Practical Hamiltonian Monte Carlo on Riemannian Manifolds via Relativity Theory

Kai Xu [1] **Hong Ge** [2]

Abstract

Hamiltonian Monte Carlo (HMC) samples from an unnormalized density by numerically integrating Hamiltonian dynamics. Girolami & Calderhead (2011) extend HMC to Riemannian manifolds, but the resulting method faces integration instability issues for practical usage. While previous works have tackled this challenge by using more robust metric tensors than Fisher's information metric, our work focuses on designing numerically stable Hamiltonian dynamics. To do so, we start with the idea from Lu et al. (2017), which designs momentum distributions to upper-bound the particle speed. Then, we generalize this Lu et al. (2017) method to Riemannian manifolds. In our generalization, the upper bounds of velocity norm become *position-dependent*, which intrinsically limits step sizes used in high curvature regions and, therefore, significantly reduces numerical errors. We also derive a more tractable algorithm to sample from relativistic momentum distributions without relying on the mean-field assumption.

1. Introduction

Hamiltonian Monte Carlo (HMC) is a Markov chain Monte Carlo (MCMC) method to sample from a target distribution π known up to its unnormalized log-density $\mathcal{L}$, i.e. $\pi(\mathbf{q}) \propto \exp(\mathcal{L}(\mathbf{q}))$. MCMC works by iteratively proposing new states based on current states to produce a chain of samples converging asymptotically to the target. In HMC, new states are proposed by integrating the Hamiltonian dynamics with position $\mathbf{q}(\tau)$ and momentum $\mathbf{p}(\tau)$ at time τ defined as

$$\mathrm{d}\mathbf{q}/\mathrm{d}\tau = \nabla_{\mathbf{p}} H \quad \text{and} \quad \mathrm{d}\mathbf{p}/\mathrm{d}\tau = -\nabla_{\mathbf{q}} H, \qquad (1)$$

where $H(\mathbf{q}, \mathbf{p}) = U(\mathbf{q}) + K(\mathbf{q}, \mathbf{p})$ is the Hamiltonian energy and U, K are the potential and kinetic energy, respectively. To sample from a target with access to its unnormalized log-density $\mathcal{L}$, we construct a Hamiltonian system with $U(\mathbf{q}) = -\mathcal{L}(\mathbf{q})$ and define the kinetic energy K or equivalently the momentum distribution. We then propose a sample $\mathbf{q}'$ from a trajectory $\hat{\mathbf{t}}$ obtained from numerical integration starting from the current $\mathbf{q}$ and a random $\mathbf{p}$. There are various ways to define the kinetic energy, to integrate Hamiltonian dynamics numerically, and to draw samples from the numerically integrated trajectory. The combinations of these choices lead to different types of HMC algorithms in the literature (Neal, 2011; Hoffman & Gelman, 2011; Girolami & Calderhead, 2011; Betancourt, 2018). Due to its high efficiency for high-dimensional, differentiable densities and its availability in modern probabilistic programming languages (Carpenter et al., 2017; Ge et al., 2018), HMC has been applied to a range of domains including statistical physics (Duane et al., 1987), neuroscience (Sengupta et al., 2016; Linderman et al., 2016; Jha et al., 2022), bioinformatics (Larranaga et al., 2006), social science (Peng et al., 2016) and machine learning (Barber, 2012).

HMC's sampling efficiency highly depends on the local geometric structure of the target. Various HMC extensions have proposed to utilize the underlying geometry through the kinetic energy function by using a global pre-conditioning matrix (Neal, 2011; Carpenter et al., 2017), or a position-dependent conditioning matrix (Girolami & Calderhead, 2011; Betancourt, 2013), commonly referred as Riemannian HMC (RHMC). For RHMC, numerical integration becomes more difficult, leading to significantly more divergent transitions and hindering sampling efficiency. This is because the position-dependent momentum distribution sometimes produces large momentum realizations in regions with high curvature. These large momentum variables often lead to instability in numerical integration. A naive solution requires using an unrealistically small step size that works across all parameter space regions during generalized leapfrog integration: regions with large curvature determine the upper bound for the acceptable step size.

Recently, Lu et al. (2017) proposed an effective way to stabilize Hamiltonian simulation inspired by Einstein's theory

[1]MIT-IBM Watson AI Lab, Cambridge MA, United States [2]University of Cambridge, Cambridge, United Kingdom. Correspondence to: Kai Xu <xuk@ibm.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

of *special relativity*, which states any particle cannot travel faster than the speed of light. In Hamiltonian simulation, the velocity, $\mathbf{v} := \mathrm{d}\mathbf{q}/\mathrm{d}\tau$, is used to update the position variable $\mathbf{q}$ together with a choice of step size (section 2.3). The main idea is to design kinetic energy such that the norm of velocity (i.e. speed) is upper-bounded, which is actually similar to the idea of gradient clipping (i.e. constraining gradients to a fixed-radius ball) in deep learning, The resulting method, relativistic Monte Carlo, consists of a kinetic energy based on the relativity theory and upper-bounds the speed of the particles in the Hamiltonian simulation (Lu et al., 2017).

This paper extends Lu et al. (2017) to the theory of *general relativity* by developing relativistic Monte Carlo on Riemannian manifolds. The proposed method, *general relativistic HMC* (GR-HMC), stabilizes HMC on Riemannian manifolds by upper-bounding particle velocity in a position-dependent manner to reduce Hamiltonian integration errors in high-curvature regions. For this, we propose a new kinetic energy corresponding to a multivariate relativistic momentum distribution with a position-dependent metric. For efficient sampling, we develop a novel sampler for the multivariate momentum distribution using the Box-Muller transform and introduce computationally efficient implementations and approximations for GR-HMC.

2. Background

In standard HMC, the transition kernel $\mathcal{K}(Q, \cdot)$, where Q is the current position state, consists of three steps

1. Sampling a momentum variable P according to the kinetic energy of the Hamiltonian system;
2. Simulating a numerical trajectory following (1);
3. Sampling a new phase point (Q', P') from the simulated trajectory, and Q' is the new state.

For step 1, the kinetic energy $K(\mathbf{q}, \mathbf{p})$ defines the momentum distribution as $\mathrm{pr}(\mathbf{p}) \propto \exp(-K(\mathbf{q}, \mathbf{p}))$, where pr denotes a probability measure. A common choice of the momentum distribution is a multivariate Gaussian with a zero-vector mean and a covariance matrix $\mathbf{G}$, which corresponds to kinetic energy $K_{\mathbf{G}}(\mathbf{p}) = \frac{1}{2}\mathbf{p}^\top \mathbf{G}^{-1}\mathbf{p}$. A common choice of $\mathbf{G}$ is some positive-definite matrix representing the *global geometry* of the target distribution, e.g. the sample covariance matrix. However, such a metric is not optimal unless the target distribution is Gaussian. Next, we review two important kinetic energy choices on which our method is based.

2.1. Position-dependent momentum

In Riemannian HMC, Girolami & Calderhead (2011) extends HMC from Euclidean manifolds to Riemannian manifolds by introducing a position-dependent Gaussian momentum with the following position-dependent kinetic energy

$$K_{\mathbf{G}}(\mathbf{q}, \mathbf{p}) = \frac{1}{2}\mathbf{p}^\top \mathbf{G}_{\mathbf{q}}^{-1}\mathbf{p} + \frac{1}{2}\log\det(\mathbf{G}_{\mathbf{q}}) \tag{2}$$

where $\mathbf{q}$ and $\mathbf{p}$ are position and momentum variables. This corresponds to a multivariate Gaussian with a zero-vector mean and a position-dependent covariance matrix $\mathbf{G}_{\mathbf{q}}$.

2.1.1. Choices of position-dependent metrics

Fisher information metric Girolami & Calderhead (2011) suggests using the Fisher information metric for $\mathbf{G}_{\mathbf{q}}$, which, for general sampling problems, can be computed as the Hessian matrix of the potential energy $\mathbf{H}_U$. However, $\mathbf{H}_U$ is not guaranteed to be positive-definite, often leading to pathological behavior in sampling.

SoftAbs metric To solve this pathological behavior, Betancourt (2013) proposes the SoftAbs metric that applies a matrix transformation to the Hessian to produce a positive definite matrix $\wr\mathbf{H}_U\wr$. Here the SoftAbs map $\wr\cdot\wr$, which ensures the transformed eigenvalues are positive, is defined as $\wr\mathbf{X}\wr := [\exp(\alpha\mathbf{X}) + \exp(-\alpha\mathbf{X})]\cdot\mathbf{X}\cdot[\exp(\alpha\mathbf{X}) - \exp(-\alpha\mathbf{X})]^{-1}$, where exp is the exponential mappings of matrices. Such transformation results in a "smooth" version of Hessian at singular positions but is still close to the original Hessian matrix. Efficient and stable computation of SoftAbs via eigen-decomposition is used in practice (Betancourt, 2013). In the rest of the paper, we assume $\mathbf{G}(\cdot)$ is SoftAbs.

2.2. Velocity-bounded momentum

Inspired by the wide use of gradient clipping in deep learning, Lu et al. (2017) propose the novel kinetic energy that bounds the particle velocity in Hamiltonian simulation

$$\tilde{K}_{\mathbf{I}}(\mathbf{p}) = mc^2\sqrt{\frac{\mathbf{p}^\top\mathbf{p}}{m^2c^2} + 1} \tag{3}$$

where $\mathbf{p}$ is the momentum variable, m is the "rest mass" and c the "speed of light". Note that we use $\tilde{}$(tilde) to indicate kinetic energy is velocity-bounded.

