Magnetic HMC with Dual Averaging (The Knight and Bishop Algorithm)

Zehaan Naik

April 15, 2025

1 Introduction

Hamiltonian Monte Carlo (HMC) (Neal et al., 2011) is a popular gradient-driven MCMC (Markov Chain Monte Carlo) algorithm that is used to generate samples efficiently from several classes of targets. HMC uses Hamiltonian dynamics to generate guided proposals and has characteristically high acceptance rates. As such, HMC struggles to explore the domain of the target as the sampler gets "stuck" near the modes of the target due to the high energy barriers. Due to its localized nature of sampling, HMC struggles to sample from multimodal targets efficiently. I illustrate this problem by visualizing a typical trajectory of an HMC sampler in Fig. 1.

Solutions to multimodal sampling using HMC are discussed in many articles such as Vishwanath and Tak (2024) (Repelling-Attracting HMC), Bornn et al. (2010) (Reimann Manifold Langevin HMC). However, this article primarily focuses on Tripuraneni et al. (2017) (Magnetic HMC). It attempts to generalize the HMC paradigm for linearly related Hamiltonian equations (a visual representation of the difference in trajectories is also shown in Fig. 1). Most approaches to "fix" HMC introduce new physical concepts that connect the position and momentum of the particle in question and, in the process, introduce new hyperparameters to the sampler. These hyperparameters often lack an intuitive contribution to the algorithm's performance and become rather difficult to tune.

In this article, I aim to implement a problem-agnostic Magnetic HMC (MHMC) version that automatically tunes the algorithm's hyperparameters without prior information. I take inspiration from Hoffman et al. (2014) (No U-Turn Sampler) and their implementation of dual averaging (Nesterov, 2009) to tune model parameters. The overall philosophy for tuning the parameters is to use the domain exploration routine described in Hoffman et al. (2014) to tune the optimal choice for L (the number of steps taken by the leapfrog), then use the dual averaging paradigm to tune the remaining hyperparameters.

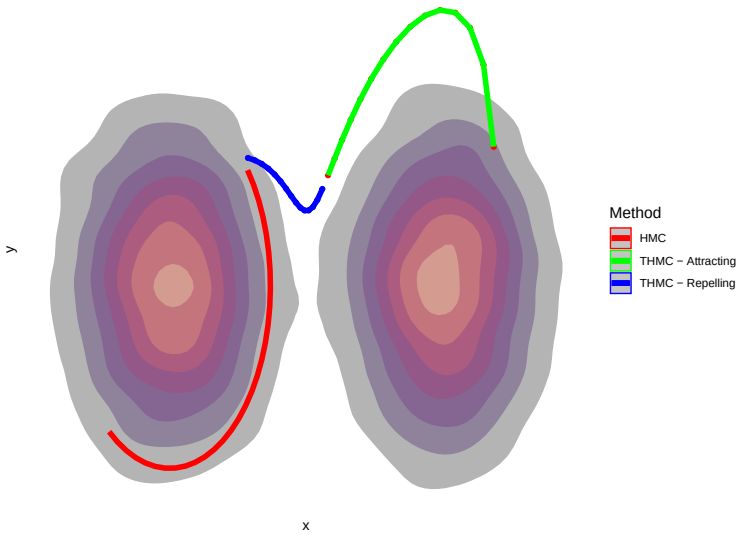

Figure 1: Comparing the HMC trajectory with another non-canonical sampler (Tempered HMC) to illustrate the problem of multimodal sampling. We see the target (mixture of two Gaussians - x-axis) augmented with the momentum distribution (standard Gaussian - y-axis)

.

2 Background

In this section, I discuss the basics of Hamiltonian Monte Carlo (HMC) and other relevant topics in physics that are used to derive the proposed algorithm. This section assumes basic knowledge of MCMC and, in many places, provides the heuristics required to understand the proposed implementation with appropriate references that give a detailed explanation of the concepts.

2.1 Hamiltonian Monte Carlo (HMC)

In Neal et al. (2011), we see the use of Hamiltonian Dynamics to sample proposals of the form $\pi(\mathbf{x}) \propto \exp\{-U(\mathbf{x})\}$, where $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$. We augment the target distribution with an auxiliary momentum component, $\mathbf{p} = (p_1, \ldots, p_d) \in \mathbb{R}^d$, with the marginal density $\pi_p(\mathbf{p}) \propto \exp(K(\mathbf{p}))$, we get the joint target distribution given by:

$$\pi_H(\mathbf{x}, \mathbf{p}) = \exp\{-[U(\mathbf{x}) + K(\mathbf{p})]\}. \tag{1}$$

Here, draws from the target $\mathbf{x}$ are seen as the position of a particle in a frictionless, d-dimensional contour, and $\mathbf{p}$ as its current momentum. Hence, with an appropriate choice of K and U, the energy function $H = K(\mathbf{p}) + U(\mathbf{x})$ can be considered the Hamiltonian of the extended state-space $(\mathbf{x}, \mathbf{p}) \in \mathbb{R}^{2d}$.

Note. The functions U and K are referred to as the potential and Kinetic energy of the particle, respectively.

In practice, we set the distribution of the momentum to be one from which we can generate draws efficiently, such as the standard Gaussian for d dimensions. We generate a draw from this distribution and say that the new state $(\mathbf{x_0}, \mathbf{p_0})$) starts at time $t = 0$. Following this, we solve the Hamiltonian equations given by:

$$\frac{d\mathbf{p}}{dt} = -\nabla U(\mathbf{x}), \tag{2a}$$

$$\frac{d\mathbf{x}}{dt} = \nabla K(\mathbf{p}), \tag{2b}$$

to map out the state to a time point $t = s$, $(\mathbf{x_s}, \mathbf{p_s})$). Our proposed state at time point s is obtained using the momentum flip operator $F : (\mathbf{x}, \mathbf{p}) \to (\mathbf{x}, -\mathbf{p})$. Finally, the value is accepted or rejected using the Metropolis-Hastings acceptance probability:

$$\alpha((\mathbf{x_0}, \mathbf{p_0}), (\mathbf{x_s}, \mathbf{p_s})) = \min\left\{1, \exp\left\{-H(\mathbf{x_s}, \mathbf{p_s})) + H(\mathbf{x_0}, \mathbf{p_0}))\right\}\right\}. \tag{3}$$

Remark. The momentum flip operator F is crucial to ensure the time-reversibility of HMC and to prove its invariant properties theoretically. However, in most implementations, this step becomes redundant.

