Geometric No-U-Turn Samplers: Concepts and Evaluation

Bernardo Williams BERNARDO.WILLIAMSMORENO@HELSINKI.FI
Hanlin Yu HANLIN.YU@HELSINKI.FI
Marcelo Hartmann MARCELO.HARTMANN@HELSINKI.FI
Arto Klami ARTO.KLAMI@HELSINKI.FI
Department of Computer Science, University of Helsinki

Editors: J.H.P. Kwisthout & S. Renooij

Abstract

We enhance geometric Markov Chain Monte Carlo methods, in particular making them easier to use by providing better tools for choosing the metric and various tuning parameters. We extend the No-U-Turn criterion for automatic choice of integration length for Lagrangian Monte Carlo and propose a modification to the computationally efficient Monge metric, as well as summarizing several previously proposed metric choices. Through extensive experimentation, including synthetic examples and posteriordb benchmarks, we demonstrate that Riemannian metrics can outperform Euclidean counterparts, particularly in scenarios with high curvature, while highlighting how the optimal choice of metric is problem-specific.

Keywords: Markov Chain Monte Carlo, Hamiltonian Monte Carlo, Riemannian Geometry.

1. Introduction

Outside of special cases, Bayesian inference for parameters of a graphical model requires approximations. Markov Chain Monte Carlo (MCMC) is one of the central techniques, being general method applicable for arbitrary joint densities. However, the practical efficiency and accuracy of MCMC depend on several non-trivial choices, ranging from the design of the proposals to the choice of several tuning parameters, such as integration length and stopping criteria in samplers, leveraging Hamiltonian dynamics and eventually also to more fundamental choices like the assumed metric for the parameter space.

We consider graphical models with continuous variables and differentiable joint density, such as Gaussian graphical models, various latent variable models, and in general a broad range of statistical models described as probabilistic programs, e.g. in the Stan syntax (Carpenter et al., 2017). That is, we assume gradients of the joint density are available. Efficient samplers have been designed around Hamiltonian (Neal et al., 2011) or Lagrangian (Lan et al., 2015) dynamics, where the parameter space is augmented by randomly proposed momentum or velocity and sampling is done by numerical integration of the dynamics. While this offers clear computational advantages over random-walk MCMC, the samplers often come with additional tuning parameters that need to be chosen. One practical solution to these challenges is the No-U-Turn (NUTS) sampler by Hoffman and Gelman (2014), which can automatically determine the integration time by monitoring when the integration trajectory curves too much and also offers practical heuristics for selecting the step length.

The performance of samplers can be controlled by modifying the assumed metric for the parameter space. In the simplest case, this corresponds to re-scaling the space by a static matrix, a metric tensor, but by making the scaling position-dependent we can improve the sampling performance, especially for complex distributions. This family of methods is called geometric or Riemannian MCMC, first introduced in Girolami and Calderhead (2011), with several follow-up works providing alternative formulations (Xifara et al., 2014; Lan et al., 2015; Cobb et al., 2019; Hartmann et al., 2022). Despite theoretical advantages, the practical use of geometric MCMC methods has been limited. This can be attributed to several reasons; the per-iteration cost of the samplers is higher due to requiring repeated inversions of the metric tensor, the choice of the best metric is far from obvious, and in many cases the sampler requires additional tuning parameters that are difficult to choose, even beyond the ones in Euclidean samplers. The additional computational burden, has to some extent, been resolved by recent works proposing metrics that are efficient to compute; for instance, Hartmann et al. (2022) proposed the Monge metric that avoids full-matrix operations, and Li et al. (2016) and Yu et al. (2023) showed that Riemannian MCMC methods can be scaled efficiently even for posterior analysis of neural networks. However, the choice of the optimal metric remains elusive, and it is difficult to use the samplers in practice due to the various tuning parameters.

We work towards solving these issues. We outline alternative metrics for geometric MCMC samplers, covering both existing metrics (Girolami and Calderhead, 2011; Hartmann et al., 2022; Betancourt, 2013a) and a new variant, and then show how the adaptive mechanism that NUTS uses for determining integration length can be extended for geometric methods, building partly on Betancourt (2013b). Although we introduce both a new metric variant and a new alternative stopping criterion, we note that the main goal of this work is not to introduce new algorithms as such, but to proceed towards wider use of the methods in practical applications. We do this primarily via an empirical comparison of the alternative approaches, in an attempt to shed light on when and how geometric methods should be used. In previous works, the validation of geometric MCMC methods has been limited to isolated cases, which makes it difficult to assess the value of Riemannian metrics in practice. Instead, we evaluate a range of alternatives by exhaustively covering the Cartesian product over the various choices, considering in total close to 300 cases. We draw a few conclusions based on the general trends and make the full results available for further analysis. We observe that it is usually possible to improve over the Euclidean metric but the optimal choice of the metric depends on the problem, and that the specific choice of the stopping criterion does not appear to be critical.

2. Preliminaries

Let us denote by $\boldsymbol{\theta} \in \mathbb{R}^D$ the vector of parameters of interest, with a prior distribution $p(\boldsymbol{\theta})$ and the likelihood $p(\boldsymbol{y} \,|\, \boldsymbol{\theta})$ for some observed data $\boldsymbol{y} = \{y_n\}_{n=1}^N$, here assumed i.i.d for simplicity. Our goal is to sample from the posterior distribution, $p(\boldsymbol{\theta} \,|\, \boldsymbol{y}) = p(\boldsymbol{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})/Z$, where $Z = \int p(\boldsymbol{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})\mathrm{d}\,\boldsymbol{\theta}$ is a normalization constant which does not depend on $\boldsymbol{\theta}$. From now on, we denote the target distribution by $\pi(\boldsymbol{\theta}) = p(\boldsymbol{y} \,|\, \boldsymbol{\theta})p(\boldsymbol{\theta})$ and the log target distribution by $\ell(\boldsymbol{\theta}) = \log \pi(\boldsymbol{\theta})$. We assume both are differentiable with respect to $\boldsymbol{\theta}$.

2.1. Riemannian MCMC

Random walk Monte Carlo methods are theoretically sufficient for solving the posterior inference problem, but today most practical samplers leverage on gradient information to reduce the correlation between the samples (Neal et al., 2011; Betancourt, 2017). Furthermore, treating distances in the parameter space from the perspective of a suitably chosen Riemannian metric has been shown to help explore difficult or pathological geometries (Girolami and Calderhead, 2011). In the following, we present the samplers directly from the perspective of the more general Riemannian formulation, explaining the commonly assumed Euclidean metric as a special case when necessary.

Riemannian Manifold Hamiltonian Monte Carlo (RMHMC) The Hamiltonian dynamics define a trajectory generated by a system of differential equations based on the joint distribution of the log-target distribution $\ell(\boldsymbol{\theta})$ and a D-dimensional auxiliary random variable $\boldsymbol{p}$, interpreted as momentum. The distribution of the auxiliary variable is often chosen as Gaussian $\boldsymbol{p} \,|\, \boldsymbol{\theta} \sim \mathcal{N}(0, \boldsymbol{G}(\boldsymbol{\theta}))$. This is now presented directly in the Riemannian formulation, where $\boldsymbol{G}(\boldsymbol{\theta})$ is a general Riemannian metric. The Hamiltonian is a function of the joint density $p(\boldsymbol{\theta}, \boldsymbol{p}) = p(\boldsymbol{\theta})p(\boldsymbol{p} \,|\, \boldsymbol{\theta})$ constructed as $p(\boldsymbol{\theta}, \boldsymbol{p}) = e^{-H(\boldsymbol{\theta}, \boldsymbol{p})}$,

$$H(\boldsymbol{\theta}, \boldsymbol{p}) = \underbrace{-\ell(\boldsymbol{\theta}) + \frac{1}{2}\log\det G(\boldsymbol{\theta})}_{\phi(\boldsymbol{\theta})} + \tfrac{1}{2}\,\boldsymbol{p}^{\top}\, G^{-1}(\boldsymbol{\theta})\,\boldsymbol{a}\,. \tag{1}$$

The dynamics are given by the system of equations $\dot{\boldsymbol{p}} = \nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}, \boldsymbol{p})$, $\dot{\boldsymbol{\theta}} = \nabla_{\boldsymbol{p}} H(\boldsymbol{\theta}, \boldsymbol{p})$. Euclidean Hamiltonian Monte Carlo (HMC) is the special case where the metric is constant $\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{M}$. It results in explicit dynamics, in the sense that $\dot{\boldsymbol{p}}$ only depends on $\boldsymbol{\theta}$ and vice versa,

$$\dot{\boldsymbol{\theta}} = \boldsymbol{M}^{-1}\boldsymbol{p}, \quad \dot{\boldsymbol{p}} = \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}). \tag{2}$$

When the covariance is $\boldsymbol{G}(\boldsymbol{\theta})$, the dynamics are no longer explicit and are given by

$$\dot{\boldsymbol{\theta}} = \boldsymbol{G}^{-1}(\boldsymbol{\theta})\,\boldsymbol{p}, \quad \dot{\boldsymbol{p}} = -\nabla_{\boldsymbol{\theta}}\phi(\boldsymbol{\theta}) - \frac{1}{2}\nabla_{\boldsymbol{\theta}}\,\boldsymbol{p}^{\top}\,\boldsymbol{G}(\boldsymbol{\theta})^{-1}\,\boldsymbol{p}\,. \tag{3}$$

Numerical Integration HMC dynamics are numerically integrated by solving Eq. (2) with an explicit numerical integrator which is time-reversible and preserves volume, often the Leapfrog integrator (Neal et al., 2011). Numerical integration of Eq. (3) can be done with the generalized Leapfrog integrator (Girolami and Calderhead, 2011) or the midpoint integrator (Brofos and Lederman, 2021). The numerical integration has an implicit step which relies on fixed point iterations. Importantly, in both cases the integration depends on two key parameters that are difficult to set in practice:

- Step-size, denoted ε;
- Integration length $T \cdot \varepsilon$, where T is the number of integration steps.