The momentum $\mathbf{p}$ and velocity $\mathbf{v}$ of a particle is connected by its "relativistic mass" $M_{\mathbf{I}}(\mathbf{p})$ as $\mathbf{v}_{\mathbf{I}} := (M_{\mathbf{I}}(\mathbf{p}))^{-1}\mathbf{p}$ where $M_{\mathbf{I}}(\mathbf{p}) = m\sqrt{\frac{\mathbf{p}^\top\mathbf{p}}{m^2c^2} + 1}$. The relativistic momentum upper bounds the velocity $\mathbf{v}_{\mathbf{I}}$ by c. To see this, note $M_{\mathbf{I}}(\mathbf{p}) = m\sqrt{\frac{\mathbf{p}^\top\mathbf{p}}{m^2c^2} + 1} > m\sqrt{\frac{\mathbf{p}^\top\mathbf{p}}{m^2c^2} + 0} = \|\mathbf{p}\|_2/c$.

Lu et al. (2017) proposes to sample a uni-variate of relativistic momentum distribution via rejection sampling. However,

it is hard to sample multivariate relativistic moment variables directly using (adaptive) rejection sampling. Lu et al. (2017) proposed a heuristic: an easy-to-sample, dimension-wise version of (3) that bounds the absolute value of each dimension of the velocity

$$\tilde{K}_{\mathbf{I}}^{\oplus}(\mathbf{p}) = mc^2 \sum_i \sqrt{\frac{p_i^2}{m^2c^2} + 1}. \tag{4}$$

Here, the superscript $^{\oplus}$ is for dimension-wise independence. This kinetic energy no longer follows the theory of relativity—it bounds each element of the velocity variable instead of its magnitude. However, this momentum distribution is easy to sample using (adaptive) rejection sampling because the variable in each dimension can be sampled independently. Empirically, Lu et al. (2017) only experiments with this dimension-wise momentum distribution on distributions they studied. We relax this dimension-wise independence assumption and propose a general sampling procedure for multivariate relativistic momentum distributions in section 5.1.

We refer to HMC based on (3) as special relativistic HMC (SR-HMC) due to its connections to the theory of special relativity and call those using (4) dimension-wise SR-HMC.

2.3. Generalized leapfrog integration

Hamiltonian systems involving position-dependent momentum are non-separable due to the interaction between momentum and position variables. Consider RHMC for a concrete example of such non-separable Hamiltonian systems:

$$\frac{\partial H}{\partial p_i} = \{\mathbf{G}_{\mathbf{q}}^{-1}\mathbf{p}\}_i \tag{5}$$

$$\frac{\partial H}{\partial \mathbf{q}_i} = -\frac{\partial \mathcal{L}}{\mathbf{q}_i} + \frac{1}{2}\operatorname{tr}\{\mathbf{G}_{\mathbf{q}}^{-1}\frac{\partial \mathbf{G}_{\mathbf{q}}}{\partial \mathbf{q}_i}\} - \frac{1}{2}\mathbf{p}^\top \mathbf{G}_{\mathbf{q}}^{-1}\frac{\partial \mathbf{G}_{\mathbf{q}}}{\partial \mathbf{q}_i}\mathbf{G}_{\mathbf{q}}^{-1}\mathbf{p}$$

One has to use the generalized leapfrog integrator to simulate the Hamiltonian dynamics to ensure the integration is reversible; thus, the overall HMC entails detailed balance. The *update* equations for generalized leapfrog are as follows

$$\mathbf{p}(\tau + \frac{\varepsilon}{2}) = \mathbf{p}(\tau) - \frac{\varepsilon}{2}\nabla_{\mathbf{q}} H\{\mathbf{q}(\tau), \mathbf{p}(\tau + \frac{\varepsilon}{2})\} \tag{6}$$

$$\begin{aligned}\mathbf{q}(\tau + \varepsilon) = \mathbf{q}(\tau) + \frac{\varepsilon}{2}[&\nabla_{\mathbf{p}} H\{\mathbf{q}(\tau), \mathbf{p}(\tau + \frac{\varepsilon}{2})\} \\ &+ \nabla_{\mathbf{p}} H\{\mathbf{q}(\tau + \varepsilon), \mathbf{p}(\tau + \frac{\varepsilon}{2})\}]\end{aligned} \tag{7}$$

$$\mathbf{p}(\tau + \varepsilon) = \mathbf{p}(\tau + \frac{\varepsilon}{2}) - \frac{\varepsilon}{2}\nabla_{\mathbf{q}} H\{\mathbf{q}(\tau + \varepsilon), \mathbf{p}(\tau + \frac{\varepsilon}{2})\}$$

where ε is the step size of choice.

Note that (6) is not a direct update rule as $\mathbf{p}(\tau + \varepsilon)$ appears at both sides of the equation. Therefore, it is usually solved by fixed-point iterations, which can be costly or unstable, depending on the proper choice of the number of iterations.

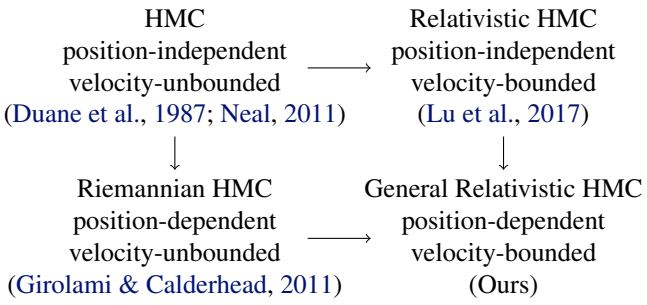


Figure 1: How the proposed method, general relativistic HMC (GR-HMC), extends related works, Riemannian HMC (RHMC) and Relativistic HMC (SR-HMC).

3. Related Work

The most related works are Girolami & Calderhead (2011), which introduced a position-dependent momentum based on Riemannian manifolds, and Lu et al. (2017) that introduces a velocity-bounded momentum based on special relativity. These works are briefly reviewed in section 2. Figure 1 compares standard HMC, these related works and our method.

It's well understood that RHMC suffers from high computational costs. A set of approximations has been proposed to improve the scalability of RHMC by reducing the cost of computing the inversion or derivative of the Hessian matrix, which has a cubic complexity (Patterson & Teh, 2013; Betancourt, 2013; Li et al., 2015) overall. RHMC also suffers from numerical instability when using the popular Fisher information metric proposed by Girolami & Calderhead (2011). Betancourt & Stein (2011); Betancourt (2013) examines the pathology of the Fisher information metric and proposes alternative numerically more stable metrics. Recently, Whalley et al. (2022) studies using randomized integration time (Bou-Rabee & Sanz-Serna, 2017) to improve RHMC's numerical stability. It shares some similarities with our method but still suffers from numerical issues rooted in the extreme velocity values when exploring regions with high curvature of the target distribution. The main development in this paper is orthogonal to such works that control the total integration time (Hoffman & Gelman, 2011; Bou-Rabee & Sanz-Serna, 2017; Whalley et al., 2022). Thus, our method could be combined with these previous methods for optimal performance.

In addition, several previous works have studied non-Gaussian momentum distributions (Zhang et al., 2016; Lu et al., 2017; Livingstone et al., 2017). Usually, Gaussian momentum variables are preferred due to their simplicity and generally good performance. Betancourt (2018) also described some theoretical benefits of Gaussian momenta for high-dimensional targets.

4. General Relativistic Hamiltonian Dynamics

Motivated by the numerical stability of SR-HMC, we extend the idea of bounding velocity to improve the numerical stability of HMC on Riemannian manifolds. Our method, general relativistic Hamiltonian Monte Carlo (GR-HMC), shares a deep connection with the theory of general relativity. That is, it "slows down" particles in a position-dependent fashion: particles will be slowed down more in regions with higher curvature. This is equivalent to using a smaller integration step size for high curvature regions of the target distribution in a physics-motivated manner.

GR-HMC relates the geometry (space) to the Hamiltonian dynamics (time) by the following kinetic energy

$$\tilde{K}_{\mathbf{G}}(\mathbf{q}, \mathbf{p}) = mc^2 \sqrt{\frac{\mathbf{p}^\top \mathbf{G}_\mathbf{q}^{-1} \mathbf{p}}{m^2 c^2} + 1} + \frac{1}{2} \log \det(\mathbf{G}_\mathbf{q}) \quad (8)$$

where $\mathbf{q}$ and $\mathbf{p}$ are the position and momentum variables, $\mathbf{G}_\mathbf{q}$ is a position-dependent metric and m, c are the rest mass and the speed of light as in (3).

We now show that (8) upper bounds norm of particle velocity

$$\mathbf{v}_\mathbf{G} := \nabla_\mathbf{q} H = \left(M_\mathbf{G}(\mathbf{q}, \mathbf{A}_\mathbf{q}\mathbf{p})\right)^{-1} \mathbf{G}_\mathbf{q}^{-1} \mathbf{p} \quad (9)$$

where $M_\mathbf{G}(\mathbf{q}, \mathbf{p}) = m\sqrt{\frac{\mathbf{p}^\top \mathbf{G}_\mathbf{q}^{-1} \mathbf{p}}{m^2 c^2} + 1}$ is the "general relativistic mass" with metric $\mathbf{G}$. For any $\mathbf{q}$, we have

$$M_\mathbf{G}(\mathbf{q}, \mathbf{A}_\mathbf{q}\mathbf{p}) = m\sqrt{\frac{\mathbf{p}^\top \mathbf{p}}{m^2 c^2} + 1} > m\sqrt{\frac{\mathbf{p}^\top \mathbf{p}}{m^2 c^2} + 0} = \frac{\|\mathbf{p}\|_2}{c},$$

where $\mathbf{A}_\mathbf{q}^\top \mathbf{A}_\mathbf{q} = \mathbf{G}_\mathbf{q}^{-1}$. The velocity norm is, therefore upper upper-bounded as $\|\mathbf{v}_\mathbf{G}\|_2 < \frac{\|\mathbf{A}_\mathbf{q}\mathbf{p}\|_2}{\|\mathbf{p}\|_2} c$. For a local geometry that corresponds to an identity matrix, the bound reduces to c. However, for a local geometry at some $\mathbf{q}$ that leads to a smaller $\|\mathbf{A}_\mathbf{q}\mathbf{p}\|_2$, the velocity norm will be reduced furthermore, which are regions with high curvature in general.