This methodology relies significantly on us being able to solve equation 2 exactly. In practice, this is often not possible. Hence, we implement the Leap Frog integrator to approximate the state's time update. For convenience of notation, we use $\mathbf{\Phi}_\epsilon$ to denote a time update of $dt = \epsilon$ to the state given by:

$$\mathbf{\Phi}_\epsilon(\mathbf{x}, \mathbf{p}) := \zeta_1 \circ \zeta_2 \circ \zeta_3(\mathbf{x}, \mathbf{p}), \tag{4}$$

where,

$$\zeta_1(\mathbf{x}, \mathbf{p}) = \zeta_3(\mathbf{x}, \mathbf{p}) = \left(\mathbf{x}, \mathbf{p} - \frac{\epsilon}{2}\nabla U(\mathbf{x})\right),$$
$$\zeta_2(\mathbf{x}, \mathbf{p}) = (\mathbf{x} + \epsilon * \mathbf{p}, \mathbf{p}).$$

And finally, the full time update denoted by $\mathbf{\Phi}_{\mathbf{L}\epsilon}$ is given by:

$$\mathbf{\Phi}_{\mathbf{L}\epsilon}(\mathbf{x}, \mathbf{p}) := (\mathbf{\Phi}_\epsilon)^L(\mathbf{x}, \mathbf{p}). \tag{5}$$

Note. The Hamiltonian equations can be solved analytically and, as such, have an acceptance rate of 1 due to the energy conservation property of the Hamiltonian. Examples of such implementations are found in literature such as Vats (2023).

The above discussion is summarised in the Algorithm. 1. I also provide a visualization of the samples generated using this algorithm for a bi-variate Gaussian target in Fig. 2. I show the samples on a "ground truth" contour plot to illustrate the sample distribution. I also compare the empirical densities of both the sample components with their theoretical counterparts to check how well HMC generates samples from a simple target.

Note. We can sample $\mathbf{p}$ from $\mathcal{N}_d(0, \Sigma)$ for a more generalized approach. However, we choose to

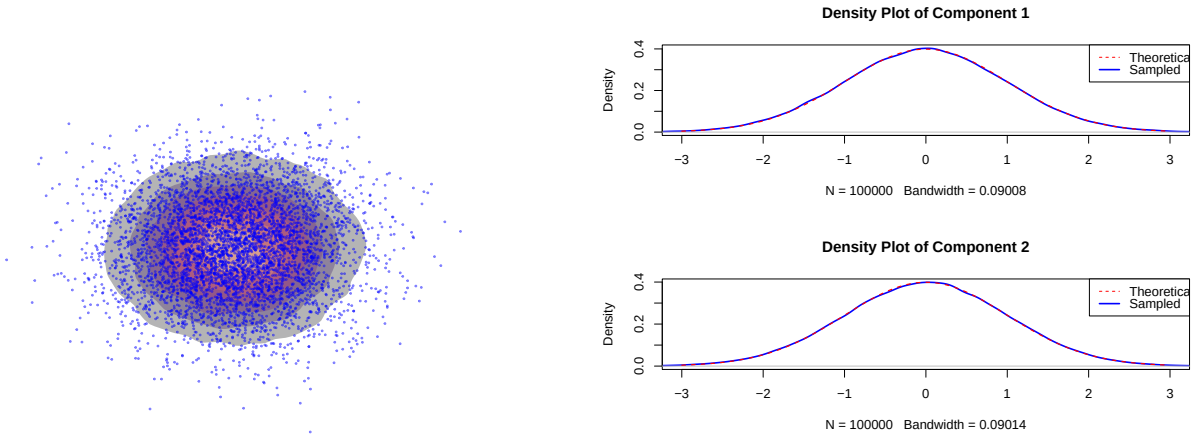

Figure 2: (left) Visualizing the samples generated using HMC for a bi-variate standard Gaussian target. (right) comparing the empirical density to the theoretical density of the target.

Algorithm 1 HMC using Leapfrog

Draw $\mathbf{p} \sim \mathcal{N}_d(0, I_d)$
Calculate $(\mathbf{x}^*, \mathbf{p}^*) = \mathbf{\Phi}_{\mathbf{L}\epsilon}(\mathbf{x}, \mathbf{p})$
Calculate the acceptance ratio:

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

if runif$(0, 1) < \alpha((\mathbf{x}, \mathbf{p}), (\mathbf{x}^*, \mathbf{p}^*))$ **then**
    $\mathbf{x}_{k+1} = \mathbf{x}^*$
else
    $\mathbf{x}_{k+1} = \mathbf{x}_k$
end if

keep the calculations simple through this article.

2.2 Key Properties of HMC

HMC is time-reversible and energy-conserving (completely in its exact form and approximately in the discretized set-up). Time-reversibility refers to the idea that we can reverse the time flow in our state update equations and reach our initial point in the state to the final point. This result ensures that our algorithm is invariant for our target by the result discussed in Tierney (1994). Energy conservation is a consequence of Hamiltonian Dynamics, and it helps HMC achieve its characteristic high acceptance rates. Hamiltonian dynamics also describe a volume-preserving flow; this helps us avoid the Jacobian term from the acceptance ratio. Finally, HMC describes a symplectic flow that allows us to implement numerical discretisers to approximate complicated targets. It is interesting to note that symplecticity implies volume preservation. A detailed description of these properties is provided in Neal et al. (2011). In this section, I discuss the other key properties of HMC: volume preservation and symplecticity.

2.3 Generalized Hamiltonian Monte Carlo

In Tripuraneni et al. (2017), we find a generalized adaptation of the Hamiltonian equations of the form:

$$\frac{d(\mathbf{x}, \mathbf{p})}{dt} = \mathbf{A}\nabla H(\mathbf{x}, \mathbf{p}). \tag{6}$$

Here, $\mathbf{A} \in \mathbb{R}^{2d\times 2d}$ is an invertible, anti-symmetric matrix. They also prove a few important results about the induced flow due to this updated relation that I state here without proving any.

Lemma 1. The map $\mathbf{\Phi_s^A}$ defined by solving the non-canonical Hamiltonian equations (Eq. 6) is both energy conserving and symplectic with respect to $\mathbf{A}$ ($(\nabla\mathbf{\Phi_H}(\mathbf{x}, \mathbf{p}))^T\mathbf{A}^{-1}\nabla\mathbf{\Phi_H}(\mathbf{x}, \mathbf{p}) = \mathbf{A}^{-1}$) which makes it volume preserving.

It is important to note that $\mathbf{A}$ takes the form:

$$\mathbf{A} = \begin{bmatrix} E & F \\ -F^T & G \end{bmatrix}_{2d\times 2d}, \tag{7}$$

where, E and G are anti-symmetric and F is taken to be general such that $\mathbf{A}$ is invertible. With these anti-symmetric properties, the next Lemma becomes more intuitive.

Lemma 2. If $(\mathbf{x_s}, \mathbf{p_s})$ is a solution to the non-canonical equations, then $(\mathbf{x_{-s}}, \mathbf{p_{-s}})$ is a solution to the modified equations, where $\mathbf{A} \to \tilde{\mathbf{A}}$, such that:

$$\tilde{\mathbf{A}} = \begin{bmatrix} -E & F \\ -F^T & -G \end{bmatrix}_{2d\times 2d},$$

if $H(\mathbf{x}, \mathbf{p}) = H(\mathbf{x}, -\mathbf{p})$. In particular, if $E = G = 0$, then $\mathbf{A} = \tilde{\mathbf{A}}$, which reduces to the traditional time-reversal symmetry of canonical Hamiltonian dynamics.