Lagrangian Monte Carlo (LMC) The Riemannian sampler by Lan et al. (2015) admits an explicit but not volume preserving integrator. Lagrangian dynamics are the Riemannian Hamiltonian dynamics under the change of variable $\boldsymbol{v} = \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p}$. The random variable $\boldsymbol{v}$ is the velocity of the system and if the momentum is Gaussian then it is distributed $\boldsymbol{v} \,|\, \boldsymbol{\theta} \sim \mathcal{N}(0, \boldsymbol{G}(\boldsymbol{\theta})^{-1})$. The energy functional is defined in the same way as the Hamiltonian $E(\boldsymbol{\theta}, \boldsymbol{v}) = -\log p(\boldsymbol{\theta}, \boldsymbol{v})$. The identity $\det \boldsymbol{G}(\boldsymbol{\theta})^{-1} = 1/\det \boldsymbol{G}(\boldsymbol{\theta})$, gives

$$E(\boldsymbol{\theta}, \boldsymbol{v}) = -\ell(\boldsymbol{\theta}) - \frac{1}{2}\log\det \boldsymbol{G}(\boldsymbol{\theta}) + \frac{1}{2}\,\boldsymbol{v}^{T}\,\boldsymbol{G}(\boldsymbol{\theta})\,\boldsymbol{v}\,. \tag{4}$$

The dynamics are the geodesic equations with an extra term

$$\dot{\boldsymbol{\theta}} = \boldsymbol{v}, \quad \dot{\boldsymbol{v}} = -\eta(\boldsymbol{\theta}, \boldsymbol{v}) - \boldsymbol{G}(\boldsymbol{\theta})^{-1}\nabla\phi(\boldsymbol{\theta}), \tag{5}$$

where the k-th component of the vector $\eta(\boldsymbol{\theta}, \boldsymbol{v})$ is, in Einstein notation, $[\eta(\boldsymbol{\theta}, \boldsymbol{v})]^k = \Gamma^k_{ij}(\boldsymbol{\theta})\, \boldsymbol{v}^i\, \boldsymbol{v}^j$, where Γ^k_{ij} are the Christoffel symbols. Lan et al. (2015) constructs a numerical integrator for this. The integrator is explicit but not volume preserving. We account for this change of volume in the acceptance probability of a new sample $(\boldsymbol{\theta}_t, \boldsymbol{v}_t)$ after t steps of step-size ε starting from $(\boldsymbol{\theta}_0, \boldsymbol{v}_0)$

$$a(\boldsymbol{\theta}_t, \boldsymbol{v}_t \,|\, \boldsymbol{\theta}_0, \boldsymbol{v}_0) = \min\left\{1,\ \frac{e^{-E(\boldsymbol{\theta}_t, \boldsymbol{v}_t)}}{e^{-E(\boldsymbol{\theta}_0, \boldsymbol{v}_0)}} |\det J_t| \right\}.$$

Given a position $\boldsymbol{\theta}_0$, each sample is drawn by first sampling $\boldsymbol{v}_0 \sim \mathcal{N}(0, \boldsymbol{G}^{-1}(\boldsymbol{\theta}))$, updating the triplet $\boldsymbol{\theta}_t$, $\boldsymbol{v}_t$ and $|\det J_t|$ using Lan's integrator for $t = 1, .., T$ steps and step-size ε, and accepting the new sample with probability $a(\boldsymbol{\theta}_t, \boldsymbol{v}_t \,|\, \boldsymbol{\theta}_0, \boldsymbol{v}_0)$. The numerical integrator can found in Appendix A.4.

2.2. The No-U-Turn Sampler

The No-U-Turn Sampler by Hoffman and Gelman (2014) provides one practical solution for automatic selection of the key tuning parameters of Euclidean HMC, including ε, t and $\boldsymbol{M}$. The method is divided into a warm up phase and the actual sampling phase.

Warm up phase. During warm up, the Dual-Averaging algorithm, which is a stochastic gradient optimization algorithm, is used to tune the step-size ε of the numerical integrator to achieve a desired acceptance probability. Additionally, $\boldsymbol{M}$ is estimated during the warm up. The warm up samples are used to estimate the global covariance, which is then inverted and used as a fixed metric and applied as a linear reparametrization to the sample space to facilitate sampling. This offers a somewhat heuristic but in practice a good method for selecting the step size and the metric.

Sampling phase. The other key choice, the integration length, is handled by an adaptive method. Rather than fixing it at a given value, the algorithm attempts to identify the correct integration length by monitoring the expanding trajectory. Given an initial position in the parameter space, the trajectory is generated forward or backward in time. Each trajectory expansion doubles the number of previous integration steps. This is carried out until either the u-turn criterion is met or the trajectory is divergent. For details, see Hoffman and Gelman (2014).

After the u-turn (or divergence) is detected, a sample is chosen out of the whole trajectory by multinomial sampling, where the probability is given by the relative acceptance probability of each element of the trajectory (Carpenter et al., 2017; Cabezas et al., 2024). NUTS is guaranteed to be a valid sampler since it satisfies detailed balance (Hoffman and Gelman, 2014). Although NUTS was conceived for Hamiltonian Monte Carlo, it has been extended to Riemannian Manifold Hamiltonian Monte Carlo (RMHMC-NUTS) (Betancourt, 2017) and, as will be shown here, can be extended to Lagrangian Monte Carlo (LMC-NUTS). We are not aware of previous works that make this explicit.

Various alternative formulations that allow for adaptive control of the parameters have been proposed, for instance (Hoffman et al., 2021; Sountsov and Hoffman, 2021; Wang and Wibisono, 2022), that may offer practical advantages like parallel computations; we leave possible use of these techniques in context of Riemannian methods as future work.

3. Method

Here we outline our general approach for constructing a practical family of Riemannian MCMC methods. We assume sampling is done by following either the Hamiltonian or Lagrangian dynamics as explained in Section 2.1 in some metric $\boldsymbol{G}(\boldsymbol{\theta})$, and the sampler follows the general principle of NUTS. That is, (a) we use the Dual-Averaging algorithm during warm-up for selecting the optimal step-size and for fine-tuning the metric when applicable, and (b) we use the u-turn criterion for stopping integration for each proposal.

This section discusses the details briefly from the perspective of the choice of the metric and adaptation of its tuning parameters and the details of the stopping criterion in Riemannian metrics.

3.1. Metrics

Several Riemannian metrics have been explored as possible choices for posterior inference, but there is no general consensus (theoretical or empirical) on what the metric should be. All works proposing specific metrics naturally have valid argumentation in favor of the choice, but ultimately the relative merits of the alternatives depend on whether the chosen metric helps solving challenging inference problems more efficiently.

The **Fisher Information Metric (FIM)** is the inverse of the lower bound of the variance of unbiased estimator and is further motivated by the second order Taylor series expansion of the Kullback–Leibler divergence. Its geometric properties have been studied as part of Information Geometry (Amari and Nagaoka, 2000). It is defined as the covariance of the score function,

$$\boldsymbol{G}(\boldsymbol{\theta}) := \mathbb{E}_{\boldsymbol{y}\,|\,\boldsymbol{\theta}}\left[\nabla \log p(\boldsymbol{y}\,|\,\boldsymbol{\theta}) \nabla \log p(\boldsymbol{y}\,|\,\boldsymbol{\theta})^\top\right].$$

For regular probabilistic models, the score function has zero expectation, therefore the FIM computed as the expected outer product of the score functions or the expected Hessian matrix are the same, that is $\boldsymbol{G}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{y}\,|\,\boldsymbol{\theta}}\left[-\nabla^2 \log p(\boldsymbol{y}\,|\,\boldsymbol{\theta})\right]$. Sometimes the Hessian of the prior is added to the metric (Girolami and Calderhead, 2011), which is positive definite as long as the covariance of the score dominates. While the metric has strong theoretical justifications and no additional tuning parameters, the form of FIM depends on the likelihood. It can be troublesome or even impossible to derive analytically and typically needs costly numerical inversions (Lan et al., 2015).

One general metric is the **Softabs metric** by Betancourt (2013a). It applies a soft-absolute value to the eigenvalues of the Hessian of the log target, and thus is guaranteed to be positive definite. However, the metric requires second order differentiation to be computed, and RMHMC and LMC require one more order of derivatives of the metric tensor, which can be prohibitively expensive. The Hessian matrix, in its eigenvalue decomposition is $\nabla^2 \ell(\boldsymbol{\theta}) = \boldsymbol{Q}\,\boldsymbol{\Lambda}\,\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is orthonormal and $\boldsymbol{\Lambda}$ is diagonal. The softabs function is applied to each element of the diagonal matrix and the metric is

$$\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{Q} \cdot \text{softabs}(\boldsymbol{\Lambda}) \cdot \boldsymbol{Q}^\top .$$

Here $\text{softabs}(\lambda_i) = \lambda \coth(\alpha \lambda_i)$, and α is the cutoff set to $1e6$ in Betancourt (2013b).

Another choice for a general metric is the **Monge metric** by Hartmann et al. (2022). This metric may appear simple at first glance, but it is inherited from the geometry of the graph of the target distribution in the inclusion of a Euclidean space. It is formed by the outer product of gradients and gives a closed form inverse for Lan's integrator. It is given by

$$\boldsymbol{G}(\boldsymbol{\theta}) = I_D + \alpha^2 \nabla \ell(\boldsymbol{\theta}) \nabla \ell(\boldsymbol{\theta})^\top .$$

The Monge metric has nice computational properties due to fast inversions etc., but one needs to tune α^2 carefully for optimal performances.