Sampling from a simulated Hamiltonian trajectory We have established the general relativistic Hamiltonian system by defining its kinetic energy, from which we can simulate numerical trajectories given the step size and the number of steps. To derive a complete HMC transition kernel, we still need to decide how to sample a phase point from simulated Hamiltonian trajectories. This work considers two options for sampling phrase points from a Hamiltonian trajectory. The first is to select the endpoint of a Hamiltonian trajectory and then apply a standard Metropolis-Hastings (MH) correction to decide whether to accept or reject it. The MH acceptance ratio defined as $a = \min(1, \exp(\Delta_H))$ where $\Delta_H = H(\mathbf{q}) - H(\mathbf{q}')$ is the difference in Hamiltonian energy of the current position $\mathbf{q}$ and the candidate position $\mathbf{q}'$. A non-negative Δ_H would lead to an acceptance of 1, which is desirable. The second is to perform multinomial sampling from the entire set of phase points on a Hamiltonian trajectory. It computes the energy differences of the initial point and all T points $(\mathbf{q}_1, \mathbf{q}_T)$ to construct an energy difference vector $\mathbf{\Delta}_H = [H(\mathbf{q}) - H(\mathbf{q}_1), \ldots, H(\mathbf{q}) - H(\mathbf{q}_T)]$ and sample from a multinomial distribution defined by that vector $\mathbf{\Delta}_H$ to pick the next state from the trajectory (Betancourt, 2018).

Physical interpretation Another interpretation of the slow-down effect of particle velocity in GR-HMC is "relativistic step size". Comparing the velocity of GR-HMC in (9) and that of RHMC in (5), the velocity of GR-HMC is scaled by $M_\mathbf{G}^{-1}(\mathbf{q}, \mathbf{A}_\mathbf{q}\mathbf{p})$. When used in the generalized leapfrog step (7), one can interpret it as changing the step size ε used by RHMC to "relativistic step size". Effectively, the larger the general relativistic mass is, the smaller the step size is. Table 3 in appendix A summarises the physical quantities discussed so far for all four HMC methods.

4.1. Illustration of relativistic Hamiltonian systems

We now give two illustrations to help understand the interaction between position and momentum variables. Figure 2 shows the general relativistic mass as a field for unit momentum variables at three selected different directions on a 2-dimensional Neal's funnel; we present only three directions due to space limit and provide plots with eight evenly spaced directions in figure 6 and figure 7 of appendix B. The relativistic field is a joint result from the interaction of the momentum as a vector and the local geometry as a matrix (*not only* its curvature, which is a scalar): the field is different for different momentum directions. Each of the three figures in each column has a different direction at $\theta = \frac{1}{4}\pi, \frac{0}{4}\pi, \frac{-1}{4}\pi$ that is not in parallel with nor orthogonal to the x-axis, and the fields are asymmetric along the x-axis even though the potential U is symmetric along the x-axis. Also, note how different the values are from the constant relativistic mass for SR-HMC (indicated by red arrows on the color bar): the slow-down effect is stronger for regions needing smaller time steps to achieve stable numerical integration.

Figure 3 shows the evolution of four key quantities, relativistic mass, norm of velocity, change of Hamiltonian energy and MH acceptance ratio, over the simulation of a set of Hamiltonian trajectories starting from the same position and momentum variables. While RHMC can explore the space better than HMC and SR-HMC, it suffers from frequent numeric errors that lead to smaller acceptance ratios. GR-HMC solves this issue by local speed upper bounds induced by its kinetic energy.

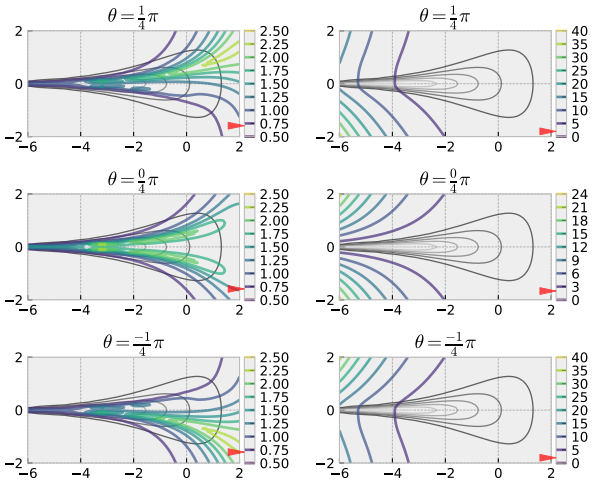


Figure 2: Contour plots of fields of general relativistic mass (left) and fields of local velocity upper bounds (right), both overlaid with the potential energy on 2D Neal's funnel. From top to bottom we have momentum $\mathbf{p} = r[\cos(\theta)\sin(\theta)]$ with $r = 1, \theta = \frac{1}{4}\pi, \frac{0}{4}\pi, \frac{-1}{4}\pi$ and $m = 0.5, c = 2.0$. Red arrows on the color bars indicate the values of the corresponding relativistic mass of SR-HMC, which is position-independent and 0.707 in all directions.

5. The Complete GR-HMC Algorithm

This section introduces the algorithmic details of a practical and computationally efficient implementation of GR-HMC. The use of our kinetic energy (8) introduces two technical challenges. First, obtaining exact samples of momentum variables in multivariate settings is difficult. While rejection sampling can be used, the efficiency quickly degrades with increased dimensionality. Second, numerical integration of the Hamiltonian dynamics requires accurate and efficient computation of $\frac{\partial H}{\partial \mathbf{q}}$ and $\frac{\partial H}{\partial \mathbf{p}}$. We address the first challenge by combining the Box Muller transform and uni-variate rejection sampling (section 5.1). The second challenge is solved by deriving compact forms of the two terms with cache-able intermediate computation (section 5.3). Finally, section 5.4 presents GR-HMC with diagonal Hessian approximations that are needed for high-dimensional problems.

5.1. Momentum sampling via Box Muller transform

Sampling from $\mathrm{pr}_{\mathbf{G}}(\mathbf{q}, \mathbf{p}) \propto -\exp\left(\tilde{K}_{\mathbf{G}}(\mathbf{q}, \mathbf{p})\right)$ for any $\mathbf{q}$ can be done by first sampling $\mathbf{p}$ from $\mathrm{pr}_{\mathbf{I}}(\mathbf{p}) \propto -\exp\left(\tilde{K}_{\mathbf{I}}(\mathbf{p})\right)$, the multivariate relativistic momentum in (3), and then applying an affine transformation as $\mathbf{A}_{\mathbf{q}}\mathbf{p}$, where $\mathbf{A}_{\mathbf{q}}^{\top}\mathbf{A}_{\mathbf{q}} = \mathbf{G}_{\mathbf{q}}^{-1}$[1]. However, sampling from $\mathrm{pr}_{\mathbf{I}}(\mathbf{p})$ in high dimensions is computationally challenging. While it is possible to use (adaptive) rejection sampling with a convex-hull as the proposal distribution, its statistical efficiency decreases quickly with the dimension of $\mathbf{p}$ increases because the rejection rate increases exponentially fast. We now show how to overcome this issue using the Box Muller transform.

We start with the description in a 2-dimensional case and then introduce the general form. Consider $\mathbf{p} = [p_1, p_2] \in \mathbb{R}^2$ sampled from $\mathrm{pr}_{\mathbf{I}}(\mathbf{p})$ and transforming $\mathbf{p}$ to the polar coordinates as $r = \|\mathbf{p}\|_2$ and $\theta = \arctan(\frac{p_1}{p_2})$. The cumulative density function of momentum $F(p_1, p_2)$ is equal to

$$F(r, \theta) = \int\int \exp\left(-\tilde{K}_{\mathbf{I}}(r)\right) r \mathrm{d}r\mathrm{d}\theta = F(r)F(\theta),$$

which implies that r is distributed as follow

$$r \sim \mathrm{pr}(r) \propto \exp\left(-\tilde{K}_{\mathbf{I}}(r)\right) r \tag{10}$$

(noting the extra term r in the end) and $\theta \sim \mathcal{U}(0, 2\pi)$. This means we can sample $\mathbf{p}$ by first sampling a uni-variate r using rejection sampling and the θ uniformly, and then transforming r, θ back to the original coordinates. This 2-dimensional case above can be generalized to d-dimensional using the spherical coordinate transformation:

$$\begin{aligned} p_1 &= r\cos(\theta_1) & & & p_{d-1} &= r\sin(\theta_1)\ldots\cos(\theta_{d-1}) \\ p_2 &= r\sin(\theta_1)\cos(\theta_2) & & \vdots & p_d &= r\sin(\theta_1)\ldots\sin(\theta_{d-1}) \end{aligned} \tag{11}$$

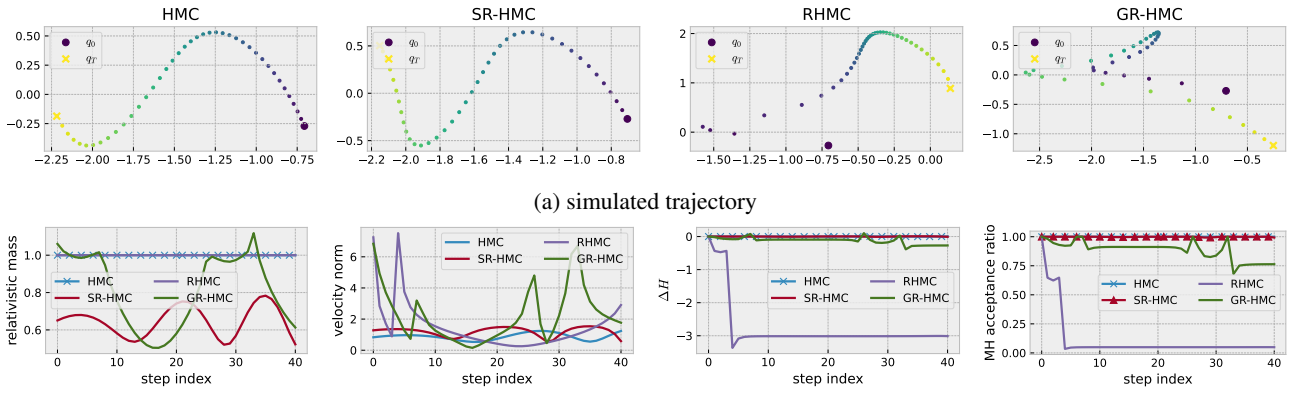
where $\theta_1, \ldots, \theta_{d-2} \sim \mathcal{U}(0, \pi)$ and $\theta_{d-1} \sim \mathcal{U}(0, 2\pi)$.