We notice that Lemma 2 poses a challenge in implementing the problem that would ensure detailed balance. Their solution to obtaining a time-reversible proposal is to flip the elements of the E and G matrices just as ordinary HMC flips the auxiliary variable $\mathbf{p}$, first, at the end of Hamiltonian flow in the proposal and, second, again after the MH acceptance step to return p to its original direction. This ensures that the algorithm is Time-reversible and, hence, target invariant.

Note. Heuristically, $F \in \mathbb{R}^{d\times d}$ can be seen as a matrix correlating the time dependence of potential and kinetic energy elements. We shall take $F = I_d$ throughout this article for ease of calculations. However, we can consider exceptional choices of F to better incorporate desired co-dependencies. We also take $E = 0$ as the only way to solve the equations. Heuristically, this means that the time increments of all the components of the target are independent of each other, which is a valid assumption to make.

As for their symplectic integrator, they only need to change ζ_2 in Eq. 4 to account for the new non-canonical system. ζ_2 changes to,

$$\zeta_2(\mathbf{x}, \mathbf{p}) = \left(\mathbf{x} + G^{-1}\left(\exp(G\epsilon) - I\right)\mathbf{p}, \exp(G\epsilon)\mathbf{p}\right). \tag{8}$$

Hence, the final leapfrog integrator for this algorithm looks like this:

$$\mathbf{\Phi}_{\mathbf{L}\epsilon}(\mathbf{x}, \mathbf{p}) := F_p \circ F_G \circ (\zeta_1 \circ \zeta_2 \circ \zeta_3)^L(\mathbf{x}, \mathbf{p}). \tag{9}$$

The above discussion is summarised in the Algorithm. 2. I also supplement the algorithm with a visualization of the samples and a sample path that the algorithm takes to arrive at a new sample in Fig. 3. I illustrate the samples generated from the MHMC algorithm and compare both components' empirical densities with their theoretical target density.

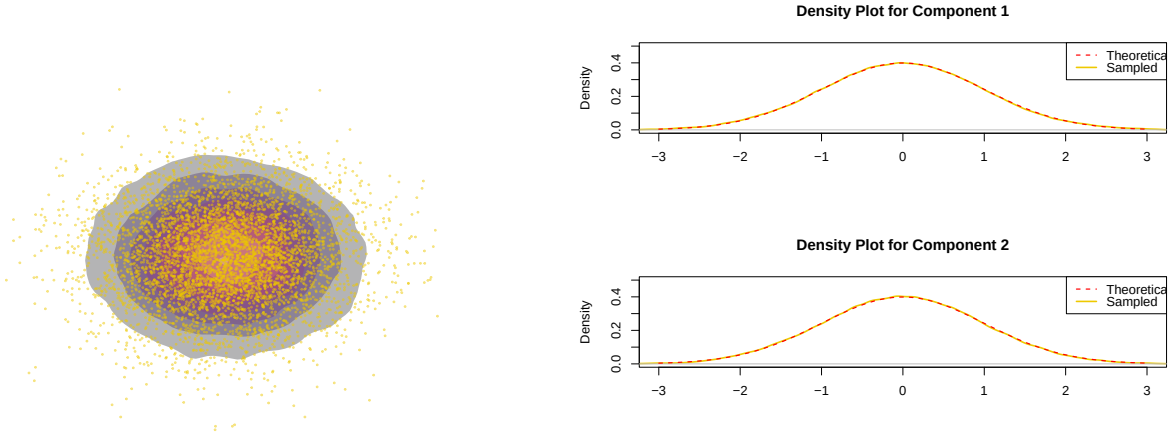

Figure 3: (left) Visualizing the samples generated using MHMC for a bi-variate standard Gaussian target. (right) comparing the empirical density to the theoretical density of the target. Note: G = 0.5

Algorithm 2 Magnetic HMC (MHMC)

Require: H, G, L, ϵ

Draw $\mathbf{p} \sim \mathcal{N}_d(0, I_d)$

Calculate $(\mathbf{x}^*, \mathbf{p}^*) = \mathbf{\Phi}^{\mathbf{G}}_{\mathbf{L}\epsilon}(\mathbf{x}, \mathbf{p})$

Calculate the acceptance ratio:

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

if runif$(0, 1) < \alpha((\mathbf{x}, \mathbf{p}), (\mathbf{x}^*, \mathbf{p}^*))$ **then**

    $\mathbf{x}_{k+1} = \mathbf{x}^*$

else

    $\mathbf{x}_{k+1} = \mathbf{x}_k$

end if

2.4 Dual Averaging

The problem of tuning MCMC parameters often involves solving a stochastic convex optimization problem. A common method implemented in literature to deal with such situations is dual averaging

(Nesterov, 2009). To implement this routine for MCMC, we replace the stochastic gradients with a statistic H_t that, speaking heuristically, measures the "goodness" of the algorithm. Assuming that we want to find a tuning for parameters $\mathbf{\Theta} \in \mathbb{R}^p$ such that $h(\mathbf{\Theta}) = E_t[H_t \mid \mathbf{\Theta}] = 0$, we apply the updates:

$$\mathbf{\Theta}_{t+1} = \mu - \frac{\sqrt{t}}{\gamma}\frac{1}{t+t_0}\Sigma_{i=1}^{t} H_i, \tag{10a}$$

$$\bar{\mathbf{\Theta}}_{t+1} = \eta\mathbf{\Theta}_{t+1} + (1-\eta)\bar{\mathbf{\Theta}}_t. \tag{10b}$$

Here, μ is an arbitrary point that $\bar{\mathbf{\Theta}}_t$ shrink towards. $\gamma > 0$ is a parameter that controls the amount of shrinkage, and $t_0 > 0$ is a free parameter that stabilizes the algorithm for the initial iterations. We take $\eta_t = t^{-\kappa}$ and set $\bar{\mathbf{\Theta}}_1 = \mathbf{\Theta}_1$. It is shown in Nesterov (2009) that as $t \to \infty$:

$$\mathbf{\Theta}_{t+1} - \mathbf{\Theta}_t = O(-H_t t^{-0.5}),$$

which goes to zero as long as the statistic H_t is bounded. The sequence of averaged iterates $\mathbf{\Theta}_{t+1}$ are hence guaranteed to converge to a value such that $h(\mathbf{\Theta}_t) \to 0$.