Euclidean HMC rarely uses the identity metric $\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{I}_D$, but instead assumes a full metric $\boldsymbol{M}_D$ or a diagonal metric $\text{diag}(\boldsymbol{m})$. A non-identity metric facilitates the exploration of the parameter space, which is limited by the eigenvalues of the metric (Carpenter et al., 2017; Beskos et al., 2013). To achieve similar benefits in a Riemannian case, we propose a modification of the original Monge metric as

$$\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{M} + \alpha^2 \nabla \ell(\boldsymbol{\theta}) \nabla \ell(\boldsymbol{\theta})^\top , \tag{6}$$

where $\boldsymbol{M}$ is the global precision estimated during warm-up and α is again a tunable warp parameter. The first term can be interpreted as a reparametrization, effectively performing sampling in the space such that the global covariance is identity. The gradient transforms as co-vector, therefore the outer product of gradient follows the transformation rule of metrics. $\boldsymbol{M}$ also follows the transformation rule of metrics for linear transformations. The differential geometry derivation can be found in Appendix A.1.

3.2. Stopping criteria

Let $\boldsymbol{v}_t = \boldsymbol{G}^{-1}(\boldsymbol{\theta}_t)\, \boldsymbol{p}_t$ and recall that $\dot{\boldsymbol{\theta}}_t = \boldsymbol{v}_t$. The Euclidean u-turn criterion aims at stopping the Hamiltonian dynamics at a time t when the velocity vector $\boldsymbol{v}_t$ at $\boldsymbol{\theta}_t$ and the vector $\boldsymbol{\theta}_t - \boldsymbol{\theta}_0$ form an angle larger than $\pi/2$. Meaning that the position $\boldsymbol{\theta}_t$ of a moving particle starts to decrease its distance from its starting point $\boldsymbol{\theta}_0$, hence the name u-turn. Formally, the criterion minimizes the Euclidean distance between the original and updated position. The distance starts to decrease for the first time t such that,

$$\frac{\partial}{\partial t} \frac{1}{2} \|\boldsymbol{\theta}_t - \boldsymbol{\theta}_0\|^2 = \langle \boldsymbol{v}_t, \boldsymbol{\theta}_t - \boldsymbol{\theta}_0 \rangle = \left\langle \boldsymbol{v}_t, \int_0^t \boldsymbol{v}_s \right\rangle \leq 0, \tag{7}$$

where the integral is approximated by the sum of the computed velocities, i.e. $\int_0^t \boldsymbol{v}_s \approx \sum_s \boldsymbol{v}_s$. This can be generalized for Riemannian manifolds, but there is no unique clear generalization. Mathematically speaking, it is not obvious how to compare different tangent vectors at different tangent spaces (see Boothby, 2002, Chapter 4). Betancourt (2013b) generalized by considering the momentum variables in a similar fashion as the original NUTS,

$$\left\langle \boldsymbol{p}_t, \sum_s \boldsymbol{p}_s \right\rangle_{\boldsymbol{G}(\boldsymbol{\theta}_t)^{-1}} = \left\langle \boldsymbol{v}_t, \sum_s \boldsymbol{p}_s \right\rangle = \sum_s \langle \boldsymbol{v}_t, \boldsymbol{v}_s \rangle_{\boldsymbol{G}(\boldsymbol{\theta}_s)} .$$

We propose an alternative generalization to a Riemannian manifold by changing the Euclidean inner product in Equation 7 with the inner product on the tangent space at $\boldsymbol{\theta}_t$,

$$\left\langle \boldsymbol{v}_t, \sum_s \boldsymbol{v}_s \right\rangle_{\boldsymbol{G}(\boldsymbol{\theta}_t)} = \left\langle \boldsymbol{p}_t, \sum_s \boldsymbol{v}_s \right\rangle.$$

We refer to the stopping criteria as *Euclidean* (original No-U-Turn), *Betancourt* and *Riemannian*, respectively. In Appendix A.2 we take a closer look at the criterion proposed by Betancourt (2013b) and its close relation to ours. These methods have not been evaluated in practical settings, which we do in the following sections.

4. Experiments

Rather than focusing directly on specific algorithms or metrics, we conducted the empirical experimentation in an exhaustive manner, similar in nature to the early influential empirical comparisons of supervised machine learning algorithms (Bauer and Kohavi, 1999; Caruana and Niculescu-Mizil, 2006; Caruana et al., 2008). We consider a range of problems, covering synthetic distributions of known complex geometry, a prototypical machine learning model of Bayesian logistic regression, and a series of inference problems from the `posteriordb` (Magnusson et al., 2023) database that provides gold standard posterior estimates for a range of probabilistic programs. For each problem, we cover the Cartesian product over the mutually compatible choices of the integrator, metric, tuning parameters, and stopping criterion. This results in a total around 300 combinations. For additional analysis, the full results (as well as the algorithm implementations) are available at https://github.com/williwilliams3/expgeomjax.

For each scenario, we carried out five inference runs for different random seeds, and for each we computed a range of performance metrics. The set of alternatives for each choice, as well as the metrics, are briefly described below, and in Section 5 we summarize the main findings via a few isolated examinations of the overall result collection. Additional details on the setup are provided in Appendix B.

4.1. Inference tasks

Synthetic target distributions For $\boldsymbol{\theta} \in \mathbb{R}^D$, Neal's Funnel distribution (Neal, 2003) is

$$\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_D \,|0, \sigma^2)\mathcal{N}(\boldsymbol{\theta}_{1:D-1} \,|\mu, \exp\{\boldsymbol{\theta}_D\}\, \boldsymbol{I}_{D-1}),$$

for $\sigma > 0$ and $\mu \in \mathbb{R}^D$. The two dimensional Rosenbrock distribution is given by

$$\pi(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta}_1 \,|a, \frac{1}{2})\mathcal{N}(\boldsymbol{\theta}_2 \,|\, \boldsymbol{\theta}_1^2, \frac{1}{2b}),$$

for $a \in \mathbb{R}, b > 0$. Both have been used as benchmarks in previous works, and even though they do not depend on observed data both can be constructed as a Gaussian reparametrization which admits a Fisher Information metric, derived from the transformation rule of Riemannian metrics (Yu et al., 2024); see Appendix A.3. The diffeomorphism ϕ allows us to obtain reference samples by sampling $\boldsymbol{\psi}^{(1)}, .., \boldsymbol{\psi}^{(N)} \sim \mathcal{N}(0, \boldsymbol{I})$ and then mapping them using ϕ to obtain $\phi(\boldsymbol{\psi}^{(n)}) = \boldsymbol{\theta}^{(n)}$.

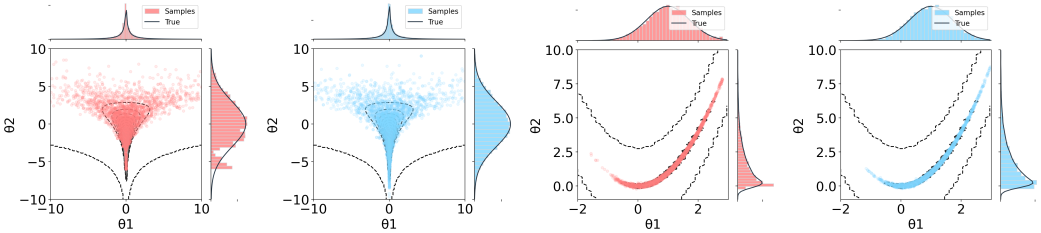

Figure 1: Euclidean (NUTS) and Riemannian (LMC-NUTS) comparison of samples in two dimensional toy distributions, LMC-NUTS go further down the neck of Neal's Funnel distribution and higher in Rosenbrock's distribution. This results in smaller distance to the true samples and higher effective sample size. The stopping criteria are Euclidean stopping and Riemannian stopping, respectively.

Bayesian Logistic Regression The model is given by

$$p(\boldsymbol{y}_i \,|\, \boldsymbol{\theta}, \boldsymbol{x}_i) = \text{Bernoulli}(\boldsymbol{y}_i \,|s(\boldsymbol{x}_i^\top \boldsymbol{\theta})), \quad p(\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\theta} \,|0, \alpha\, \boldsymbol{I}_D),$$

for $i = 1, .., N$ where $\alpha = 100$ and $s(\cdot)$ is the Sigmoid function. We consider three datasets (Henery and Taylor, 1992): Australian ($D = 15$), German ($D = 25$) and Heart ($D = 14$). Reference samples are computed following the procedure of `posteriordb`. That is, we run `Stan` for 10 chains to produce $10,000$ samples with a thinning of 10. We assure the effective sample size on each dimension is close to $10,000$ and $\hat{R} < 1.01$.

Posteriordb models We consider 3 different models from the Posteriordb database Magnusson et al. (2023). These are *eight schools non-centered* (8sn), *garch-11* (gar) and *low dimensional gaussian mixture* (ldg). Reference samples obtained by carefully tuned NUTS have been provided. For these models the derivation of the Fisher Information is possible but involves computing difficult expectations, and hence we do not use FIM in the experiments since it is not readily available in the implementation.

4.2. Inference method details

The experiments are run for all valid combinations of the following dimensions:

- **Sampling algorithms** (Section 2.1): Euclidean metric: Hamiltonian Monte Carlo, Riemannian metrics: Riemannian Manifold HMC and Lagrangian Monte Carlo;
- **Metrics** (Section 3.1): Euclidean, Fisher, Softabs, and two variants of the Monge metric (Monge-I and Monge-M);
- **Stopping rules** (Section 3.2): *Euclidean*, *Betancourt*, and *Riemannian*;
- **Tuning parameters**: For Monge metrics, we additionally cover $\alpha^2 \in \{0.001, 0.01, 0.1.1.0\}$. For SoftAbs we use constant $\alpha = 1e^6$, based on Betancourt (2013b).