The steps above enable sampling $\mathrm{pr}_{\mathbf{I}}(\mathbf{p})$ using only a single uni-variate rejection sampling for r and independent uniform sampling for θ. Combining these steps with a linear transformation, we have algorithm 1 for sampling from multivariate relativistic momentum distribution $\mathrm{pr}(\mathbf{p})$.

5.2. Validity of GR-HMC

The correctness of the overall GR-HMC algorithm closely follows Girolami & Calderhead (2011). We only provide an brief informal justification of validity here. To start, let us recall the Gibbs view of HMC stated at the start of section 2, the overall correctness of our algorithm can be proved if each step is shown to be correct. Our paper only modifies Step 1 while using established, provable techniques for Step 2 (generalized leapfrog integration) and Step 3 (Metropolis or multinomial sampling). For Step 1, the only requirement for momentum distribution is symmetry, which is satisfied by the definition of (8).

[1]Note that the log-determinant term in (8) accounts for the change in normalization term of the corresponding momentum distribution due to an affine transformation with $\mathbf{A}_{\mathbf{q}}$.



(a) simulated trajectory

(b) relativistic mass (c) velocity norm / speed (d) Δ_H (e) MH acceptance ratio

Figure 3: Evolution of relativistic mass, norm of velocity, change of Hamiltonian energy and MH acceptance ratio (figure 3b–figure 3e) through Hamiltonian simulation (figure 3a). Compared with SR-HMC, GR-HMC has a larger velocity norm that helps exploration. Compared with RHMC, GR-HMC has small numerical errors, leading to higher acceptance rates.

Algorithm 1 Momentum sampling via Box Muller transform

Require: mass m, speed c, dimension d, current position $\mathbf{q}$
Ensure: $\mathbf{p} \in \mathbb{R}^d \sim \mathrm{pr}_{\mathbf{G}}(\mathbf{q}, \mathbf{p})$
1: sample r as in (10) using adaptive rejection sampling
2: **for** $i = 1, \ldots, d$ **do**
3: sample θ_i from $\mathcal{U}(0, \pi)$ if $i < d - 1$ else $\mathcal{U}(0, 2\pi)$
4: compute p_i according to the transformation in (11)
5: **end for**
6: $\mathbf{p_I} \leftarrow [p_1, \ldots, p_d]$
7: $\mathbf{p} \leftarrow \mathbf{A}_{\mathbf{q}}^{\top} \mathbf{p_I}$ where $\mathbf{A}_{\mathbf{q}}^{\top} \mathbf{A}_{\mathbf{q}} = \mathbf{G}_{\mathbf{q}}^{-1}$
Return $\mathbf{p}$

5.3. Efficient derivative computation during integration

To obtain computationally efficient analytical expressions for the derivative terms we need, we first derive them from the SoftAbs metric and apply the results from Betancourt (2013) to obtain cache-able forms.

We start by noticing that $\frac{\partial K}{\partial q_i}$ can be written as

$$\frac{\partial K}{\partial q_i} = M_{\mathbf{G}}^{-1}(\mathbf{q}, \mathbf{p}) \frac{1}{2} \mathbf{p}^{\top} \underbrace{\mathbf{G}_{\mathbf{q}}^{-1} \frac{\partial \mathbf{G}_{\mathbf{q}}}{\partial q_i} \mathbf{G}_{\mathbf{q}}^{-1}}_{\text{cacheable}} \mathbf{p} + \underbrace{\frac{1}{2} \operatorname{tr}\{\mathbf{G}_{\mathbf{q}}^{-1} \frac{\partial \mathbf{G}_{\mathbf{q}}}{\partial q_i}\}}_{\text{cacheable}}. \tag{12}$$

Importantly, Betancourt (2013) has established cache-able computation in the fixed-point iteration in (6) for the same position variable, noted by the two terms in (12) with under braces. Note that we have the same computation as for Riemannian HMC with SoftAbs metric while the first term is inversely scaled by the general relativistic mass $M_{\mathbf{G}}(\mathbf{q}, \mathbf{p})$.

Next, we apply the chain rule of multivariate calculus to obtain the derivative for the position variable update

$$\frac{\partial H}{\partial p_i} = \langle M_{\mathbf{G}}^{-1}(\mathbf{q}, \mathbf{A_q p})\, \mathbf{A_q p}, \{\mathbf{A_q}\}_{:,i} \rangle, \tag{13}$$

where $\{\cdot\}_{:,i}$ is a notation of taking i-th column of a matrix. This computation is straightforward and enjoys a compact form of $\nabla_{\mathbf{p}} H = M_{\mathbf{G}}^{-1}(\mathbf{q}, \mathbf{A_q p})\, \mathbf{A}_{\mathbf{q}}^{\top} \mathbf{A_q p}$ or simply $\nabla_{\mathbf{p}} H = M_{\mathbf{G}}^{-1}(\mathbf{q}, \mathbf{A_q p})\, \mathbf{G}_{\mathbf{q}}^{-1} \mathbf{p}$. Similarly, the form is similar to that of Riemannian HMC with SoftAbs metric with the only difference being an inverse scale of $M_{\mathbf{G}}(\mathbf{q}, \mathbf{A_q p})$.

5.4. Diagonal approximations

For HMC using the SoftAbs metric, the computation of the Hessian dominates the overall computational cost and scales cubically with the problem dimension. Therefore, to use GR-HMC in high-dimensional problems, it is more practical to use the SoftAbs metric with a Hessian approximation for which only the diagonal entries of the Hessian are computed; we do not use other approximations such as inner or outer products of the gradient as they are found to be inefficient or unstable in previous work (Betancourt, 2013).

To make the Hessian computation work on medium or high dimensional targets, we use the diagonal approximations $\hat{\mathbf{H}}$ following (Betancourt, 2013) as

$$\hat{\mathbf{H}} = \mathrm{diagm}(\mathbf{h}),$$

where $\mathbf{h} = \mathrm{diag}(\mathbf{H})$, diag takes in a matrix and returns its diagonal entries as a vector, and diagm takes in a vector and returns a matrix with the vector as the diagonal. The SoftAbs on $\hat{\mathbf{H}}$ can be computed as

$$\wr \hat{\mathbf{H}} \wr = \mathrm{diagm}\left(\mathbf{h} \odot \coth(\alpha \mathbf{h})\right),$$

where $\odot$ is the element-wise product and $\coth$ is also applied element-wisely. This reduces the time complexity from $O(d^3)$ to $O(d^2)$ where d is the dimension of the target.

Implementation-wise, we use a hybrid of forward-mode and reverse-mode automatic differentiation (AD). To compute the Hessian diagonal efficiently, we use the `diaghessian` function from `Zygote.jl`[2]. To compute the higher-order derivatives over Hessian required in (12), we use forward-mode AD over the `diaghessian` function via the standard `jacobian` function from `ForwardDiff.jl`.

6. Experiments

We implement all samplers studied in this paper using `AdvancedHMC.jl` (Xu et al., 2020).[3] Derivative implementation in (5.3) is tested by finite differentiation. Geweke tests (Geweke, 2004; Grosse & Duvenaud, 2014) are used to validate the correctness of samplers (detailed in appendix C). Appendix D lists default hyper-parameters used across experiments.

6.1. Stability and efficiency of Hamiltonian simulation

We first study the stability and efficiency of the Hamiltonian dynamics used in HMC, RHMC, SR-HMC and GR-HMC. In particular, we will show that relativistic momentum stabilizes numerical integration, reducing computation waste and improving acceptance rates. The use of Riemannian manifolds leads to longer travel distances, improving space exploration, which GR-HMC enjoys. For this study, we use the 2-dimensional Neal's funnel (Neal, 2011): we vary the step sizes used by the (generalized) leapfrog integration for Hamiltonian simulation and compare the stability and efficiency (definitions detailed below) across methods. To reduce the variance in comparisons, we ensure that the initial phase points are the same for all methods in each seeded simulation: we sample the initial position from a 2-dimensional standard normal distribution and the initial momentum as a unit Gaussian momentum. We simulate the trajectory for $L = 200$ leapfrog steps for all methods and repeat the simulation for 500 times.

Stability: fewer divergent simulations In our experiments, divergences in a Hamiltonian simulation are defined as situations where numerical errors occur or when the total energy differences exceed $10,000$ in the simulation. These divergences represent a waste of computation in HMC, as the entire simulated trajectory would be discarded. Table 1 shows the percentages of divergent simulation for all methods at five different step sizes.[4] GR-HMC is numerically more stable than RHMC due to the local upper bounds of the particle velocity, which are induced by the general relativistic momentum. We also confirm the finding from Lu et al. (2017) that global speed upper bounds in SR-HMC improve stability than HMC.