In common implementations for MCMC algorithms such as Hoffman et al. (2014), we set $t_0 = 0$ and H_t as:

$$H_t = \alpha(\mathbf{z_t}, \mathbf{z_{t-1}}); \tag{11a}$$

$$h(\mathbf{\Theta}) = E_t[H_t \mid \mathbf{\Theta}]. \tag{11b}$$

In the case of HMC, our parameters are $\mathbf{\Theta} = \{\epsilon\}$ using these updates. The algorithm heuristic for selecting a "good" initial value is explained in Algorithm. 3. Finally, we use Algorithm. 4 to use these initial values and tune them to an ideal tuned value for the specific target. This algorithm requires as input a target simulation length $\lambda \approx L\epsilon$, a mean acceptance probability δ, and several iterations M^{adapt} after which we stop adaptation. Hoffman et al. (2014) Section. 4 gives a detailed analysis of the set of values that work best for these hyper-parameters, and I use the same in the implementations of this algorithm.

2.5 Reccursive Domain Exploration

The final parameter I analyze for the algorithm is the number of leapfrog steps that the algorithm takes. The question of when the algorithm has simulated the path of the particle for "long enough" is answered by the heuristic presented in Hoffman et al. (2014) in the form of the sampler taking a "U-Turn." This heuristic aims to find the point where the distance between the proposed position, say $\tilde{\mathbf{x}}$ and the starting position $\mathbf{x}$ does not increase further. Using this criterion, we run the sampler till the point,

$$(\tilde{\mathbf{x}} - \mathbf{x}) \circ \mathbf{p} < \delta_{max}, \tag{12}$$

Algorithm 3 Heuristic for selecting an initial ϵ

Initialize $\epsilon = 1$, $\mathbf{p} \sim \mathcal{N}_d(0, I_d)$
Set $(\mathbf{x}^*, \mathbf{p}^*) = \mathbf{\Phi}_{\mathbf{1}\epsilon}(\mathbf{x}, \mathbf{p})$
Calculate the acceptance ratio:

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

set $a = 2I[\alpha((\mathbf{x}, \mathbf{p}), (\mathbf{x}^*, \mathbf{p}^*)) > 0.5] - 1$
while $\left(\frac{H(\mathbf{x}^*, \mathbf{p}^*)}{H(\mathbf{x}, \mathbf{p})}\right)^a > 2^{-a}$ **do**
    Set $\epsilon \leftarrow 2^a \epsilon$
    Set $(\mathbf{x}^*, \mathbf{p}^*) = \mathbf{\Phi}_{\mathbf{L}\epsilon}(\mathbf{x}, \mathbf{p})$
end while

Algorithm 4 Hamiltonian Monte Carlo with Dual Averaging

Given $\mathbf{x}^0, \delta, \lambda, \mathcal{L}, M, M^{\text{adapt}}$
Set $\epsilon_0 = \text{FindReasonableEpsilon}(\mathbf{x})$, $\mu = \log(10\epsilon_0)$, $\bar{\epsilon}_0 = 1$, $\bar{H}_0 = 0$, $\gamma = 0.05$, $t_0 = 10$, $\kappa = 0.75$
for $m = 1$ to M **do**
    Resample $\mathbf{p}^0 \sim \mathcal{N}(0, I)$
    Set $\mathbf{x}^m \leftarrow \mathbf{x}^{m-1}$, $\hat{\mathbf{x}} \leftarrow \mathbf{x}^{m-1}$, $\tilde{\mathbf{p}} \leftarrow \mathbf{p}^0$
    $L_m = \max\{1, \text{Round}(\lambda/\epsilon_{m-1})\}$
    for $i = 1$ to L_m **do**
        Set $(\tilde{\mathbf{x}}, \tilde{\mathbf{p}}) \leftarrow \text{Leapfrog}(\hat{\mathbf{x}}, \tilde{\mathbf{p}}, \epsilon_{m-1})$
    end for
    With probability $\alpha = \min\left\{1, \frac{\exp\{\mathcal{L}(\tilde{\mathbf{x}}) - \frac{1}{2}\tilde{\mathbf{p}}^\top\tilde{\mathbf{p}}\}}{\exp\{\mathcal{L}(\mathbf{x}^{m-1}) - \frac{1}{2}\mathbf{p}^0 \cdot \mathbf{p}^0\}}\right\}$, set $\mathbf{x}^m \leftarrow \tilde{\mathbf{x}}$, $\mathbf{p}^m \leftarrow -\tilde{\mathbf{p}}$
    if $m \leqslant M^{\text{adapt}}$ **then**
        Set $\bar{H}_m = \left(1 - \frac{1}{m+t_0}\right)\bar{H}_{m-1} + \frac{1}{m+t_0}(\delta - \alpha)$
        Set $\log \epsilon_m = \mu - \frac{\sqrt{m}}{\gamma}\bar{H}_m$
        Set $\log \bar{\epsilon}_m = m^{-\kappa} \log \epsilon_m + (1 - m^{-\kappa}) \log \bar{\epsilon}_{m-1}$
    else
        Set $\epsilon_m = \bar{\epsilon}_{M^{\text{adapt}}}$
    end if
end for

where δ_{max} is a pre-defined error tolerance. I use this understanding to construct a binary exploration paradigm to obtain a "good enough" value of L for a given state $(\mathbf{x}, \mathbf{p})$. In Algorithm. 5, I describe the binary-tree-like methodology that generates the path of the sampler till the criteria in Eq. 12 is fulfilled for a given δ_{max} and the maximum iteration for the algorithm for, "max-iter".

3 Eliminating the need to Hand-Tune MHMC

After discussing the required background for HMC, MHMC, and various tuning techniques for HMC algorithms, we are prepared to develop the problem-agnostic implementation of Magnetic HMC. In this section, I discuss the detailed calculations and derive the final proposed algorithm.

Algorithm 5 Reccursive Exploration to Tune L

function BUILDTREE($\mathbf{x}, \mathbf{p}, \epsilon, \delta_{max}$, max-iter)
  Set $\mathbf{x}_+ \leftarrow \mathbf{x}$, $\mathbf{x}_- \leftarrow \mathbf{x}$, $\mathbf{p}_+ \leftarrow \mathbf{p}$, and $\mathbf{p}_- \leftarrow \mathbf{p}$
  Set $s = 1$ and $j = 1$
  while $s <$ max-iter **do**
    Choose a direction $v_j \sim$ Uniform($\{-1, 1\}$).
    if $v_j = -1$ **then** ▷ We want to move ahead in the counter-clockwise direction
      for $i = 1$ to j **do**
        Set $(\mathbf{x}_-, \mathbf{p}_-) \leftarrow$ Leapfrog$(\mathbf{x}_-, \mathbf{p}_-, \epsilon)$
        if $(\mathbf{x}_- - \mathbf{x}_+) \circ \mathbf{p}_- \leqslant \delta_{\mathbf{max}}$ **then**
          return $i + j$
        end if
      end for
    else ▷ We want to move ahead in the clockwise direction
      for $i = 1$ to j **do**
        Set $(\mathbf{x}_+, \mathbf{p}_+) \leftarrow$ Leapfrog$(\mathbf{x}_+, \mathbf{p}_+, \epsilon)$
        if $(\mathbf{x}_+ - \mathbf{x}_-) \circ \mathbf{p}_+ \leqslant \delta_{\mathbf{max}}$ **then**
          return $i + j$
        end if
      end for
    end if
    $j \leftarrow 2j$
    $s \leftarrow s + 1$
  end while
  return j
end function

Before going into the implementation details, I made a few comments about the efficacy of the MHMC algorithm, particularly invariance and ergodicity.