4.3. Evaluation Metrics

For each combination of a method and an inference problem, we compute the following metrics explained here only briefly, while reporting the wall clock time in seconds. In the result tables we use the notation [mean, std], where the deviation is over the five runs.

- Effective sample size (ESS): Measures the number of independent samples produces by a set of dependent samples of the MCMC simulation; higher is better (Neal, 1993). ESS is computed for each dimension at a time, and hence we report both the minimum value and the average value. The minimum ESS provides information on the most challenging marginals and is important in ensuring robustness.

- 1-Wasserstein distance to reference samples: Measures the cost in Euclidean distance of transporting the samples to the reference samples; lower is better. Similar evaluation criterion is used in e.g. Zhang et al. (2022). For more details, see Flamary et al. (2021).

5. Results

We first summarize the overall results by selecting for each combination of a sampler and metric the method with the best accuracy (lowest Wasserstein distance to reference) over the possible choices of the stopping criterion. We keep Euclidean NUTS for comparison. Tables 1 to 3 report the results for the three sets of tasks. Note that if we would further select the best one among the three alternative stopping criteria, we obtain the best overall method for each task. This investigation provides information on two things: (a) The relative quality of the different metrics and samplers and (b) The effect of the stopping criterion. Below, we summarize the main observations from both perspectives.

The choice of the metric The first key observation is that in the whole set of results, the Euclidean metric may reach the same accuracy as the Riemannian metrics but not once surpasses the best Riemannian metric. This confirms that the Riemannian metrics are generally useful, outperforming the commonly used Euclidean choice as long as the metric is suitable. For the synthetic (Table 1) and logistic regression (Table 2) problems where the Fisher metric is available we always obtain the best overall result with a Fisher metric. However, the Softabs metric also performs well, and for the `posteriordb` examples (Table 3) it is always the best.

Stopping criteria Except for the synthetic data, the exact stopping criterion is largely inconsequential; we reach effectively identical Wasserstein distance with all three choices, despite using Riemannian metric in the sampling. For the synthetic cases (Table 1) there are small differences between the choices, but the ranking is not consistent.

Problem dimensionality A good metric is more important for complex distributions, and one way of increasing the difficulty is increasing the dimensionality. Figure 2 compares the Euclidean and Riemannian methods for the dimensionality in $\{2, 4, 8, \ldots, 64\}$, again splitting the results over the three stopping criteria. The Riemannian methods fare well in all dimensionalities, and we again observe no difference between the stopping criteria. Euclidean NUTS here fails for the higher-dimensional problems, even though it still appears to be working well from the perspective of mean ESS since it manages to explore the bulk

model	sampler	metric	stop	Wass	min ESS	avg ESS	t(s)
funnel	nuts	euc	euc	[1.21, 0.52]	[149, 75]	[289, 36]	1.3
D=2	nutslmc	softabs	euc	[0.82, 0.24]	[741, 111]	[777, 121]	9.2
	nutslmc	fisher	bet	[0.81, 0.24]	[1929, 207]	[2857, 558]	2.6
	nutslmc	fisher	riem	[0.88, 0.17]	[1877, 210]	[2406, 231]	2.1
rosenbrock	nuts	euc	euc	[0.17, 0.15]	[427, 406]	[521, 400]	2.3
D=2	nutslmc	softabs	euc	[0.1, 0.04]	[1139, 138]	[1282, 174]	10.6
	nutslmc	fisher	bet	[0.11, 0.04]	[669, 57]	[718, 66]	3.1
	nutslmc	fisher	riem	[0.09, 0.04]	[620, 88]	[645, 90]	3.1

Table 1: Synthetic distributions. For each stopping criterion we select the best combination of sampler and metric in terms of the 1-Wasserstein distance to reference samples.

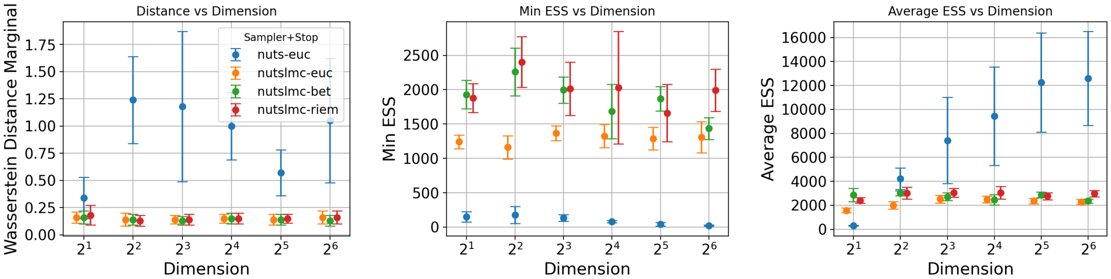

Figure 2: Left: Funnel's Wasserstein distance to marginal $\boldsymbol{\theta}_D$ for NUTS, LMC-NUTS and different stopping criteria for varying D. Middle and right: Minimum and average ESS.

of the distribution well. The minimum ESS, corresponding to the funnel dimensions, is extremely low, as the method does not sample from the right target.

Monge metric Fisher and Softabs metrics were generally the best in previous results. The Monge metric has a computational advantage over these and hence we inspect separately its behavior in the `posteriordb` problems in Table 4, to better understand the reasons. We observe that the newly proposed Monge-M is more robust for the choice of the tuning parameter, with $\alpha^2 = 1$ being best in all cases. For one of the tasks, ldg, it also achieves better overall accuracy. Comparing the final accuracies against the ones reported in Table 3 reveals that the metric tensors are actually comparable, with no discernible difference between the best Monge metric and the best overall solution. In other words, the Monge and Softabs metrics are in practice equally accurate assuming the tuning parameter α is chosen well. The Monge metric is clearly faster per iteration, whereas Softabs has higher ESS that approximately compensates it.

6. Conclusions

Although the concept of geometric MCMC has already been around for roughly two decades and there are both theoretical and empirical evidence that they can help in difficult inference

model	sampler	metr	stop	Wass	min ESS	avg ESS	t(s)
aus	nuts	euc	euc	[1.13, 0.02]	[1610, 116]	[2574, 129]	2.9
D=15	nutslmc	α^2=1.0(M)	euc	[1.13, 0.02]	[1221, 173]	[1932, 149]	2.9
	nutsrmhmc	softabs	bet	[1.13, 0.02]	[635, 53]	[816, 30]	4.9
	nutslmc	softabs	riem	[1.13, 0.02]	[657, 39]	[839, 26]	8.5
ger	nuts	euc	euc	[0.43, 0.0]	[10897, 732]	[15233, 1076]	2.9
D=25	nutslmc	fisher	euc	[0.43, 0.0]	[18600, 653]	[21438, 527]	3.5
	nutslmc	softabs	bet	[0.43, 0.0]	[19773, 355]	[23133, 691]	8.8
	nutslmc	fisher	riem	[0.43, 0.0]	[20263, 765]	[23918, 706]	3.5
hrt	nuts	euc	euc	[0.59, 0.0]	[10174, 1269]	[12942, 1524]	1.7
D=4	nutsrmhmc	softabs	euc	[0.59, 0.0]	[17297, 839]	[19147, 399]	4.6
	nutsrmhmc	fisher	bet	[0.6, 0.0]	[17409, 732]	[19078, 448]	3.5
	nutsrmhmc	softabs	riem	[0.6, 0.0]	[17102, 860]	[19172, 398]	5.1

Table 2: Logistic Regression. For each stopping criterion we select the best combination of sampler and metric in terms of the 1-Wasserstein distance to reference samples.

model	sampler	metr	stop	Wass	min ESS	avg ESS	t(s)
8sn	nuts	euc	euc	[16.74, 0.21]	[4697, 460]	[9740, 205]	1.9
	nutsrmhmc	softabs	euc	[16.73, 0.21]	[2613, 309]	[10551, 1240]	5.8
	nutslmc	softabs	bet	[16.73, 0.21]	[2032, 197]	[14062, 459]	9.1
	nutslmc	α^2=1.0(I)	riem	[16.72, 0.2]	[133, 36]	[792, 88]	2.8
gar	nuts	euc	euc	[0.21, 0.01]	[4743, 436]	[5585, 59]	1.6
	nutslmc	softabs	euc	[0.21, 0.01]	[2632, 295]	[5142, 209]	38
	nutslmc	softabs	bet	[0.22, 0.01]	[3283, 369]	[5916, 310]	47.6
	nutslmc	softabs	riem	[0.21, 0.01]	[3421, 243]	[6239, 328]	33.6
ldg	nuts	euc	euc	[0.02, 0.0]	[10019, 574]	[11332, 498]	2.2
	nutsrmhmc	softabs	euc	[0.02, 0.0]	[16294, 898]	[20723, 849]	26.1
	nutsrmhmc	softabs	bet	[0.02, 0.0]	[24017, 1513]	[25548, 752]	28.3
	nutsrmhmc	softabs	riem	[0.02, 0.0]	[24344, 1005]	[25946, 853]	25.8

Table 3: Posteriordb models. For each stopping criterion we select the best combination of sampler and metric in terms of the 1-Wasserstein distance to reference samples.

