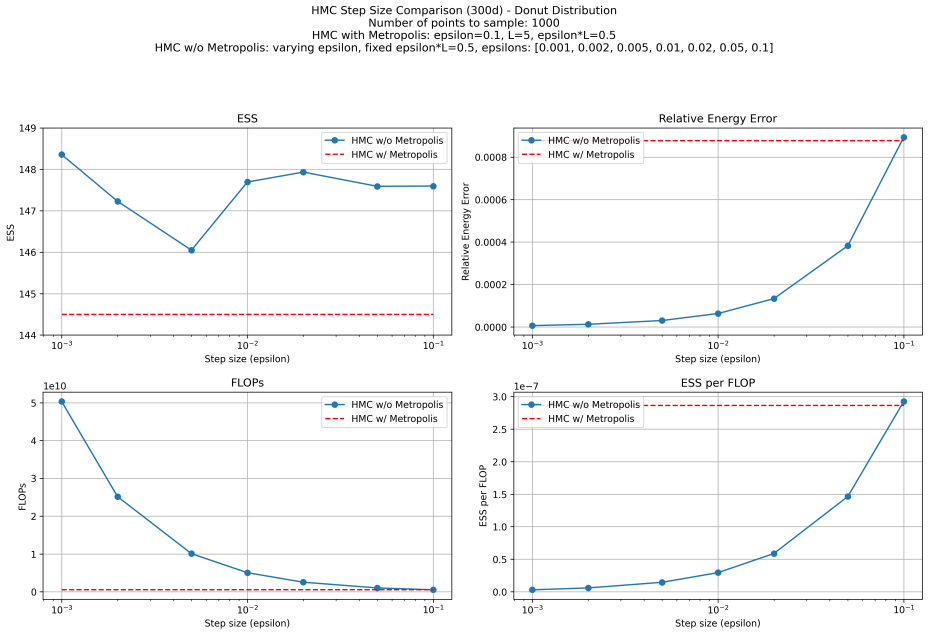
Donut Distribution



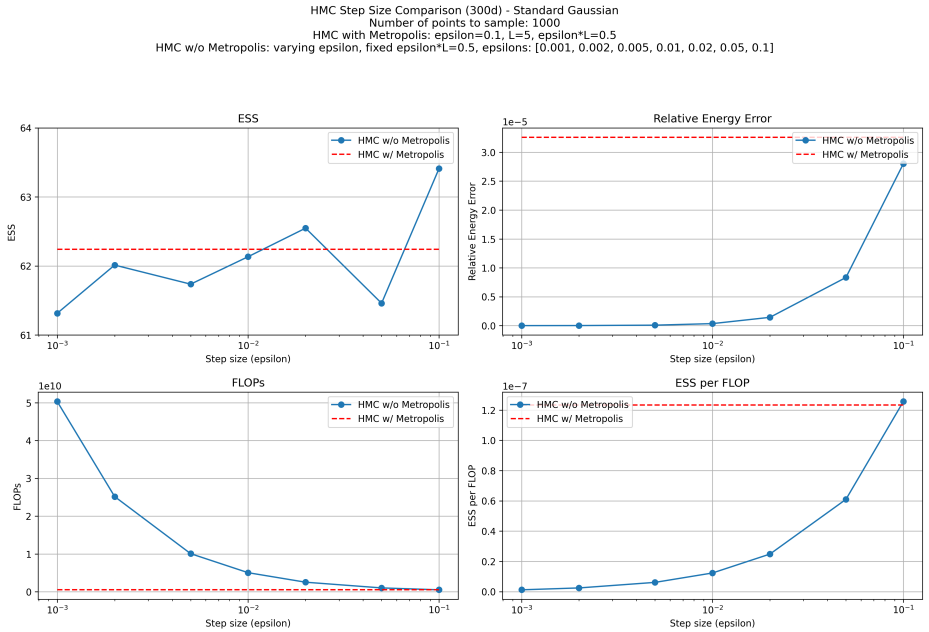
Gaussian Distribution

6. High dimension HMC variants Experiment Results - 300d

Under 300d distribution we have:



Donut Distribution



Gaussian Distribution

Outline


Michael Betancourt.
A conceptual introduction to hamiltonian monte carlo, 2018.


Tianqi Chen, Emily Fox, and Carlos Guestrin.
Stochastic gradient hamiltonian monte carlo.
In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1683–1691, Bejing, China, 22–24 Jun 2014. PMLR.


Yilun Du and Igor Mordatch.
Implicit generation and generalization in energy-based models.
ArXiv, abs/1903.08689, 2019.


Ernst Hairer, Christian Lubich, and Gerhard Wanner.
Geometric numerical integration. Structure-preserving algorithms for ordinary differential equations. 2nd ed, volume 31.
01 2006.


Radford Neal.

Mcmc using hamiltonian dynamics.
Handbook of Markov Chain Monte Carlo, 06 2012.

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu.
On the anatomy of mcmc-based maximum likelihood learning of energy-based models.
In *AAAI Conference on Artificial Intelligence*, 2019.

Max Welling and Yee Whye Teh.
Bayesian learning via stochastic gradient langevin dynamics.
In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 681–688, Madison, WI, USA, 2011. Omnipress.