3.1 Comments on HHMC

Here, I discuss the invariance and ergodicity of the Magnetic HMC algorithm.

Theorem 1. *The MHMC kernel is π invariant.*

Proof. Looking at Lemma. 1, we realize that the non-canonical Hamiltonian dynamics underscoring MHMC are volume-preserving and symplectic.

Further, looking at Lemma. 2, we realize that one cannot simply substitute the leapfrog with the canonical leapfrog we develop in Eq.9. We need to augment E and G into the state variable to ensure that the proposed update is still time-reversible.

In particular, say that we wish to use $\Phi^A_{\tau,H}(\theta, \mathbf{p})$ as a transition kernel with fixed, non-zero values of $\mathbf{E} = \mathbf{E}_0$ and $\mathbf{G} = \mathbf{G}_0$. We first augment the state-space by placing a symmetric, binary distribution independently over $\mathbf{E}$ and $\mathbf{G}$ such that $p(\mathbf{E} = \mathbf{E}_0) = p(\mathbf{E} = -\mathbf{E}_0) = 1/2$ and $p(\mathbf{G} = \mathbf{G}_0) = p(\mathbf{G} = -\mathbf{G}_0) = 1/2$, independently of $\theta, \mathbf{p}$:

$$\rho(\theta, \mathbf{p}, \mathbf{E}, \mathbf{G}) \propto e^{-H(\theta,\mathbf{p})} p(\mathbf{E}) p(\mathbf{G}). \tag{13}$$

Importantly, this augmentation leaves the distribution over $\theta, \mathbf{p}$ intact when $\mathbf{E}$ and $\mathbf{G}$ are marginalized. Just as applying the momentum flip operator, $\Phi_{\mathbf{P}} : (\theta, \mathbf{p}, \mathbf{E}, \mathbf{G}) \to (\theta, -\mathbf{p}, \mathbf{E}, \mathbf{G})$, is a deterministic, energy-preserving, volume-preserving transformation, the $\mathbf{E}, \mathbf{G}$ flip operators, $\Phi_{\mathbf{E}} : (\theta, \mathbf{p}, \mathbf{E}, \mathbf{G}) \to (\theta, \mathbf{p}, -\mathbf{E}, \mathbf{G})$ and $\Phi_{\mathbf{G}} : (\theta, \mathbf{p}, \mathbf{E}, \mathbf{G}) \to (\theta, \mathbf{p}, \mathbf{E}, -\mathbf{G})$, are also deterministic, energy-preserving, volume-preserving transformations that leave (13) invariant for this particular augmentation with $p(\mathbf{E})$ and $p(\mathbf{G})$. We can now build a self-inverse operator $\tilde{\Phi}^A_{\tau,H}(\theta, \mathbf{p})$, composed of simulating the Hamiltonian flow as $\Phi^A_{\tau,H}(\theta, \mathbf{p})$ plus $\Phi_{\mathbf{E}} \circ \Phi_{\mathbf{G}} \circ \Phi_{\mathbf{P}}$, a flip of $\mathbf{p}, \mathbf{E}, \mathbf{G}$, as:

$$\tilde{\Phi}^A_{\tau,H}(\theta, \mathbf{p}) = \Phi_{\mathbf{E}} \circ \Phi_{\mathbf{G}} \circ \Phi_{\mathbf{P}} \circ \Phi^A_{\tau,H}(\theta, \mathbf{p}). \tag{14}$$

This now ensures that we have a time-reversible and volume-preserving algorithm. From the results discussed in Tierney (1994) and Neal et al. (2011), we can be assured that MHMC follows detailed balance and, hence, is π invariant.

□

Note. I take parts of this proof from Appendix Section 2 Tripuraneni et al. (2017).

Before discussing the ergodicity of MHMC, I state a few results without proof that allow us to establish constraints on MH-like MCMC samplers.

Theorem 2. *If $\pi P = \pi$, P is π-irreducible, aperiodic, and Harris recurrent, then for every initial distribution λ*

$$\lim_{n\to\infty} \|\lambda P^n - \pi\| = 0.$$

Consequently, for all $x \in \mathcal{X}$

$$\lim_{n\to\infty} \|P^n(x, \cdot) - \pi(\cdot)\| = 0.$$

Moreover, for any two initial distributions λ_1 and λ_2

$$\lim_{n\to\infty} \|\lambda_1 P^n - \lambda_2 P^n\| = 0.$$

The Markov chain is then said to be ergodic.

Theorem 3 (Roberts and Rosenthal (2009))**.** *Let $FP = F$, P is F-irreducible. Then, the following are equivalent.*

1. *P is Harris recurrent.*
2. *$\forall x \in \mathcal{X}, A \in \mathcal{B}(\mathcal{X})$ with $F(A) = 1$, $\Pr(\tau_A < \infty \mid X_0 = x) = 1$.*
3. *$\forall x \in \mathcal{X}, A \in \mathcal{B}(\mathcal{X})$ with $F(A) = 0$, $\Pr(X_n \in A, \forall n \mid X_0 = x) = 0$.*

Proof. $1 \Rightarrow 2 \Leftrightarrow 3$ are straightforward. I do not prove the rest, but one can find the proof of $3 \Rightarrow 1$ in Roberts and Rosenthal (2009). □

Theorem 4. *Suppose Markov chain P is F-irreducible and there exists S such that $F(S) > 0$ and $P(x, \{x\}) > 0$ for all $x \in S$, then P is aperiodic.*

Note. This shows that this result will follow any sampler with a positive rejection probability.

Theorem 5. *The MHMC algorithm is ergodic.*

Proof. Given that we've established the sampler's invariance, I focus this proof on irreducibility (from Theorem 4 consequently aperiodicity) and Harris recurrence.

The irreducibility of the algorithm comes from the nature of Hamiltonian dynamics, where we can sample from the entire momentum space. As clearly presented in Section-2 Durmus et al. (2017), we know that for a continuously differentiable class of targets π and a stochastically chosen value of L, HMC is irreducible. We ensure both the conditions in the mapping $\mathbf{\Phi}^{G}_{L,\epsilon}$. Hence, we can conclude that MHMC is irreducible and, therefore, aperiodic (Theorem 4).

For Theorem. 3, we observe that MHMC satisfies condition *2.* The proof for this result is similar to a generalized version for MH algorithms.