tasks, the methods are rarely used in practice. We believe this is largely due to the lack of easy-to-use and reliable alternatives, and in part due to a lack of empirical evidence on the practical value. Our work addresses both of these issues, by discussing how we can automate some of the tuning parameter choices of geometric samplers, by relying on the established techniques in NUTS, and by showcasing some alternative choices for the most obvious question of which metric to use. We avoid deep theoretical aspects of Riemannian geometry to keep the paper accessible to practitioners. We then complement the methodological overview by exhaustive empirical evaluation of the different choices, in a series of different kinds of inference problems, making the first attempt at quantifying the performance of

model	Monge	param	stop	Wass	min ESS	avg ESS	t(s)
8sn	M	α^2=1.0	bet	[16.74, 0.2]	[376, 113]	[750, 56]	2.7
	M	α^2=1.0	euc	[16.73, 0.21]	[3216, 358]	[7597, 499]	2.9
	M	α^2=1.0	riem	[16.74, 0.21]	[430, 108]	[767, 65]	2.2
	I	α^2=1.0	bet	[16.74, 0.2]	[131, 22]	[848, 85]	2.9
	I	α^2=1.0	euc	[16.73, 0.21]	[1430, 58]	[9724, 153]	2.6
	I	α^2=1.0	riem	[16.72, 0.2]	[133, 36]	[792, 88]	2.8
gar	M	α^2=1.0	bet	[0.23, 0.03]	[472, 78]	[593, 61]	4.2
	M	α^2=1.0	euc	[0.22, 0.02]	[2065, 232]	[2404, 246]	3.6
	M	α^2=1.0	riem	[0.23, 0.02]	[495, 62]	[641, 29]	3.6
	I	α^2=0.01	bet	[3.77, 0.13]	[4, 0]	[5, 1]	4.2
	I	α^2=1.0	euc	[0.22, 0.01]	[1370, 119]	[3097, 160]	4.4
	I	α^2=0.01	riem	[3.73, 0.07]	[4, 0]	[5, 1]	4.2
ldg	M	α^2=1.0	bet	[0.02, 0.0]	[1128, 117]	[1335, 103]	4.6
	M	α^2=1.0	euc	[0.02, 0.0]	[6764, 110]	[7862, 484]	4.8
	M	α^2=1.0	riem	[0.02, 0.0]	[1296, 134]	[1508, 141]	4.5
	I	α^2=0.1	bet	[3.26, 0.01]	[4, 0]	[5, 0]	5
	I	α^2=0.001	euc	[3.27, 0.03]	[4, 0]	[5, 0]	6.6
	I	α^2=0.001	riem	[3.27, 0.01]	[4, 0]	[5, 0]	4.8

Table 4: LMC-NUTS with Monge-I and Monge-M metrics, reporting the best result over $\alpha \in \{0.001, 0.01, 0.1, 1, 0\}$ for each stopping criterion.

geometric methods in more general settings. We considered a small set of problems, but the evaluation could easily be extended to a wider range of target densities, new metrics, or other sampler variants.

The main conclusions are that Riemannian MCMC methods work in practice and in most cases it is possible to outperform Euclidean samplers. All three metrics were found useful, with small differences in accuracy. We also observed that although the exact definition of a Riemannian u-turn is non-trivial, the exact stopping criterion does not seem to matter much. However, our results are only exploratory in the sense that we did not consider the question of how the metric would be chosen in practice; we showed that a combination of choices exists that works better than the commonly used Euclidean NUTS. Nevertheless, we did not yet study how a practitioner could make the choices without relying on knowledge (the true posterior) that would not be available in real cases. Simulation-Based Calibration (Modrák et al., 2023) could provide an alternative when reference samples are unavailable, by assessing the ability of the sampler in a model learned from synthetic samples, but further experimentation would be required to validate the approach.

Acknowledgments

The work was supported by the Research Council of Finland Flagship programme: Finnish Center for Artificial Intelligence FCAI, and additionally by the grant 345811. The authors acknowledge support from CSC – IT Center for Science, Finland.

References

S. Amari and H. Nagaoka. *Methods of Information Geometry*, volume 191. American Mathematical Soc., 2000.

E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, Boosting, and variants. *Machine learning*, 36:105–139, 1999.

A. Beskos, N. Pillai, G. Roberts, J.-M. Sanz-Serna, and A. Stuart. Optimal tuning of the hybrid Monte Carlo algorithm. *Bernoulli*, 19(5A):1501 – 1534, 2013. Publisher: Bernoulli Society for Mathematical Statistics and Probability.

M. Betancourt. A General Metric for Riemannian Manifold Hamiltonian Monte Carlo. In F. Nielsen and F. Barbaresco, editors, *Geometric Science of Information*, pages 327–334, Berlin, Heidelberg, 2013a. Springer Berlin Heidelberg.

M. Betancourt. A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*, 2017.

M. J. Betancourt. Generalizing the No-U-Turn sampler to Riemannian manifolds. *arXiv preprint arXiv:1304.1920*, 2013b.

W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Pure and Applied Mathematics. Academic Press, 2 edition, 2002.

J. Brofos and R. R. Lederman. Evaluating the implicit midpoint integrator for Riemannian Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1072–1081. PMLR, 2021.

A. Cabezas, A. Corenflos, J. Lao, and R. Louf. Blackjax: Composable Bayesian inference in JAX, 2024.

B. Carpenter, A. Gelman, M. D. Hoffman, D. Lee, B. Goodrich, M. Betancourt, M. Brubaker, J. Guo, P. Li, and A. Riddell. Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 2017.

R. Caruana and A. Niculescu-Mizil. An empirical comparison of Supervised Learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 161–168, New York, NY, USA, 2006. Association for Computing Machinery.

R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of Supervised Learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, page 96–103, New York, NY, USA, 2008. Association for Computing Machinery.

A. D. Cobb, A. G. Baydin, A. Markham, and S. J. Roberts. Introducing an explicit symplectic integration scheme for Riemannian manifold Hamiltonian Monte Carlo. *arXiv preprint arXiv:1910.06243*, 2019.

R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer. POT: Python Optimal Transport. *Journal of Machine Learning Research*, 22(78):1–8, 2021.

M. Girolami and B. Calderhead. Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73 (2):123–214, 2011.

M. Hartmann, M. Girolami, and A. Klami. Lagrangian manifold Monte Carlo on Monge patches. In *International Conference on Artificial Intelligence and Statistics*, pages 4764–4781. PMLR, 2022.

R. Henery and C. Taylor. StatLog: An evaluation of Machine Learning and Statistical Algorithms. In *Computational Statistics: Volume 1: Proceedings of the 10th Symposium on Computational Statistics*, pages 157–162. Springer, 1992.

M. Hoffman, A. Radul, and P. Sountsov. An Adaptive-MCMC Scheme for Setting Trajectory Lengths in Hamiltonian Monte Carlo . In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 3907–3915. PMLR, 13–15 Apr 2021.

M. D. Hoffman and A. Gelman. The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014.

A. Kristiadi, F. Dangel, and P. Hennig. The Geometry of Neural Nets' Parameter Spaces Under Reparametrization. In *Advances in Neural Information Processing Systems*, volume 36, pages 17669–17688. Curran Associates, Inc., 2023.

S. Lan, V. Stathopoulos, B. Shahbaba, and M. Girolami. Markov Chain Monte Carlo from Lagrangian Dynamics. *Journal of Computational and Graphical Statistics*, 24(2):357–378, 2015.

J. M. Lee. *Introduction to Riemannian manifolds*, volume 2. Springer, 2018.

C. Li, C. Chen, D. Carlson, and L. Carin. Preconditioned Stochastic Gradient Langevin Dynamics for Deep Neural Networks. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 1788–1794. AAAI Press, 2016.

M. Magnusson, P. Bürkner, and A. Vehtari. posteriordb: a set of posteriors for Bayesian inference and probabilistic programming, Oct. 2023.

M. Modrák, A. H. Moon, S. Kim, P. Bürkner, N. Huurre, K. Faltejsková, A. Gelman, and A. Vehtari. Simulation-Based Calibration Checking for Bayesian Computation: The Choice of Test Quantities Shapes Sensitivity. *Bayesian Analysis*, pages 1 – 28, 2023. Publisher: International Society for Bayesian Analysis.

R. M. Neal. Probabilistic inference using Markov Chain Monte Carlo methods. *Technical Report CRGT*, 1993.

R. M. Neal. Slice Sampling. *The annals of statistics*, 31(3):705–767, 2003.

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

P. Sountsov and M. D. Hoffman. Focusing on difficult directions for learning HMC trajectory lengths. *arXiv preprint arXiv:2110.11576*, 2021.

J.-K. Wang and A. Wibisono. Accelerating Hamiltonian Monte Carlo via Chebyshev Integration Time. In *The Eleventh International Conference on Learning Representations*, 2022.

T. Xifara, C. Sherlock, S. Livingstone, S. Byrne, and M. Girolami. Langevin Diffusions and the Metropolis-Adjusted Langevin Algorithm. *Statistics & Probability Letters*, 91:14–19, 2014.

H. Yu, M. Hartmann, B. Williams, and A. Klami. Scalable Stochastic Gradient Riemannian Langevin Dynamics in Non-Diagonal Metrics. *Transactions on Machine Learning Research*, 2023.

H. Yu, M. Hartmann, B. Williams Moreno Sanchez, M. Girolami, and A. Klami. Riemannian Laplace Approximation with the Fisher Metric. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pages 820–828. PMLR, May 2024.

L. Zhang, B. Carpenter, A. Gelman, and A. Vehtari. Pathfinder: Parallel quasi-Newton variational inference. *Journal of Machine Learning Research*, 23(306):1–49, 2022.