METHOD $\backslash \epsilon$	0.05	0.06	0.07	0.08	0.09	0.10
HMC	1.8%	2.2%	3.4%	3.6%	5.4%	5.4%
SR-HMC	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%
RHMC	2.0%	6.4%	13.2%	21.8%	35.4%	44.6%
GR-HMC	0.0%	0.0%	0.0%	0.0%	0.0%	1.8%

Table 1: Percentage of simulations with divergent integration (numeric errors or total energy differences $> 10,000$)

Efficiency: less correlation and more acceptance The correlation between adjacent states and the acceptance rates determines the efficiency of HMC's Hamiltonian simulation. Longer trajectory lengths can make samples less correlated, and reducing Hamiltonian energy errors increases acceptance rates. To quantitatively measure them, we use four metrics computed from a simulated trajectory.

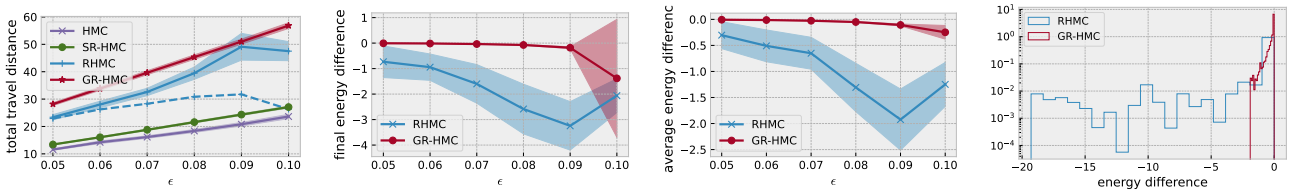
- The *total travel distance* measures how long in the position space the particle travels during Hamiltonian simulation. A large total travel distance is the prerequisite for efficient space exploration and less correlation.
- The *final energy difference*, Δ_H, measures the difference of Hamiltonian energy between the initial and the final state, clamped by 0, i.e. $\min(0, \Delta_H)$. It is used in the MH criterion to accept or reject a proposal with probability $\alpha = \min(1, \exp(\Delta_H))$; a larger (negative) energy difference indicates a high acceptance rate (a non-negative difference means 100% acceptance).
- The *energy difference per state* in $\boldsymbol{\Delta}_H$ is the difference of Hamiltonian energy between the initial one and all states. The *average energy difference* is an indication of the average quality of states used by multinomial trajectory sampling (section 4); better quality means closer to the perfect simulation.

Figure 4 shows the three metrics at 6 different increasing step sizes and the energy difference per state for $\varepsilon = 0.1$. First, we can see that both RMC and GR-HMC have longer total travel distances than HMC and SR-HMC, indicating the effectiveness of Riemannian manifolds at challenging targets like Neal's funnel. Next, GR-HMC consistently has longer total travel distances than RHMC for all step

[2] https://fluxml.ai/Zygote.jl/v0.6/utils/#Zygote.diaghessian

[3] Available at https://github.com/TuringLang/AdvancedHMC.jl

[4] Table 5 in appendix E.1 also provides the percentages calculated at intermediate lengths $L = 50, 100$ for inspection.



(a) total travel distance (b) final energy difference (c) average energy difference (d) energy difference per state

Figure 4: Efficiency metrics for HMC's numerical integration. Figure 4a, total travel distance (larger the better), tells how much space the trajectory explores; the dashed line is the product of the total travel distance and the percentage of non-divergent simulations. Figure 4b, final energy difference (larger the better), indicates the chance of the final point in a trajectory is accepted according to a Metropolis trajectory sampler and figure 4c, average energy difference (larger the better), indicates the average quality of proposed state from multinomial sampling. The error bars are plotted using 0.1 standard deviation for better presentation. Figure 4d, energy difference per state, is a zoom-in view of figure 4c at $\varepsilon = 0.1$.

sizes, likely because it can travel faster in regions with low curvature. The gaps become even larger when we consider the acceptance rates, shown by the dashed line, which is the product of the total travel distance and the percentage of non-divergent simulations; we do not plot that for GR-HMC as it has almost 100% acceptance (table 1). Due to the noticeable gap between using and not using Riemannian manifolds, we focus only on RHMC and GR-HMC in the next three plots. Third, GR-HMC consistently has smaller final (absolute) energy differences than RHMC for all step sizes. The gaps are large for most step sizes, with only $\varepsilon = 0.1$ having only overlap for the confidence interval (error bars). This suggests that GR-HMC would work better than RHMC with MH trajectory sampling. Finally, the average (absolute) energy differences are smaller for GR-HMC compared with RHMC. With the energy difference per state zoomed in for $\varepsilon = 0.1$, we can see while most of the values are around 0 for GR-HMC, the values for RHMC can spread up to -20. These together suggest that GR-HMC would work better than RHMC with multinomial trajectory sampling.

6.2. Sampling efficiency of HMC

After studying low-level efficiency metrics in trajectory simulation, we now focus on the overall sampling efficiency. We measure sampling efficiency using effective sample size (ESS) (averaged over all dimensions).

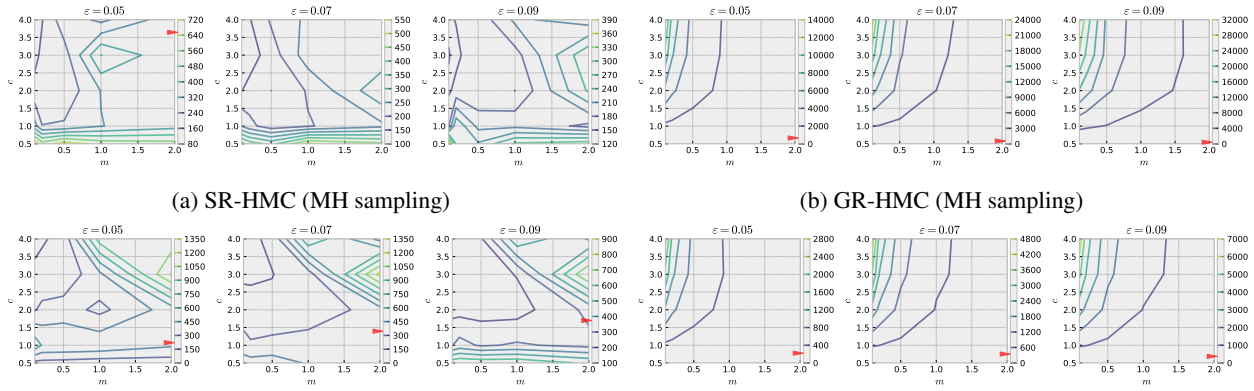
6.2.1. Effects of hyperparameters

We first study how the two key hyperparameters, m and c, affect the ESS using Neal's funnel in 10D. We vary $m = 0.1, 0.2, 0.5, 1.0, 2.0$, $c = 0.5, 1.0, 2.0, 3.0, 4.0$ for $\varepsilon = 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.15, 0.2, 0.25, 0.3$, with each configuration simulating 20 chains of 2,000 samples. For each step size, we make a contour plot for SR-HMC and GR-HMC, with the two axes representing different values of m and c. We annotate the color bar with the corresponding ESS of RHMC with the same step size. Figure 5 shows these contour plots for a selected range of step sizes that no sampler has very low acceptance rates.[5] Focusing on the first row with MH sampling, as it can be seen, GR-HMC generally achieves much better ESS than SR-HMC and could have better efficiency with proper choices of m and c (majority of the regions) than RHMC (red arrows). Note that some color bars of SR-HMC are not annotated by red arrows because the ESS for all values of m and c we experiment are lower than that of RHMC.

6.2.2. Which trajectory sampling method benefits more: MH or multinomial?

This section compares two trajectory sampling methods, MH and multinomial. Multinomial sampling is commonly used in practice because it is more robust against numerical errors in simulation. It can accept intermediate, high-quality points and rarely "rejects" a trajectory (while rejection is equivalent to sampling the initial point). Figure 5c and figure 5d show the ESS contour plots with multinomial sampling. First, GR-HMC improves upon SR-HMC for both trajectory sampling methods. This confirms the benefits of using the general relativistic momentum proposed in this paper. Second, SR-HMC with multinomial sampling is generally better than that with MH sampling. The common practice of using multinomial sampling also supports this finding. Third, the relative improvement of GR-HMC to SR-HMC is smaller when using multinomial sampling than MH. Most of this is because the multinomial sampling method is more robust to numerical errors, making improvements in numerical simulation less visible. Nevertheless, general relativistic momentum is still beneficial for both methods.

[5] We check the performance for dimension-wise special relativistic momentum (section 2.2), for which the same set plots are provided figure 10 in appendix E.2. We also find that the dimension-wise variant works better (Lu et al., 2017), but GR-HMC still performs much better than the dimension-wise SR-HMC.



(a) SR-HMC (MH sampling)

(b) GR-HMC (MH sampling)

(c) SR-HMC (multinomial sampling)

(d) GR-HMC (multinomial sampling)

Figure 5: Effects of m and c on ESS for SR-HMC and GR-HMC with different ε. Red arrows on the color bars indicate the corresponding ESS of RHMC. The trajectory sampling method used is MH for top figures and multinomial for bottom ones.