Finally, we can use the Theorem using all the results we have discussed so far.2 to ensure that MHMC is ergodic. □

3.2 The Knight and Bishop Algorithm

According to the dual averaging paradigm discussed in Section.2.4 I choose H_t to be the acceptance probability of MHMC algorithm. We aim to tune the parameters so that the acceptance probability of the algorithm tends to $\delta = 0.65$ (Beskos et al., 2013). Within the updates discussed in 4, we need to account for the fact that G flips its sign to fulfill the detailed balance. The solution is to tune $\mid G \mid$ and account for the sign separately. A detailed explanation of this tuning strategy is described in the proposed Algorithm.6, which I call, "The Knight and Bishop" algorithm. The algorithm adaptively tunes the values of all the parameters independently for pre-determined M_{adapt} steps and ensures that it uses those values of the parameters to sample from the target distribution. After tuning, we run the same routine as the one mentioned in the algorithm.2, which I prove to be ergodic in Section.3.1. Hence, we can be assured that the Knight and Bishop (KNB) Algorithm is ergodic.

I also provide a visualization of the working of this algorithm on a uni-variate standard normal target in Fig.4. I provide the visualization of the efficacy of my proposed algorithm on more complicated target distributions in the later sections. Here, we notice that the KNB algorithm ensures convergence to the target distribution without manual tuning of any of the algorithm parameters.

Algorithm 6 Magnetic HMC with Dual Averaging (The Knight and Bishop Algorithm)

Given Desired sample size N
Set $\epsilon = \text{FindReasonableEpsilon}(0)$, $G = \text{FindReasonableG}(0)$
Set $\mu_1 = \log(10\epsilon)$, $\mu_2 = \log(10\gamma)$, $\epsilon_0 = 1$, $\gamma_0 = 0.5$, $H_0 = 0$
Set $H = H_0$, Madapt $= N/100$, $L_{\text{tot}} = 0$, $a = 0$
for $i = 2$ to N **do**
    $\mathbf{p} \sim \mathcal{N}(0, 1)$
    $\mathbf{x} \leftarrow \mathbf{x_{i-1}}$
    $(L, v) \leftarrow \text{BuildTree}(\mathbf{x}, \mathbf{p}, \epsilon, \gamma)$
    if $i < \text{M}_{\text{adapt}}$ **then**
        $H \leftarrow \left(1 - \frac{1}{i+t_0}\right) H + \frac{1}{i+t_0}(\delta - \alpha)$
        $\epsilon \leftarrow \exp\left(\mu_1 - \frac{\sqrt{i}}{\gamma} H\right)$
        $\epsilon_0 \leftarrow \exp\left(i^{-\kappa} \log \epsilon + (1 - i^{-\kappa}) \log \epsilon_0\right)$
        $\text{sign} \leftarrow 1$
        if $G < 0$ **then** $G \leftarrow -G$, $\text{sign} \leftarrow -1$
        end if
        $G \leftarrow \exp\left(\mu_2 - \frac{\sqrt{i}}{\gamma} H\right)$
        $G_0 \leftarrow \exp\left(i^{-\kappa} \log G + (1 - i^{-\kappa}) \log G_0\right)$
        $G \leftarrow \text{sign} \cdot G$
        $L_{\text{tot}} \leftarrow L_{\text{tot}} + L$
    else
        $\epsilon \leftarrow \epsilon_0$
        $L \leftarrow L_{\text{tot}}/\text{M}_{\text{adapt}}$
        if $G < 0$ **then** $G \leftarrow -G_0$
        else $G \leftarrow G_0$
        end if
    end if
    $(\mathbf{p_{prop}}, \mathbf{x_{prop}}) \leftarrow \mathbf{\Phi}^{\mathbf{G}}_{\mathbf{L}, \mathbf{v}\epsilon}(\mathbf{x}, \mathbf{p})$
    $a \leftarrow H(\mathbf{x}, \mathbf{p}) - H(\mathbf{x_{prop}}, \mathbf{p_{prop}})$
    if $\log U(0, 1) \leqslant a$ **then**
        $\mathbf{x}_i \leftarrow \mathbf{x}_{prop}$
        $G \leftarrow -G$
    else
        $\mathbf{x}_i \leftarrow \mathbf{x}_{i-1}$
    end if
end for

4 Experiments

In this section, I implement the MHMC algorithm on several popular targets and compare the sampler results with common samplers such as HMC, random walk MH, and raHMC to see how the algorithm performs in relation to other samplers. For all of my implementations, I let my algorithm tune the parameters for the initial 1000 samples and then generate 5000 samples to compare them with other algorithms. For samplers that require tuning, I tune them to the ideal

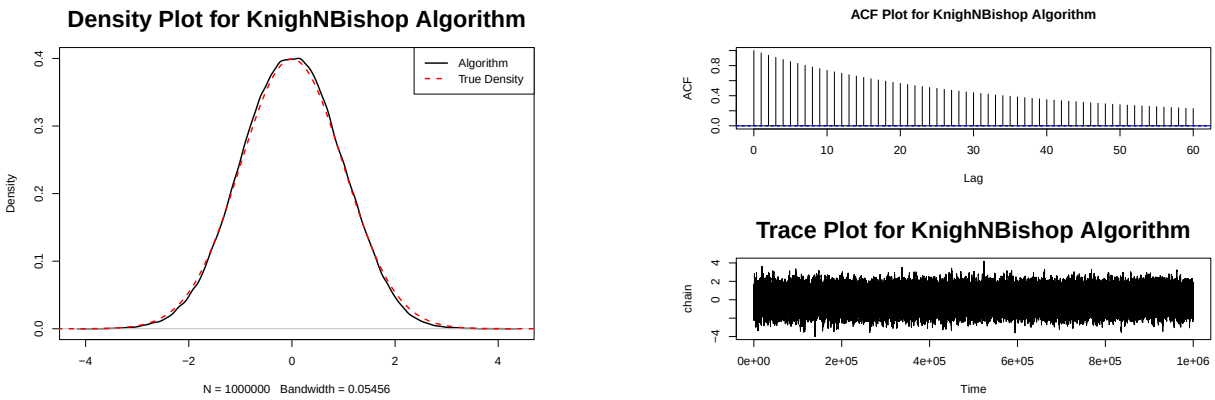

Figure 4: Illustrating the sample quality and density for the proposed Knight and Bishop Algorithm.

acceptance ratios as established in the literature and also provide the tuned parameters used.

Before more complicated targets, I also implemented the algorithm for the bi-variate Gaussian target that I've used to illustrate all other algorithms in this paper. I present a visualization of these samples and their estimated densities compared to the theoretical density of the target in Fig. 5.

Note. To implement the heuristic for tuning ϵ and G, we need to use log ratios instead of α as presented in the Algorithm. 3.