Appendix A. Mathematical derivations

A.1. Derivation of Monge Metric

One can estimate the global precision by using e.g. empirical samples, and use it as a fixed preconditioner. Probabilistic programming languages include this, which has been shown to be beneficial for certain problems (Carpenter et al., 2017; Cabezas et al., 2024). We therefore improve the formulation of the metric to account for such information. The metric can be derived from a differential geometric transformation view point. Consider a parametrization $\boldsymbol{\psi}$, such that the global covariance becomes identity. It can be transformed back to the current $\boldsymbol{\theta}$ parametrization using a linear transformation $\boldsymbol{\theta} = \boldsymbol{L}\,\boldsymbol{\psi}$ where $\boldsymbol{L}$ is a fixed matrix, with Jacobian $\boldsymbol{L}$ satisfying $\boldsymbol{L}_{ij} = \frac{\partial \boldsymbol{\theta}_i}{\partial \boldsymbol{\psi}_j}$, which can be viewed as a transformation applied globally. Since the transformation results in the global covariance of $\boldsymbol{\theta}$, we have

$$\boldsymbol{M}^{-1} = \mathrm{Var}(\boldsymbol{\theta}) = \boldsymbol{L}\,\mathrm{Var}(\boldsymbol{\psi})\,\boldsymbol{L}^\top = \boldsymbol{L}\,\boldsymbol{L}^\top\,.$$

Using notations based on Kristiadi et al. (2023), we write the following. The transformation law for gradients 14 gives $\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta}) = \boldsymbol{L}^{-\top}\,\nabla_{\boldsymbol{\psi}}\ell(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\boldsymbol{L}\,\boldsymbol{\psi}}$. The inverse mass metric is $\boldsymbol{M} = \boldsymbol{L}^{-\top}\,\boldsymbol{L}^{-1}$. Therefore, the proposed metric is

$$\boldsymbol{G}(\boldsymbol{\theta}) = \boldsymbol{M} + \alpha^2 \nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta})\nabla_{\boldsymbol{\theta}}\ell(\boldsymbol{\theta})^\top = \boldsymbol{L}^{-\top}\left(\boldsymbol{I} + \alpha^2\,\nabla_{\boldsymbol{\psi}}\ell(\boldsymbol{\theta})\nabla_{\boldsymbol{\psi}}\ell(\boldsymbol{\theta})^\top\Big|_{\boldsymbol{\theta}=\boldsymbol{L}\,\boldsymbol{\psi}}\right)\boldsymbol{L}^{-1}\,.$$

In practice we may replace $\boldsymbol{M}$ with $\boldsymbol{m} = \mathrm{diag}(\boldsymbol{M})$, i.e. a diagonal metric. This facilities the implementation of the algorithm to not store any matrices in memory, only vectors. Note that the second term in proposed metric is not the results of the transformation law of metrics from the original Monge metric, nevertheless gave good results in practice.

A.2. Derivation of alternative Riemannian stopping criterion for Lagrangian Flow

In the following, we provide a high-level sketch for the derivation of the alternative Riemannian stopping criterion. Let us adopt the notation of Betancourt (2013b) and follow along similar steps. The difference being that we consider Lagrangian dynamics instead of Hamiltonian dynamics. Denote positions by $\boldsymbol{q}$, and velocities by $\boldsymbol{v}$. We denote an element in the tangent bundle by $S \in T\mathcal{M}$ and its projection to the manifold by $R = \pi(S)$. The inner product

$$\langle \boldsymbol{v}(R_t), \boldsymbol{\rho}(R_t)\rangle_{\boldsymbol{G}(\boldsymbol{q}(R_t))}\,,$$

is defined on the tangent space, where $\boldsymbol{\rho}(R_t) = \int_0^t \boldsymbol{v}(R_s)\mathrm{d}s$. The canonical one form $\theta = \sum_j \boldsymbol{v}^j \frac{\partial}{\partial\,\boldsymbol{q}^j}$ is a horizontal form in $T\mathcal{M}$. It can be dragged along the Lagrangian flow denoted H,

$$\theta_t^* = \theta + \int_0^t \mathcal{L}_H \theta d\tau.$$

Where $\mathcal{L}_H\,\theta$ is the Lie derivative. Following the steps by Betancourt (2013b), we obtain the value of the components of the Lie Derivative,

$$\begin{aligned}
(\mathcal{L}_H\,\theta)_j &= \sum_k \left[\frac{\mathrm{d}\,\boldsymbol{q}^k}{\mathrm{d}t}\frac{\partial}{\partial\,\boldsymbol{q}^k} + \frac{\mathrm{d}\,\boldsymbol{v}^k}{\mathrm{d}t}\frac{\partial}{\partial\,\boldsymbol{v}_k}\right]\boldsymbol{\theta}_j + \boldsymbol{\theta}_k\,\frac{\partial}{\partial\,\boldsymbol{q}^j}\frac{\mathrm{d}\,\boldsymbol{q}^k}{\mathrm{d}t} \\
&= \sum_k \left[\frac{\mathrm{d}\,\boldsymbol{q}^k}{\mathrm{d}t}\frac{\partial}{\partial\,\boldsymbol{q}^k} + \frac{\mathrm{d}\,\boldsymbol{v}^k}{\mathrm{d}t}\frac{\partial}{\partial\,\boldsymbol{v}^k}\right]\boldsymbol{v}_j + \boldsymbol{v}_k\,\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial\,\boldsymbol{q}^k}{\partial\,\boldsymbol{q}^j} \\
&= \sum_k \frac{\mathrm{d}\,\boldsymbol{q}^k}{\mathrm{d}t}\frac{\partial\,\boldsymbol{v}_j}{\partial\,\boldsymbol{q}^k} + \frac{\mathrm{d}\,\boldsymbol{v}^k}{\mathrm{d}t}\delta_k^j + \boldsymbol{v}_k\,\frac{\mathrm{d}}{\mathrm{d}t}\delta_j^k \\
&= \sum_k \frac{\mathrm{d}\,\boldsymbol{q}^k}{\mathrm{d}t}0 + \frac{\mathrm{d}\,\boldsymbol{v}^k}{\mathrm{d}t}\delta_k^j + \boldsymbol{v}_k\,0 \\
&= \frac{\mathrm{d}\,\boldsymbol{v}^j}{\mathrm{d}t}.
\end{aligned}$$

Lie dragging is defined against the flow, if we drag from beginning to end of the trajectory we drag along the flow. Giving us,

$$\begin{aligned}\theta^*_{-t}(S_t)_j &= \left(\boldsymbol{v}^j(S_t) + \int_t^0 \frac{\mathrm{d}\,\boldsymbol{v}^j}{\mathrm{d}t}\right)\frac{\partial}{\partial q^j} \\ &= \left(\boldsymbol{v}^j(S_t) + \boldsymbol{v}^j(S_0) - \boldsymbol{v}^j(S_t)\right)\frac{\partial}{\partial q^j} \\ &= \boldsymbol{v}^j(S_0)\frac{\partial}{\partial q^j}.\end{aligned}$$

It defines a unique vector $\boldsymbol{v}^*(R_t)$ with components $\boldsymbol{v}^{*j}(R_t) = \boldsymbol{v}^j(R_0)$. Then define $\boldsymbol{\rho}(R_t) := \int_0^t \boldsymbol{v}^{*j}(R_s)\mathrm{d}s$, which is approximated by sum of the velocities given through the numerical integration

$$\langle \boldsymbol{v}(R_t), \boldsymbol{\rho}(R_t)\rangle_{\boldsymbol{G}(\boldsymbol{q}(R_t))} \approx \left\langle \boldsymbol{v}(R_t), \sum_s \boldsymbol{v}^*(R_s)\right\rangle_{\boldsymbol{G}(\boldsymbol{q}(R_t))}.$$

This is the *Riemmanian* stopping criterion in the main paper.

A.3. Derivation of the Fisher Information metric

Synthetic distributions Consider Neal's Funnel and Rosenbrock distribution introduced in Section 4. While not being probabilistic models dependent on observed data $\boldsymbol{y}$, they can both be constructed from a transformation in whose original space, the related random variable follows a Gaussian distribution. As observed by Yu et al. (2024), we can still define a metric-tensor inspired by the FIM through the transformation rule of Riemannian metrics. Let $\boldsymbol{\theta} = \phi(\boldsymbol{\psi})$ be a diffeomorphism. If for $\boldsymbol{\psi} \sim \mathcal{N}(\boldsymbol{\psi} \,|\, \boldsymbol{\mu}, \Sigma)$, then the transformation rule 13 yields

$$G(\boldsymbol{\theta}) = \frac{\partial \phi^{-1}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}^\top \Sigma^{-1} \frac{\partial \phi^{-1}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{8}$$

Let $\boldsymbol{\psi} \sim \mathcal{N}(0, \boldsymbol{I}_D)$, the transformation for Neal's Funnel and the inverse Jacobian are

$$\phi(\boldsymbol{\psi}) = \begin{bmatrix} \exp^{\sigma\,\boldsymbol{\psi}_D/2} \boldsymbol{\psi}_{1:D-1} \\ \sigma\,\boldsymbol{\psi}_D \end{bmatrix}, \quad \frac{\partial \phi^{-1}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \exp^{-\boldsymbol{\theta}_D/2} I & -\frac{1}{2}\exp^{-\boldsymbol{\theta}_D/2} \boldsymbol{\theta}_{1:D-1} \\ 0 & \frac{1}{\sigma} \end{bmatrix}.$$

The two dimensional Rosenbrock distribution has the diffeomorphism and inverse Jacobian,

$$\phi(\boldsymbol{\psi}) = \begin{bmatrix} a + \frac{1}{\sqrt{2}}\boldsymbol{\psi}_1 \\ \boldsymbol{\theta}_1^2 + \frac{1}{\sqrt{2b}}\boldsymbol{\psi}_2 \end{bmatrix}, \quad \frac{\partial \phi^{-1}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \begin{bmatrix} \sqrt{2} & 0 \\ -2\sqrt{2b}\,\boldsymbol{\theta}_1 & \sqrt{2b} \end{bmatrix}.$$

These diffeomorphisms allow for the computation of reference samples by sampling first $\boldsymbol{\psi}^{(1)}, .., \boldsymbol{\psi}^{(N)} \sim \mathcal{N}(0, \boldsymbol{I})$ and mapping $\phi(\boldsymbol{\psi}^{(n)}) = \boldsymbol{\theta}^{(n)}$.