METHOD $\backslash \epsilon$	0.02	0.04	0.06	0.08	0.1
SR-HMC	18.3	14.8	15.6	14.1	13.3
RHMC	19.3	52.3	105.8	179.2	**258.7**
GR-HMC	**20.6**	**80.7**	**147.5**	**210.2**	217.8

(a) HBLR w/ 1,000 samples ($m = 0.2, c = 4.0$ and $\alpha = 1 \times 10^6$)

METHOD $\backslash \epsilon$	0.1	0.2	0.3	0.4	0.5
SR-HMC	**272.8**	3968.0	3794.3	918.4	**172.4**
RHMC	272.0	1490.6	4756.8	4.7	3.0
GR-HMC	160.9	**4446.3**	**4764.1**	**4456.7**	127.1

(b) logGCPP w/ 500 samples ($m = 0.5, c = 2.0$ and $\alpha = 1 \times 10^6$)

Table 2: Sample efficiency (measured by ESS) with different step sizes on two real-world problems.

6.2.3. REAL-WORLD DATASETS

To validate if the improved sampling efficiency of GR-HMC is useful in practice, we conduct experiments on two real-world problems, a hierarchical Bayesian logistic regression (HBLR) model and a log-Gaussian Cox point process (log-GCPP) model, that are previously used to benchmark HMC algorithms (Heng & Jacob, 2019; Xu et al., 2021). The HBLR problem has a dimensionality of 26, and the log-GCPP problem has a dimensionality of 64. Due to the space limitations, we deter the details of these problems to appendix F. Running HMC on Riemannian manifolds with full Hessian is not computationally efficient due to the cubic complexity; therefore, for these two problems, we use the diagonal approximation in section 5.4. For each problem, we sweep the step sizes and report GR-HMC with a specified combination of m and c as well as baseline methods; we do not sweep m and c as in the previous section because of computation limitations. The results average from 3 runs are given in table 2. As it can be seen, GR-HMC has the best ESS for most of the step sizes except (i) RHMC being the best for $\varepsilon = 0.1$ on HBLR and (ii) SR-HMC being the best for $\varepsilon = 0.1$ and $\varepsilon = 0.5$ on logGCPP. For point (i), by inspecting the average acceptance rates, we find that that of RHMC does not decrease quickly with increased step sizes for HBLR, which is likely because the posterior is not too complex. As a result, RHMC does not suffer much from the numeric errors with large step sizes, leading to its best performance with $\varepsilon = 0.1$. For point (ii), the step sizes where SR-HMC excels are notably distant from the optimal values and lack practical benefits; hence, one should avoid using these step sizes regardless. GR-HMC achieves notable ESS across a broad spectrum of step sizes, showing its potential as a default HMC sampler.

7. Conclusion and Future Work

This paper shows how to stabilize the numerical integration in Riemannian HMC by constructing relativistic momentum distributions with position-dependent metrics, leading to a novel HMC sampling method, GR-HMC. It has interesting connections to the theory of general relativity and substantially improves the stability of HMC methods on Riemannian manifolds. It paves the way for the practical use of Riemannian HMC by solving one of the two most important issues in practice: computational cost and numerical stability. Future work would focus on improving their computational complexity to make them even more practical beyond section 5.4. For that purpose, alternative methods to approximate Hessian based on iterative, first-order gradient-only approaches have been proven to be effective in optimization, e.g. limited-Memory BFGS (L-BFGS) (Nocedal & Wright, 1999) and MCMC sampling, e.g. HMC-BFGS (Zhang & Sutton, 2011).

Impact Statement

This paper presents work that aims to advance the field of Bayesian Inference. Our work has many potential societal consequences, none of which we feel must be specifically highlighted here.

References

Asuncion, A. and Newman, D. UCI machine learning repository, 2007.

Barber, D. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo, July 2018.

Betancourt, M. and Stein, L. C. The Geometry of Hamiltonian Monte Carlo, December 2011.

Betancourt, M. J. A General Metric for Riemannian Manifold Hamiltonian Monte Carlo. 8085:327–334, 2013. doi: 10.1007/978-3-642-40020-9_35. URL http://arxiv.org/abs/1212.4693.

Bou-Rabee, N. and Sanz-Serna, J. M. Randomized Hamiltonian Monte Carlo. *The Annals of Applied Probability*, 27(4):2159–2194, August 2017. ISSN 1050-5164, 2168-8737. doi: 10.1214/16-AAP1255.

Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M. A., Guo, J., Li, P., and Riddell, A. Stan: A Probabilistic Programming Language. *Journal of statistical software*, 76:1, 2017. ISSN 1548-7660. doi: 10.18637/jss.v076.i01.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, September 1987. ISSN 0370-2693. doi: 10.1016/0370-2693(87)91197-X.

Ge, H., Xu, K., and Ghahramani, Z. Turing: A Language for Flexible Probabilistic Inference. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pp. 1682–1690. PMLR, March 2018.

Geweke, J. Getting It Right: Joint Distribution Tests of Posterior Simulators. *Journal of the American Statistical Association*, 99(467):799–804, 2004. ISSN 0162-1459.

Girolami, M. and Calderhead, B. Riemann manifold langevin and hamiltonian monte carlo methods. *J. R. Stat. Soc. Series B Stat. Methodol.*, 73(2):123–214, March 2011. URL https://onlinelibrary.wiley.com/doi/10.1111/j.1467-9868.2010.00765.x.

Grosse, R. B. and Duvenaud, D. K. Testing MCMC code. https://arxiv.org/abs/1412.5218v1, December 2014.

Heng, J. and Jacob, P. E. Unbiased Hamiltonian Monte Carlo with couplings. *Biometrika*, 106(2):287–302, June 2019. ISSN 0006-3444, 1464-3510. doi: 10.1093/biomet/asy074.

Hoffman, M. D. and Gelman, A. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. https://arxiv.org/abs/1111.4246v1, November 2011.

Jha, J., Hashemi, M., Vattikonda, A. N., Wang, H., and Jirsa, V. Fully bayesian estimation of virtual brain parameters with self-tuning hamiltonian monte carlo. *Machine Learning: Science and Technology*, 3(3):035016, 2022.

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armananzas, R., Santafé, G., Pérez, A., et al. Machine learning in bioinformatics. *Briefings in bioinformatics*, 7(1):86–112, 2006.

Li, C., Chen, C., Carlson, D., and Carin, L. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. https://arxiv.org/abs/1512.07666v1, December 2015.

Linderman, S. W., Johnson, M. J., Wilson, M. A., and Chen, Z. A bayesian nonparametric approach for uncovering rat hippocampal population codes during spatial navigation. *Journal of neuroscience methods*, 263:36–47, 2016.

Livingstone, S., Faulkner, M. F., and Roberts, G. O. Kinetic energy choice in Hamiltonian/hybrid Monte Carlo. https://arxiv.org/abs/1706.02649v3, June 2017.

Lu, X., Perrone, V., Hasenclever, L., Teh, Y. W., and Vollmer, S. Relativistic Monte Carlo. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1236–1245. PMLR, April 2017.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. Log Gaussian Cox Processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998. ISSN 0303-6898.

Neal, R. M. *MCMC Using Hamiltonian Dynamics*. May 2011. doi: 10.1201/b10905.

Nocedal, J. and Wright, S. J. *Numerical optimization*. Springer, 1999.

Patterson, S. and Teh, Y. W. Stochastic Gradient Riemannian Langevin Dynamics on the Probability Simplex. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

Peng, S., Wang, G., and Xie, D. Social influence analysis in social networking big data: Opportunities and challenges. *IEEE network*, 31(1):11–17, 2016.

Sengupta, B., Friston, K. J., and Penny, W. D. Gradient-based mcmc samplers for dynamic causal modelling. *NeuroImage*, 125:1107–1118, 2016.

Whalley, P. A., Paulin, D., and Leimkuhler, B. Randomized Time Riemannian Manifold Hamiltonian Monte Carlo, August 2022.

Xu, K., Ge, H., Tebbutt, W., Tarek, M., Trapp, M., and Ghahramani, Z. AdvancedHMC.jl: A robust, modular and efficient implementation of advanced HMC algorithms. In *Proceedings of The 2nd Symposium on Advances in Approximate Bayesian Inference*, pp. 1–10. PMLR, February 2020.

Xu, K., Fjelde, T. E., Sutton, C., and Ge, H. Couplings for Multinomial Hamiltonian Monte Carlo. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, pp. 3646–3654. PMLR, March 2021.

Zhang, Y. and Sutton, C. Quasi-Newton Methods for Markov Chain Monte Carlo. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011.

Zhang, Y., Wang, X., Chen, C., Henao, R., Fan, K., and Carin, L. Towards Unifying Hamiltonian Monte Carlo and Slice Sampling. https://arxiv.org/abs/1602.07800v5, February 2016.

quantity	HMC	SR-HMC	RHMC	GR-HMC
kinetic energy	$\frac{1}{2}\mathbf{p}^\top\mathbf{p}$	$mc^2\sqrt{\frac{\mathbf{p}^\top\mathbf{p}}{m^2c^2}+1}$	$\frac{1}{2}\mathbf{p}^\top\mathbf{G}_\mathbf{q}^{-1}\mathbf{p}$	$mc^2\sqrt{\frac{\mathbf{p}^\top\mathbf{G}_\mathbf{q}^{-1}\mathbf{p}}{m^2c^2}+1}$
relativistic mass	–	$m\sqrt{\frac{\mathbf{p}^\top\mathbf{p}}{m^2c^2}+1} =: M_\mathbf{I}(\mathbf{p})$	–	$m\sqrt{\frac{\mathbf{p}^\top\mathbf{G}_\mathbf{q}^{-1}\mathbf{p}}{m^2c^2}+1} =: M_\mathbf{G}(\mathbf{q},\mathbf{p})$
velocity	$\mathbf{p}$	$\frac{1}{M_\mathbf{I}(\mathbf{p})}\mathbf{p}$	$\mathbf{G}_\mathbf{q}^{-1}\mathbf{p} =: \mathbf{p}'$	$\frac{1}{M_\mathbf{G}(\mathbf{q},\mathbf{A}_\mathbf{q}\mathbf{p})}\mathbf{G}_\mathbf{q}^{-1}\mathbf{p} = \frac{1}{M_\mathbf{I}(\mathbf{p}')}\mathbf{p}'$
velocity upper bound	–	c	–	$\frac{\\|\mathbf{A}_\mathbf{q}\mathbf{p}\\|_2}{\\|\mathbf{p}\\|_2}c$
effective step size	–	$\frac{1}{M_\mathbf{I}(\mathbf{p})}\varepsilon$	–	$\frac{1}{M_\mathbf{G}(\mathbf{q},\mathbf{A}_\mathbf{q}\mathbf{p})}\varepsilon$

Table 3: Physical interpretation for Hamiltonian quantities. Here $\mathbf{A}_\mathbf{q}^\top\mathbf{A}_\mathbf{q} = \mathbf{G}_\mathbf{q}^{-1}$ and $\mathbf{p}' = \mathbf{A}_\mathbf{q}\mathbf{p}$.