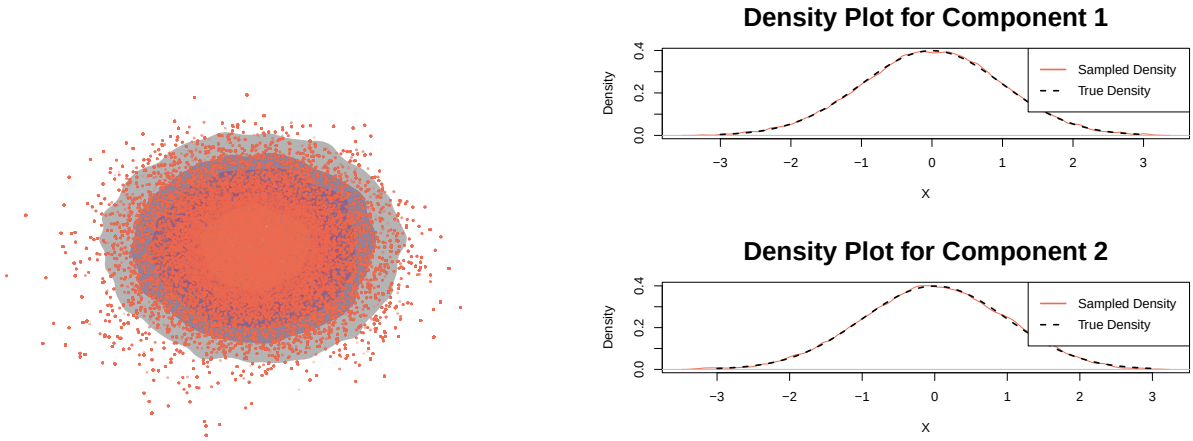

Figure 5: (left) Visualizing the samples generated using KnB for a bi-variate standard Gaussian target. (right) comparing the empirical density to the theoretical density of the target.

4.1 Bi-variate Mixture of Gaussians

First, we consider a mixture of 20 bi-variate Gaussian distributions with different means (as presented in Kou et al. (2006)). The target density is as follows:

$$\pi(\mathbf{x}) \propto \sum_{j=1}^{20} \exp\left(-\frac{(\mathbf{x}-\boldsymbol{\mu}_j)^\top(\mathbf{x}-\boldsymbol{\mu}_j)}{2\sigma^2}\right),$$

where, $\sigma^2 = 0.05$ and the list of μ_is is as presented in Kou et al. (2006)). This data set is characteristic of difficult domain exploration and is popular for testing sampling algorithms like raHMC, THMC, and other equivalents. I visualize the performance of HMC and THMC in Fig. 6. We analyze the performance of Magnetic HMC on the same target in Fig. 7. We notice that the acceptance rate of the MHMC sampler is extremely low because tuning the sampler for complicated targets is extremely difficult. It is unfortunate that the KnB sampler fails to compute the optimum values of L, ϵ and G.

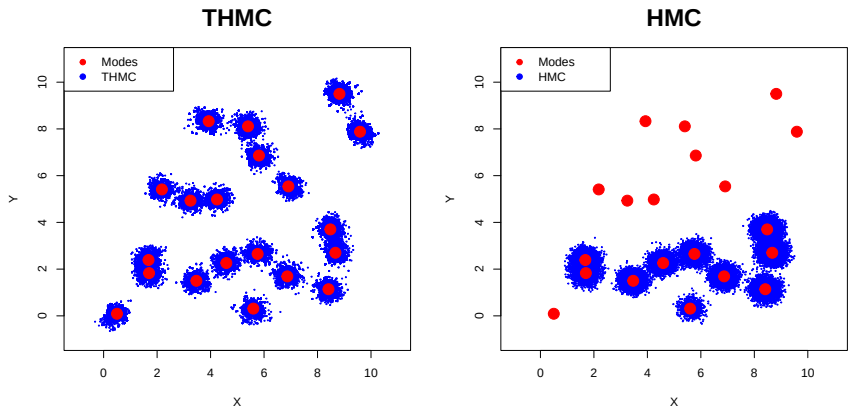

Figure 6: Comparing the performance of THMC and HMC on the 20-modal target distribution. We notice that HMC gets localized to a few modes, and in contrast, THMC is efficiently able to explore the entire target domain.

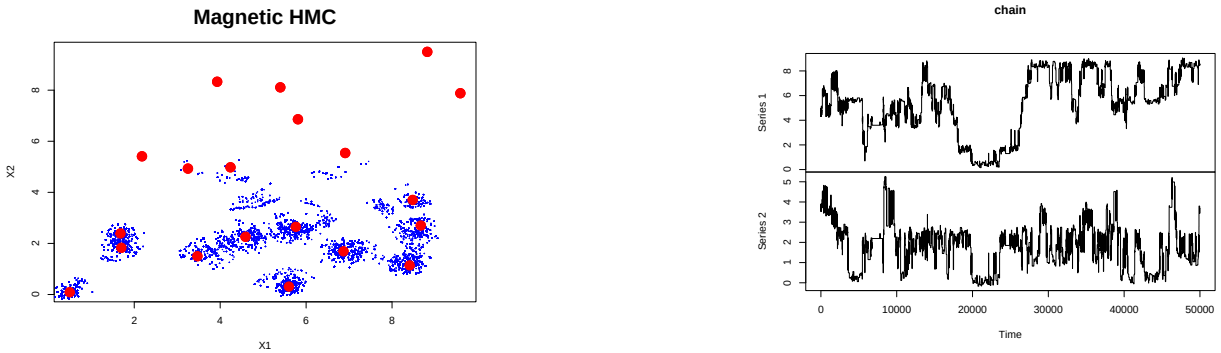

Figure 7: The performance of MHMC on the 20-modal target distribution

4.2 Targets with Geometric Patterns

In this section, I analyze the target distributions with subtle geometric structures in addition to multimodality. The first of these is a mixture of distributions supported on the boundary of three *concentric* ℓ_1 balls in $\mathbb{R}^d$, and the target $\pi(q)$ is given by:

$$\pi(q) \propto \sum_{i=1}^{3} \exp\left(-\frac{(\|q\|_1 - r_i)^2}{2\sigma^2}\right),$$

where $r_1 = 4$, $r_2 = 8$, $r_3 = 16$ and $\sigma = 0.5$. We consider the cases when $d = 2$ and $d = 3$. I visualize the performance of the MHMC algorithm on the target and contrast it to the performance of THMC in Fig. 8. We observe that the MHMC algorithm is not very well able to explore the entire domain space of the target. I note again that the KnB sampler was not able to tune the parameters due to a steep rise in the gradients during dual averaging.

