Lagrangian manifold Monte Carlo on Monge patches

Hartmann, Marcelo

Journal of Machine Learning Research
2022-01-29

Hartmann , M , Girolami , M & Klami , A 2022 , Lagrangian manifold Monte Carlo on Monge patches . in G Camps-Vall , F J R Ruiz & I Valera (eds) , Proceedings of The 25th International Conference on Artificial Intelligence and Statistics . Proceedings of Machine Learning Research, PMLR , vol. 151 , Journal of Machine Learning Research , pp. 4764-4781 , International Conference on Artificial Intelligence and Statistic , 28/03/2022 . < https://proceedings.mlr.press/v151/hartmann22a.html >

http://hdl.handle.net/10138/347534

Lagrangian Manifold Monte Carlo on Monge Patches

Marcelo Hartmann
University of Helsinki
Department of Computer Science

Mark Girolami
University of Cambridge
Department of Engineering
& The Alan Turing Institute

Arto Klami
University of Helsinki
Department of Computer Science

Abstract

The efficiency of Markov Chain Monte Carlo (MCMC) depends on how the underlying geometry of the problem is taken into account. For distributions with strongly varying curvature, Riemannian metrics help in efficient exploration of the target distribution. Unfortunately, they have significant computational overhead due to e.g. repeated inversion of the metric tensor, and current geometric MCMC methods using the Fisher information matrix to induce the manifold are in practice slow. We propose a new alternative Riemannian metric for MCMC, by embedding the target distribution into a higher-dimensional Euclidean space as a Monge patch and using the induced metric determined by direct geometric reasoning. Our metric only requires first-order gradient information and has fast inverse and determinants, and allows reducing the computational complexity of individual iterations from cubic to quadratic in the problem dimensionality. We demonstrate how Lagrangian Monte Carlo in this metric efficiently explores the target distributions.

1 INTRODUCTION

Markov Chain Monte Carlo (MCMC) algorithms provide samples from complex distributions for which direct sampling is difficult, and are routinely used in Bayesian statistics for sampling from the posterior distribution of a model (Chkrebtii et al., 2016; Calderhead, 2012). The conditions for asymptotically valid samplers are mild, but efficiently exploring high-dimensional distributions remains a major challenge. Modern methods typically convert the problem into numerical integration of an augmented dynamic system, based e.g. on Langevin diffusion (Roberts and Tweedie, 1996; Roberts and Stramer, 2002; Green et al., 2015), Hamiltonian Dynamics (Duane et al., 1987; Neal et al., 2011; Betancourt, 2017) or Lagrangian dynamics (Fang et al., 2014; Lan et al., 2015).

The augmented dynamics combine the logarithm of the target distribution with a kinetic term and simulate the time-evolution of the system. By using gradient information to drive the evolution they both convergence to the target distribution faster and improve exploration of the likely set. However, high-dimensional problems with strong correlations between individual dimensions and/or vastly different marginal variances are still challenging (Roberts and Stramer, 2002; Betancourt, 2017). To an extent this can be addressed by tuning a mass matrix M controlling the kinetic energy to globally de-correlate the parameter's dependency. This is equivalent to changing the *metric* on the parameter space, but still assuming some Euclidean metric (Neal et al., 2011). However, every global metric is necessarily a compromise between efficiency in regions of low curvature and accurate exploration of regions of high curvature.

Geometric MCMC algorithms (Girolami and Calderhead, 2011; Xifara et al., 2014; Lan et al., 2015; Betancourt, 2017; Beskos et al., 2017) use differential geometry to account for local curvature, replacing the mass matrix M with position-dependent matrix $G(\boldsymbol{x})$ that is the metric tensor of a suitable Riemannian manifold. Accounting for the local curvature improves the efficiency of the sampler especially in high-curvature regions (see Xifara et al., 2014; Girolami and Calderhead, 2011; Beskos et al., 2017, for many examples). The choice of the manifold and hence the metric is free, but existing literature focuses almost solely on the manifold and metric induced by the Fisher Information (FI) matrix of an underlying probabilistic model (Schervish, 2011). It is a natural choice that can be de-

Proceedings of the 25$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

rived from local Kullback-Leibler divergence, but only applicable for the specific case of posterior sampling as it is derived from a probabilistic model which mimics random variation in real data-sets.