Proposition 1 *The Riemannian metric in equation 8, coincides with the FIM of the probabilistic model*

$$\begin{aligned} p(\boldsymbol{\mu} \,|\, \boldsymbol{\theta}, \boldsymbol{\Sigma}) &= \mathcal{N}(\boldsymbol{\mu} \,|\phi^{-1}(\boldsymbol{\theta}), \boldsymbol{\Sigma}), \\ p_J(\boldsymbol{\theta}) &= \sqrt{\det G(\boldsymbol{\theta})}. \end{aligned}$$

Where p_J is Jeffrey's prior. Furthermore the target distribution coincides with the posterior distribution of the model over the change of variables.

Proof The FIM of the model is

$$\mathbb{E}_{\boldsymbol{\mu}}\left(-\nabla^2_{\boldsymbol{\theta}}\mathcal{N}(\boldsymbol{\mu}\,|\phi^{-1}(\boldsymbol{\theta}),\boldsymbol{\Sigma})\right)=\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}^{\top}\Sigma^{-1}\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}.$$

Which coincides with equation 8. Now let us proof that the posterior coincides with the target distribution. Notice that Jeffrey's prior encodes the change of variable of the transformation $\phi(\boldsymbol{\psi})$,

$$\sqrt{\det\boldsymbol{G}(\boldsymbol{\theta})}=\left(\det\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}^{\top}\right)^{\frac{1}{2}}\left(\det\boldsymbol{\Sigma}^{-1}\right)^{\frac{1}{2}}\left(\det\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)^{\frac{1}{2}}=\left(\det\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)\left(\det\boldsymbol{\Sigma}^{-1}\right)^{\frac{1}{2}}.$$

Then the posterior yields,

$$\begin{aligned}p(\boldsymbol{\theta}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma})&\propto\mathcal{N}(\boldsymbol{\mu}\,|\phi^{-1}(\boldsymbol{\theta}),\boldsymbol{\Sigma})\sqrt{\det G(\boldsymbol{\theta})}\\&\propto\mathcal{N}(\phi^{-1}(\boldsymbol{\theta})|\mu,\boldsymbol{\Sigma})\left|\det\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right|.\end{aligned}$$

Where the last line is the distribution of $\boldsymbol{\theta}$ under the transformation. We conclude the posterior $p(\boldsymbol{\theta}\,|\,\boldsymbol{\mu},\boldsymbol{\Sigma})$ of the model matches the target $p(\boldsymbol{\theta}=\phi(\boldsymbol{\psi}))$. ■

Logistic regression The derivation of the Fisher Information metric for Logistic Regression can be found in Girolami and Calderhead (2011). The right hand side adds the Hessian of the prior, which is positive definite. It is given by

$$\boldsymbol{G}=\boldsymbol{X}^{\top}\boldsymbol{\Lambda}\,\boldsymbol{X}+\alpha^{-1}\boldsymbol{I},$$

where the covariates are stacked in a $N\times D$ matrix $\boldsymbol{X}=[\boldsymbol{x}_1,..,\boldsymbol{x}_N]^{\top}$ and the diagonal $N\times N$ matrix $\boldsymbol{\Lambda}_{nn}=\sigma(\boldsymbol{x}_i^{\top}\boldsymbol{\theta})\big(1-\sigma(\boldsymbol{x}_i^{\top}\boldsymbol{\theta})\big)$.

A.4. Derivation of the Lagrangian Monte Carlo dynamics with modified Monge metric

In this subsection we derive the numerical integrator for Lagrangian Monte Carlo with the modified Monge metric. The numerical integrator of Lan et al. (2015) for Lagrangian dynamics has the half step update of the volume term and the velocity given by,

$$\begin{aligned}\log\det J&=\log\det J+\log\det\Big(G(\boldsymbol{\theta}^{(n)})-\frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}^{(n)},\boldsymbol{v}^{(n)})\Big),\\\boldsymbol{v}^{(n+1/2)}&=\Big[G(\boldsymbol{\theta}^{(n)})+\frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}^{(n)},\boldsymbol{v}^{(n)})\Big]^{-1}\Big[G(\boldsymbol{\theta}^{(n)})\,\boldsymbol{v}^{(n)}-\frac{\varepsilon}{2}\nabla\phi(\boldsymbol{\theta}^{(n)})\Big],\\\log\det J&=\log\det J-\log\det\Big(G(\boldsymbol{\theta}^{(n)})+\frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}^{(n)},\boldsymbol{v}^{(n+1/2)})\Big),\end{aligned}\tag{9}$$

where $\phi(\cdot)$ is defined in Equation 1. The matrices $\Omega(\boldsymbol{\theta},\boldsymbol{v})$ and $\tilde{\Omega}(\boldsymbol{\theta},\boldsymbol{v})$ are defined as

$$\Omega(\boldsymbol{\theta},\boldsymbol{v})_{kj}:=\sum_{i=1}^{D}\boldsymbol{v}^i\,\Gamma^k_{i,j}(\boldsymbol{\theta}),\tag{10}$$

$$\tilde{\Omega}(\boldsymbol{\theta},\boldsymbol{v}):=G(\boldsymbol{\theta})\Omega(\boldsymbol{\theta},\boldsymbol{v}).\tag{11}$$

Where Γ^k_{ij} are the Christoffel symbols under the Levi-Citiva connection (Lee, 2018). Let us consider the Modified Monge metric,

$$\boldsymbol{G}(\boldsymbol{\theta}) = \operatorname{diag} \boldsymbol{m} + \alpha^2 \nabla\ell(\boldsymbol{\theta})\nabla\ell(\boldsymbol{\theta})^\top .$$

From this point onward we drop dependency on $\boldsymbol{\theta}$ to reduce notation cluster. Define $L_\alpha := 1 + \alpha^2 \left\| 1/\sqrt{\boldsymbol{m}} \odot \nabla\ell \right\|^2$. The symbol $\odot$ defines the Hadamard (element-wise) product between the two vectors. The Sherman-Morrison formula gives,

$$\boldsymbol{G}^{-1}(\boldsymbol{\theta}) = \operatorname{diag}(1/\boldsymbol{m}) - \tfrac{\alpha^2}{L_\alpha}(\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell)(\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell)^\top .$$

$$\det \boldsymbol{G} = L_\alpha \prod_{i=1}^{D} \boldsymbol{m}^i \,.$$

The Energy functional It is given by $E(\boldsymbol{\theta}, \boldsymbol{v}) = -\log p(\boldsymbol{v} \mid \boldsymbol{\theta}) p(\boldsymbol{\theta})$, this is,

$$\begin{aligned} E(\boldsymbol{\theta}, \boldsymbol{v}) &= -\ell(\boldsymbol{\theta}) - \tfrac{1}{2}\log\det \boldsymbol{G}(\boldsymbol{\theta}) + \tfrac{1}{2}\|\boldsymbol{v}\|^2_G \\ &= -\ell(\boldsymbol{\theta}) - \tfrac{1}{2}(\log L_\alpha + \sum \log \boldsymbol{m}^i) + \tfrac{1}{2}(\left\|\sqrt{\boldsymbol{m}} \odot \boldsymbol{v}\right\|^2 + \alpha^2 \langle \boldsymbol{v}, \nabla\ell\rangle^2). \end{aligned}$$

The Christoffel symbols The modified Monge metric has associated the global chart $\eta(\boldsymbol{\theta}) = (\sqrt{\boldsymbol{m}} \odot \boldsymbol{\theta}, \alpha\ell(\boldsymbol{\theta}))$. The Christoffel symbols can derived from the simpler formula, where the inner product is w.r.t. the ambient space $\mathbb{R}^{D+1}$

$$\begin{aligned} \Gamma^k_{ij} &= g^{kl} \left\langle \frac{\partial^2 \eta}{\partial \boldsymbol{\theta}^i \, \partial \boldsymbol{\theta}^j}, \frac{\partial \eta}{\partial \boldsymbol{\theta}^l} \right\rangle \\ &= \sum_l \left(\tfrac{1}{\boldsymbol{m}^k}\delta^l_k - \tfrac{\alpha^2}{L_\alpha}(\tfrac{1}{\boldsymbol{m}^k}\partial_k\ell)(\tfrac{1}{\sqrt{\boldsymbol{m}^l}}\partial_l\ell) \right) \left\langle (0, .., 0, \alpha\partial_i\partial_j\ell), (0, .., \boldsymbol{m}^l, .., \alpha\partial_l) \right\rangle \\ &= \alpha^2 \partial^2_{ij}\ell \sum_l \left(\tfrac{1}{\boldsymbol{m}^k}\delta^l_k - \tfrac{\alpha^2}{L_\alpha}(\tfrac{1}{\boldsymbol{m}^k}\partial_k\ell)(\tfrac{1}{\boldsymbol{m}^l}\partial_l\ell) \right) \; \partial_l\ell \\ &= \alpha^2 \partial^2_{ij}\ell \left(1 - \tfrac{\alpha^2}{L_\alpha}\|\nabla\ell\|^2_{\operatorname{diag}(1/\boldsymbol{m})} \right) \tfrac{1}{\boldsymbol{m}^k}\partial_k\ell \\ &= \tfrac{\alpha^2}{L_\alpha}\partial^2_{ij}\ell \; \tfrac{1}{\boldsymbol{m}^k}\partial_k\ell. \end{aligned}$$