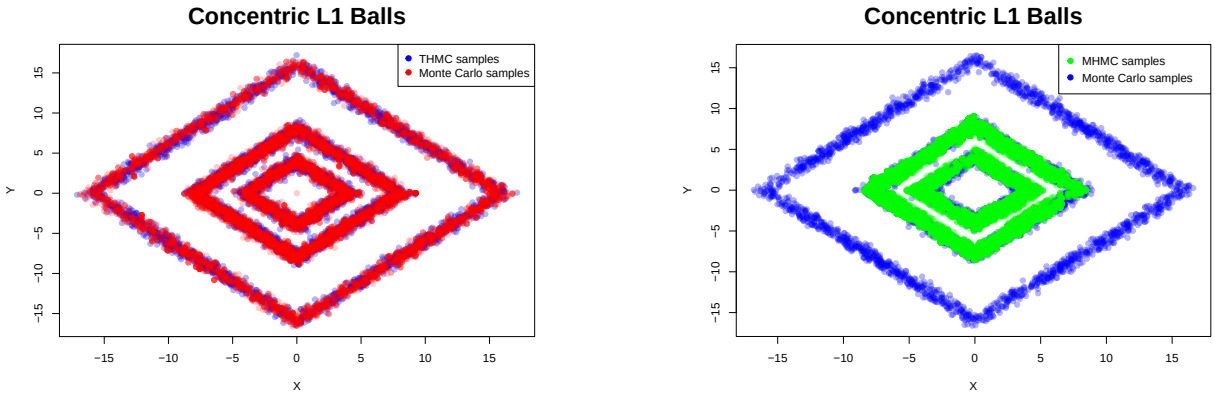

Figure 8: Contrasting the performance of THMC (left) with MHMC (right) for the ℓ_1 concentric balls target

	Component-1	Component-2	Acceptance %
MHMC	24.803	34.774	18%
THMC	86.901	137.746	38%

Table 1: Comparing ESS values for MHMC and THMC for the concentric ℓ_1 concentric balls target

My final example is a mixture of distributions supported on the boundary of *nested* ℓ_1 balls in $\mathbb{R}^d$. The target distribution, $\pi(q)$, is given by:

$$\pi(q) \propto \sum_{i=1}^{5} \exp\left(-\frac{(\|q - \mu_i\|_1 - r_i)^2}{2\sigma^2}\right),$$

where $\mu_1 = 0$, $r_1 = 20$, and $\|\mu_i\|_1 = 2$, $r_i = 2$ for $2 \leqslant i \leqslant 5$. I visualize the results of the sampler in Fig. 9 and compare the ESS values of the MHMC algorithm and the THMC sampler in Table. 2.

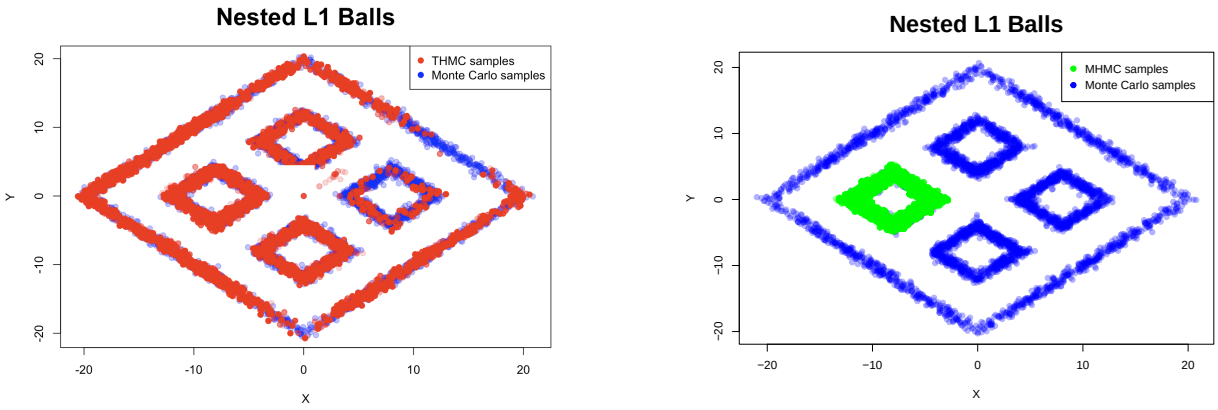

Figure 9: Contrasting the performance of THMC (left) with MHMC (right) for the ℓ_1 nested balls target

	Component-1	Component-2	Acceptance %
MHMC	23.438	55.745	14%
THMC	56.901	117.746	34%

Table 2: Comparing ESS values for MHMC and THMC for the nested ℓ_1 balls target

5 Discussion

MHMC tries to generalize the Hamiltonian dynamics that underscore the HMC algorithm, promising a sampler that can tackle a much broader scope of targets. However, this generalization adds parameters to the model that are un-intuitive to tune, and even with popular tuning paradigms like dual averaging, they do not yield stable mechanisms that enable us to tune these models. The KnB sampler is my attempt at developing a problem-agnostic tuning strategy that is able to solve this problem of MHMC and develop a solution that helps users sample from complicated targets without much prior knowledge. However, as evident from the experiments, a lot of improvement needs to be made on the tuning strategy implemented in KnB to ensure that the algorithm does not break down when faced with complicated targets.

In my future work, I also want to expand the scope of tuning G for multi-variate distributions where G would be a matrix capable of tuning different barrier parameters for various chain components. The sampler has the potential to account for the underlying correlation patterns in the target that no other variant of HMC is able to account for in popular literature. I want to be able to look at specific, high dimensional targets (for example Vishwanath and Tak (2024)) and implement the KnB sampler to understand the quality of samples that it can generate and compare them to popular alternatives like NUTS, MALA, Barker's proposal, etc.

References

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., and Stuart, A. (2013). Optimal tuning of the hybrid monte carlo algorithm.

Bornn, L., Cornebise, J., and Peters, G. W. (2010). Discussion of "riemann manifold langevin and hamiltonian monte carlo methods" by m. girolami and b. calderhead.

Durmus, A., Moulines, E., and Saksman, E. (2017). On the convergence of hamiltonian monte carlo. *arXiv preprint arXiv:1705.00166*.

Hoffman, M. D., Gelman, A., et al. (2014). The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623.

Kou, S. C., Zhou, Q., and Wong, W. H. (2006). Equi-energy sampler with applications in statistical inference and statistical mechanics. *The Annals of Statistics*, 34(4).

Neal, R. M. et al. (2011). Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2.

Nesterov, Y. (2009). Primal-dual subgradient methods for convex problems. *Mathematical programming*, 120(1):221–259.

Roberts, G. O. and Rosenthal, J. S. (2009). Examples of adaptive mcmc. *Journal of computational and graphical statistics*, 18(2):349–367.

Tierney, L. (1994). Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, 22(4):1701 – 1728.

Tripuraneni, N., Rowland, M., Ghahramani, Z., and Turner, R. (2017). Magnetic hamiltonian monte carlo. In *International Conference on Machine Learning*, pages 3453–3461. PMLR.

Vats, D. (2023). Hamiltonian monte carlo for (physics) dummies.

Vishwanath, S. and Tak, H. (2024). Repelling-attracting hamiltonian monte carlo.