A. Additional Discussion on Physical Interpretation

Connection to the theory of general relativity The theory of general relativity, formulated by Einstein, is a fundamental theory in physics that describes how matter and energy influence the fabric of spacetime, resulting in the curvature of this four-dimensional manifold. In this theory, space and time are intertwined, forming a dynamic interplay known as the spacetime continuum. The Hamiltonian system with our proposed kinetic energy has a similar spacetime interaction through the position-dependent metric, leading to position-dependent velocity upper bounds. Note while this does not induce a global upper bound of the velocity strictly as in the general relativity theory, we can introduce extra assumptions on the local concavity of the potential energy and its Lipschitz constant, and obtain a global upper bound for the system.

Quantities in physical interpretation In addition to the high-level connections, many quantities studied in the paper have direct physical interpretations. Table 3 provides a summary of the physical quantities discussed for all four HMC methods.

B. Additional Illustrations

Figure 6 (relativistic mass) and figure 7 (speed upper bounds) provide a complete set of plots with eight evenly spaced directions (surrounding plots in each) as well as the density of the target as a standalone plot (middle plot in each), which are only partially displayed in figure 2 of section B due to space limitations.

C. Geweke tests for implementation correctness validation

Geweke tests (Geweke, 2004; Grosse & Duvenaud, 2014) is a standard method to validate the overall correctness of MCMC implementation, i.e. integrated tests. To perform Geweke tests, one requires a generative model with latent variables z and data/observations x: $p(z, x) = p(z)p(x \mid z)$ where $p(z)$ is the prior and $p(x \mid z)$ is the likelihood.

The main idea behind Geweke tests is that there are two equivalent ways to sample from the joint $p(z, x)$:

1. marginal-conditional simulator:
   (a) sample $z' \sim p(z)$ and
   (b) sample $x' \sim p(x \mid z = z')$
2. successive-conditional simulator:
   (a) given a sample x^* (e.g. x' from the marginal-conditional simulator or x'' the successive-conditional simulator of the previous round), sample $z'' \sim p(z \mid x = x^*)$ using MCMC samplers and
   (b) sample $x'' \sim p(x \mid z = z')$

We then use the equivalence of these simulators to perform the Geweke tests. These two simulators are run iteratively to produce two sets of N samples from the joint: $\mathcal{D}_{\text{fwd}} = \{(z'_i, x'_i)\}_{i=1}^N$ and $\mathcal{D}_{\text{bwd}} = \{(z''_i, x''_i)\}_{i=1}^N$. These two sets of samples

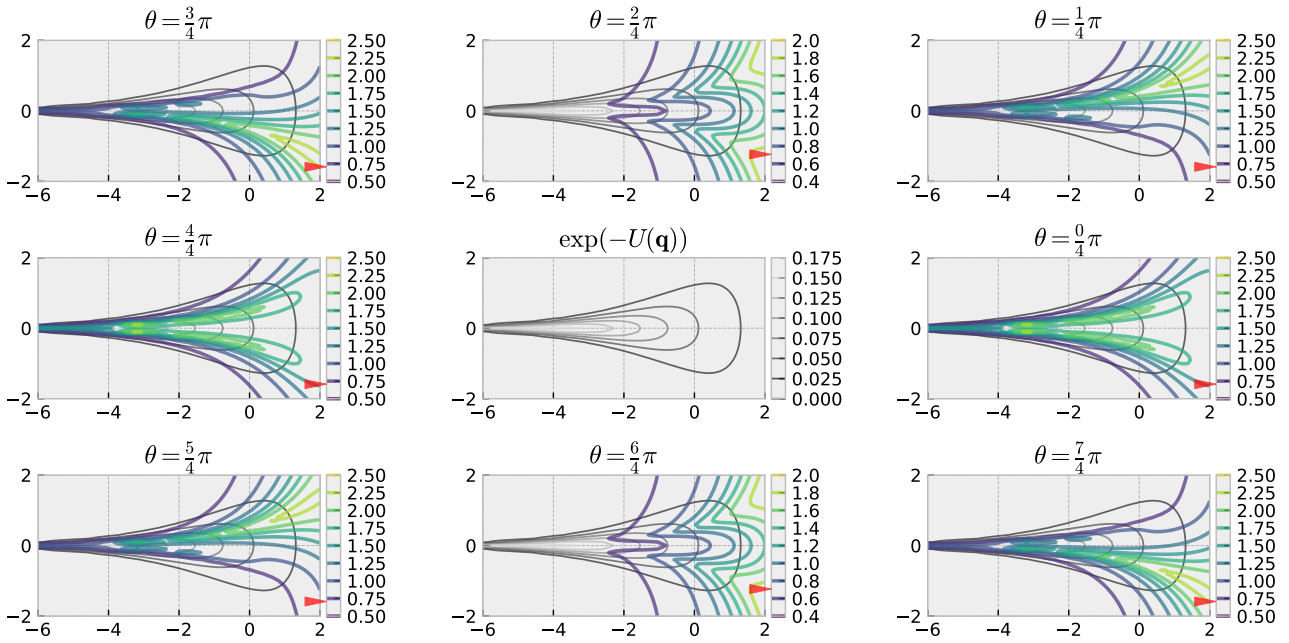


Figure 6: Contour plots of the potential energy (middle) and the general relativistic mass overlaid (the rest surrounding) with $m = 0.5, c = 2.0$ for $\mathbf{p} = r[\cos(\theta)\sin(\theta)]$ ($r = 1, \theta = \frac{0}{4}\pi, \ldots, \frac{7}{4}\pi$) on 2D Neal's funnel. Red arrows indicate the values on the color bars for the corresponding relativistic mass of SR-HMC, which is position-independent and constantly 0.707 in all directions.

are then used to produce the quantile-quantile plot (Q-Q plot) to check the sampler's correctness: since the two sets of samples should follow the same distribution for a correct sampler, the ideal Q-Q plot should be a diagonal line. We use the `MCMCDebugging.jl` package with a 3-D latent funnel model in the Turing language (Ge et al., 2018) to perform the Geweke tests with $N = 500$ for all samplers studied. The hyper-parameters, number of iterations M, leapfrog steps L, mass m, speed c, step size ε, fixed-point iteration steps n, scale of identity matrix added to Hessian λ and SoftAbs parameter α, used for each sampler are

- HMC: $M = 200, \varepsilon = 0.15, L = 16$
- SR-HMC: $M = 200, \varepsilon = 0.1, L = 16, m = 0.2, c = 20.0$
- RHMC: $M = 200, \varepsilon = 0.025, L = 16, n = 6, \lambda = 1 \times 10^{-2}, \alpha = 20.0$
- GR-HMC: $M = 200, \varepsilon = 0.05, L = 16, n = 6, \lambda = 1 \times 10^{-2}, \alpha = 20.0, m = 0.2, c = 4.0$

The probabilistic program of the generative model is given in figure 8 and the Q-Q plots for all samplers are given in figure 9. As it can be seen, all samplers pass the test by showing Q-Q plots closer to the ideal one.

D. Default Hyper-parameters Used Across All Experiments

Other than the hyper-parameters we studied in the paper, such as step size ε, mass m, etc., some other important hyper-parameters used in HMC are not specified in the main text, mostly due to limited space and the need to improve the presentation of the paper. For completeness, we enumerate them all in table 4.