They can be expressed in matrix form of size $D \times D$, for $k = 1, .., D$

$$\Gamma^k = \tfrac{\alpha^2}{L_\alpha}\nabla^2\ell \tfrac{1}{\boldsymbol{m}^k}\partial_k\ell.$$

Matrix $\Omega(\boldsymbol{\theta}, \boldsymbol{v})$ Lan's integrator updates in Equation 9 depend on the matrix $\Omega(\boldsymbol{\theta}, \boldsymbol{v})$, which is a matrix whose (k, j) element is given by $\boldsymbol{v}^i \, \Gamma^k_{i,j}$

$$\Omega_{kj} = \boldsymbol{v}^i \, \Gamma^k_{i,j} = [\Gamma^k \, \boldsymbol{v}]^j = \tfrac{\alpha^2}{L_\alpha}[\nabla^2\ell \, \boldsymbol{v}]^j [\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell]^k .$$

It is the outer product of two vectors,

$$\Omega = \tfrac{\alpha^2}{L_\alpha}(\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell)(\boldsymbol{v}^\top \, \nabla^2\ell).$$

The matrix $\tilde{\Omega}(\boldsymbol{\theta}, \boldsymbol{v}) = \boldsymbol{G}(\boldsymbol{\theta})\Omega(\boldsymbol{\theta}, \boldsymbol{v})$ is an outer product as well,

$$\begin{aligned}\tilde{\Omega} = \boldsymbol{G}\,\Omega &= \left(\operatorname{diag}\boldsymbol{m} + \alpha^2\nabla\ell\nabla\ell^\top\right)\tfrac{\alpha^2}{L_\alpha}(\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell)(\boldsymbol{v}^\top\,\nabla^2\ell)^\top \\ &= \tfrac{\alpha^2}{L_\alpha}(1 + \alpha^2\|\nabla\ell\|^2_{\operatorname{diag}(\frac{1}{\boldsymbol{m}})})\nabla\ell(\boldsymbol{v}^\top\,\nabla^2\ell)^\top \\ &= \alpha^2\nabla\ell(\boldsymbol{v}^\top\,\nabla^2\ell).\end{aligned}$$

Gradient of potential energy The potential energy for LMC is $\phi(\boldsymbol{\theta}) = -\ell(\boldsymbol{\theta}) + \frac{1}{2}\log\det\boldsymbol{G}(\boldsymbol{\theta})$ and we require the first order derivatives. The first term $\nabla\ell$ depends on the log-target and the second can be computed as

$$\frac{\partial}{\partial\boldsymbol{\theta}}\log\det\boldsymbol{G} = \tfrac{1}{\det G}\frac{\partial}{\partial\boldsymbol{\theta}}\det\boldsymbol{G} = \tfrac{1}{L_\alpha}\frac{\partial}{\partial\boldsymbol{\theta}}L_\alpha = \tfrac{\alpha^2}{L_\alpha}\frac{\partial}{\partial\boldsymbol{\theta}}\nabla\ell^\top(\operatorname{diag}\tfrac{1}{\boldsymbol{m}})\nabla\ell = \tfrac{2\alpha^2}{L_\alpha}\nabla^2\ell(\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell).$$

The last step uses the identity $\frac{\partial}{\partial\boldsymbol{x}}\boldsymbol{x}^\top B\,\boldsymbol{x} = (B + B^\top)\,\boldsymbol{x}$. All together gives,

$$\nabla\phi = -\nabla\ell + \tfrac{\alpha^2}{L_\alpha}\nabla^2\ell(\tfrac{1}{\boldsymbol{m}} \odot \nabla\ell) \tag{12}$$

Inverse and Determinant updates The determinant $\det\left(\boldsymbol{G}(\boldsymbol{\theta}) \pm \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}, \boldsymbol{v})\right)$, and $\boldsymbol{G}(\boldsymbol{\theta}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}, \boldsymbol{v})$ which are necessary in the half-step integrator updates in equation 9, are expressed as a matrix plus an outer product,

$$\boldsymbol{G}(\boldsymbol{\theta}) + \tfrac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}, \boldsymbol{v}) = \operatorname{diag}\boldsymbol{m} + \alpha^2\nabla\ell\nabla\ell^\top + \tfrac{\alpha^2\varepsilon}{2}\nabla\ell(\boldsymbol{v}^\top\,\nabla^2\ell) = \operatorname{diag}\boldsymbol{m} + \boldsymbol{b}\,\boldsymbol{a}^\top.$$

Where $\boldsymbol{b} = \nabla\ell$, $\boldsymbol{a} = \alpha^2(\nabla\ell + \frac{\varepsilon}{2}\,\boldsymbol{v}^\top\,\nabla^2\ell)$ and $\boldsymbol{M} = \operatorname{diag}\boldsymbol{m}$. Sherman Morrison gives the inverse and determinant.

$$\begin{aligned}\det\left(\boldsymbol{G}(\boldsymbol{\theta}) + \tfrac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}, \boldsymbol{v})\right) &= \det(\boldsymbol{M})(1 + \boldsymbol{a}^\top\,\boldsymbol{M}^{-1}\,\boldsymbol{b}) = \det(\boldsymbol{M})(L_\alpha + \tfrac{\alpha^2\varepsilon}{2}\left\langle\boldsymbol{v}, \nabla\ell \odot \tfrac{1}{\boldsymbol{m}\nabla^2\ell}\right\rangle) \\ (\boldsymbol{G}(\boldsymbol{\theta}) + \tfrac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{\theta}, \boldsymbol{v}))^{-1} &= \boldsymbol{M}^{-1} - \boldsymbol{M}^{-1}\,\boldsymbol{b}\,\boldsymbol{a}^\top\,\boldsymbol{M}^{-1}/(1 + \boldsymbol{b}\,\boldsymbol{M}^{-1}\,\boldsymbol{a}^\top).\end{aligned}$$

Plug in all quantities in Equation 9 we have Lan's numerical integrator on the modified Monge metric. Note that all terms are given in closed form which reduces the cost from cubic to linear in operations. Although the computation of the Hessian of the log-target with automatic differentiation increases the cost with respect to HMC. The code implementation uses Hessian vector products for second order derivatives.

A.5. Transformation laws

The transformation law for Riemannian tensors. Let $\boldsymbol{\psi} \in \mathbb{R}^D$ and $\boldsymbol{\theta} = \phi(\boldsymbol{\psi})$ be a smooth bijective transformation such that $\phi : \mathbb{R}^D \to \mathbb{R}^D$. If $\boldsymbol{G}(\boldsymbol{\psi})$ is a Riemannian metric tensor in the $\boldsymbol{\psi}$-coordinates, then the metric tensor in the $\boldsymbol{\theta}$-coordinates, $\boldsymbol{G}(\boldsymbol{\theta})$, transforms according to the rule:

$$\boldsymbol{G}(\boldsymbol{\theta}) = \left(\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}\right)^\top\boldsymbol{G}(\phi^{-1}(\boldsymbol{\theta}))\frac{\partial\phi^{-1}(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \left(\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\psi}}\right)^{-T}\boldsymbol{G}(\boldsymbol{\psi})|_{\boldsymbol{\psi}=\phi^{-1}(\boldsymbol{\theta})}\left(\frac{\partial\boldsymbol{\theta}}{\partial\boldsymbol{\psi}}\right)^{-1}. \tag{13}$$

The transformation law for Euclidean gradients. Let $\ell : \mathbb{R}^D \to \mathbb{R}$ be a smooth vector function. The application of the chain rule,

$$\nabla_\theta \ell(\boldsymbol{\theta}) = \left(\frac{\partial \boldsymbol{\psi}}{\partial \boldsymbol{\theta}}\right)^\top \nabla_{\boldsymbol{\psi}} \ell(\phi(\boldsymbol{\psi})) = \left(\frac{\partial \boldsymbol{\theta}}{\partial \boldsymbol{\psi}}\right)^{-\top} \nabla_{\boldsymbol{\psi}}\ \ell(\boldsymbol{\theta})|_{\boldsymbol{\theta}=\phi(\boldsymbol{\psi})}\,. \tag{14}$$

Appendix B. Experimental setup

B.1. Adaptation and sampling

When $\boldsymbol{M}$ and the step-size are needed we use window adaptation as in Cabezas et al. (2024). When only the step-size is needed we use dual-averaging. The adaptation period is of 1,000 iterations. We draw 8 chains of 10,000 samples for each configuration.

B.2. Cluster used

We ran the experiments in a cluster of CPUs *Intel(R) Xeon(R) Gold 6230 CPU @ 2.10GHz*.

B.3. Total number of combinations

The sampler considered, with respective metric and stopping options are:

- NUTS: Metric: Euclidean
  Stopping: Euclidean
- RMHMC-NUTS
  Metrics: Fisher, Softabs
  Stopping: Euclidean, Betancourt, Riemannian
- LMC-NUTS
  Metrics: Fisher, Softabs, Monge-I, Monge-M (4 values α's each)
  Stopping: Euclidean, Betancourt, Riemannian

The models considered are Synthetic models (3 different models), Bayesian Logistic Regression (3 datasets) and `posteriordb` (3 different models). Notice that Fisher Information metric is not used in `posteriordb`. We have a count of 278 different combinations. For the increasing dimension experiment we sample NUTS, LMC-NUTS (Fisher metric and Euclidean, Betancourt, Riemannian) giving a total of 25 new combinations. The total count of different combinations considered is 303.