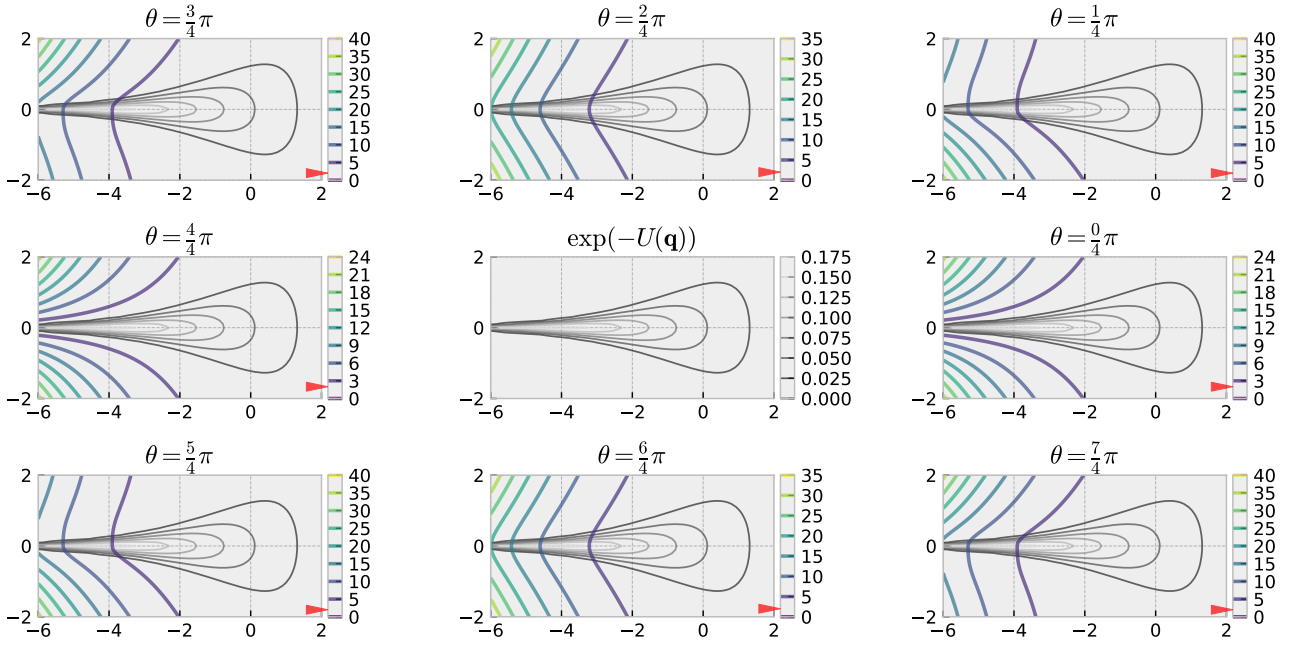


Figure 7: Contour plots of the potential energy (middle) and the local velocity upper bounds overlaid (the rest surrounding) with $m = 0.5, c = 2.0$ for $\mathbf{p} = r[\cos(\theta)\sin(\theta)]$ ($r = 1, \theta = \frac{0}{4}\pi, \ldots, \frac{7}{4}\pi$) on 2D Neal's funnel. Red arrows indicate the values on the color bars for the corresponding relativistic mass of SR-HMC, which is position-independent and constantly 0.707 in all directions.

parameter name	value	comment
the number of leapfrog steps	8	
the number of fixed-point iterations	6	
scale of identity matrix added to Hessian (λ)	1×10^{-2}	we use $\mathbf{H} + \lambda\mathbf{I}$ to regularize the Hessian
initial position distribution	$\mathcal{U}(-1, 1)$	this follows Betancourt (2013)

Table 4: Common hyper-parameters used across experiments in section 6.

E. Additional Results

E.1. Complete results for stability of numerical integration

As a complementary to table 1 in section 6.1, table 5 provides the percentages of divergent simulation for all methods at intermediate inspection steps $L = 50, 100$ and the full steps 200.

E.2. Contour plots of ESS for dimension-wise special relativistic momentum

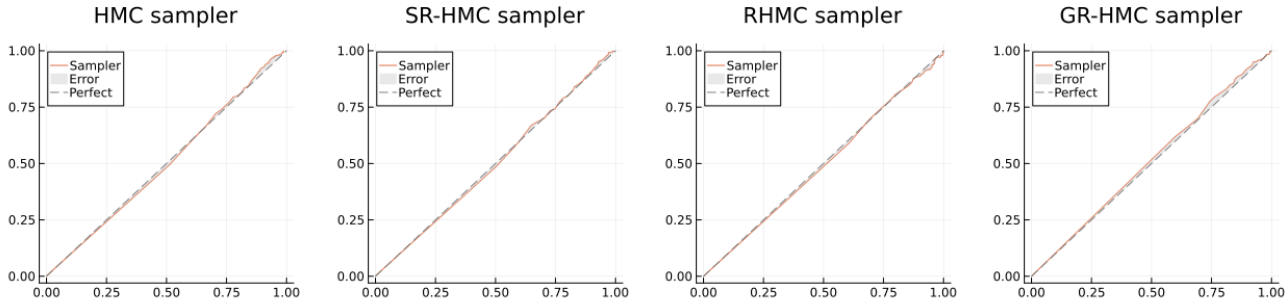
Figure 10 provides the same set of contour plots over different step sizes studied in section 6.2.1 for dimension-wise relativistic HMC by varying m, c.

F. Details of Real-World Problems

Here, we provide details on the real-world problems used in section 6.2.3. To prepare the datasets, we follow the pre-processing steps in (Heng & Jacob, 2019) for the German credit dataset (Asuncion & Newman, 2007) and the Finnish pine saplings dataset (Møller et al., 1998) used in logistic regression and log-Gaussian Cox point process respectively.

```julia
@model function TuringFunnel(θ=missing, x=missing)
    if ismissing(θ)
        θ = Vector(undef, 2)
    end
    θ[1] ~ Normal(0, 3)
    s = exp(θ[1] / 2)
    θ[2] ~ Normal(0, s)
    x ~ Normal(0, s)
    return θ, x
end
```

Figure 8: Probabilistic program of the generative model used in Geweke tests



(a) HMC (0.0117) (b) SR-HMC (0.0113) (c) RHMC (0.0109) (d) GR-HMC (0.00622)

Figure 9: Q-Q plots from Geweke tests for all samplers. The number in each sub-figure next to the sampler name is the quantile error, measuring the difference from the ideal line.

Hierarchical Bayesian logistic regression We use the first 100 data points in the German credit dataset, resulting in a $\mathbb{R}^{24\times 100}$ design matrix. Denote an Exponential distribution with rate λ as $\mathcal{E}(\lambda)$. Given data $\{(\mathbf{x}^i, y^i)\}_{i=1}^{100}$ where $\mathbf{x}^i$ is the features and $y^i \in \{0, 1\}$ is the binary responses, the hierarchical Bayesian logistic regression follows the following generative process:

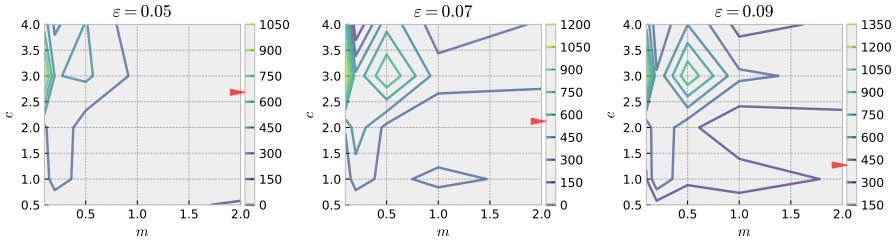
$$\begin{aligned} s^2 &\sim \mathcal{E}(\lambda) \\ a &\sim \mathcal{N}(0, s^2) \\ b_d &\sim \mathcal{N}(0, s^2) \qquad \text{for} \quad d = 1, \ldots, 24 \\ y^i &\sim \mathcal{B}(\sigma\left(\mathbf{b}^\top \mathbf{x}^i + a\right)) \quad \text{for} \quad i = 1, \ldots, 100 \end{aligned},$$

where the variance $s^2 \in \mathbb{R}^+$, the intercept $a \in \mathbb{R}$ and the coefficients $b \in \mathbb{R}^{24}$, giving a total dimension $d = 26$.

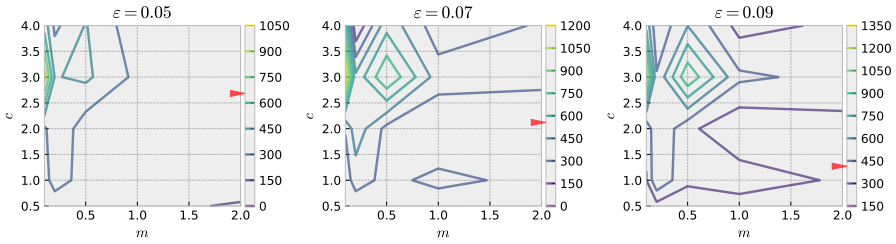
Log-Gaussian Cox point process Firstly, the plot of the forest is discretized into an $n \times n$ grid. For $i \in \{1, \ldots, n\}^2$, the number of points in each grid cell $y_i \in \mathbb{N}$ is assumed to be conditionally independent given a latent intensity variable Λ_i and follows a Poisson distribution with mean $a\Lambda_i$, $\mathcal{P}(a\Lambda_i)$, where $a = n^{-2}$ is the area of each cell. We denote the logarithm of Λ as X and put a Gaussian process prior with mean $\mu \in \mathbb{R}$ and exponential covariance function $\Sigma_{i,j} = s^2 \exp\left(-|i - j|/(nb)\right)$ on it, where s^2, b and μ are hyperparameters. The generative process of the number of grid cell points follows $X \sim \mathcal{GP}(\mu, \Sigma),\ \forall\, i \in \{1, \ldots, n\}^2 : \Lambda_i = \exp(X_i),\ y_i \sim \mathcal{P}(a\Lambda_i)$. Following (Møller et al., 1998), we use a dataset of 126 Scot pine saplings in a natural forest in Finland, and adapt the parameters $s^2 = 1.91$, $b = 1/33$ and $\mu = \log(126) - s^2/2$. We use $n = 8$ for discretization, leading to a target of dimension 64.

ϵ	0.05			0.06			0.07			0.08			0.09			0.10		
L	50	100	200	50	100	200	50	100	200	50	100	200	50	100	200	50	100	200
HMC	0.0	1.0	0.5	0.0	2.0	1.0	3.9	4.0	3.0	3.9	4.0	3.0	3.9	5.0	5.0	5.9	7.9	5.5
SR-HMC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
RHMC	2.0	1.0	2.0	7.8	7.9	6.5	13.7	11.9	11.9	25.5	23.8	22.9	37.3	33.7	34.3	45.1	43.6	44.8
GR-HMC	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.9	3.0	2.5

Table 5: Percentage (%) of simulations with divergent leapfrog integration (numeric errors or total energy differences $> 10,000$) at $L = 50, 100, 200$.



(a) MH sampling



(b) multinomial sampling

Figure 10: Effects of m and c on ESS for dimension-wise SR-HMC with different ε. Red arrows on the color bars indicate the corresponding ESS of RHMC. Subfigures are for two different trajectory sampling methods.