The improved exploration comes with significant computational cost, and hence geometric MCMC methods are not widely used in practice. As the metric tensor is position-dependent, we now need to compute and invert it in every step of the numerical integration, sometimes several times. Already forming the FI matrix is demanding as it requires expected second derivatives of the log density of the model, and inversion has cubic complexity in the problem dimensionality D.

We present a new Riemannian metric that also relates to local curvature of the distribution but that is computationally efficient and generally applicable, based on pure geometric reasoning rather than relying on statistical properties of a model. We propose an embedding based on the graph of the target distribution $\pi_{\boldsymbol{X}}$ as a manifold in a higher-dimensional Euclidean space, using a scaled Monge parameterization $\Xi(\boldsymbol{x}) = (\boldsymbol{x}, \alpha \log \pi_{\boldsymbol{X}}(\boldsymbol{x}))$. The manifold is generated by the *Monge patch* embedding named after Gaspard Monge, one of the inventors of differential geometry (O'Neill, 2006). This operation defines a Riemannian manifold with a natural metric tensor. The metric tensor $G_M(\boldsymbol{x})$ is expressed as rank-one perturbation of the identity matrix with the rank-one term being the outer product of the gradients of the log target density. Consequently, it has efficient closed-form inverse as well as efficient closed-form determinant, offering significant computational savings.

The new metric captures the local curvature of the target density directly via simultaneous relations between the second fundamental form of the manifold, the Hessian of the target density and the Christoffel symbols. It provides similar advantages in exploration of complex regions of the distribution as the Fisher metric, and in expectation can be interpreted as regularized FI matrix. The control parameter α allows fine-tuning the embedding and the metric for overall computational efficiency.

The metric is general and applicable for various geometric MCMC algorithms. We demonstrate it with the Lagrangian Monte Carlo (LMC) (Lan et al., 2015). Compared to Riemannian manifold HMC (RMHMC), LMC has the advantage of an explicit numerical integrator that only requires two matrix inversions per iteration. However, it is not symplectic (volume-preserving) and hence requires also computing determinant adjustment for the proposals acceptance check. The costly computation of the determinants and *Christoffel symbols* required for the numerical integrator have limited the interest in LMC, but in our metric both can be computed efficiently. In our experiments, LMC in the Monge metric outperforms algorithms operating in Euclidean or Fisher metrics.

2 BACKGROUND

We briefly summarize Hamiltonian Monte Carlo (HMC) as an example algorithm using augmented dynamics and discuss the role of metrics for the simulation. We then provide the foundations of differentiable manifolds, introducing concepts relating to curvatures of the manifolds and their relationship to metrics.

2.1 Hamiltonian Monte Carlo and Metrics

Hamiltonian Monte Carlo (Neal et al., 2011) provides samples from a probability distribution $\pi_{\boldsymbol{X}}(\boldsymbol{x})$ by simulating the time-evolution of the Hamiltonian

$$H(\boldsymbol{x}, \boldsymbol{p}) = -\log \pi_{\boldsymbol{X}}(\boldsymbol{x}) + \frac{1}{2} \log |M| + \frac{1}{2} \boldsymbol{p}^T M^{-1} \boldsymbol{p}$$

where the momentum variables $\boldsymbol{p}$ are sampled (typically) from a normal distribution. A new proposal is generated by simulating the trajectory of the pair $(\boldsymbol{x}, \boldsymbol{p})$ using numeric integration that alternates between updates for the position $\boldsymbol{x}$ and the momentum $\boldsymbol{p}$. This simulation is done for L iterations before determining whether the proposal is accepted. Variants of HMC, such as the No-U-Turn-Sampler (NUTS; Hoffman and Gelman, 2014), are today the most common methods for statistical inference and are widely implemented in probabilistic programming languages.

The efficiency of HMC depends on the choice of the *mass matrix* or *metric tensor* M, which is typically tuned during warm-up. For instance, M proportional to the covariance of the target distribution effectively de-correlates the dimensions and improves exploration (Neal et al., 2011). However, no global metric can help coping with differences in local stretching or squeezing of the manifold, and hence techniques like explicit reparameterization are used for complex distributions (Papaspiliopoulos et al., 2007).

Rather than using a global metric, we can conduct HMC on Riemannian manifolds (RMHMC) by using a position-dependent metric tensor $G(\boldsymbol{x})$ instead (Girolami and Calderhead, 2011). This allows coping with changes in local curvature, assuming the metric is chosen suitably. This extension results in an implicit numerical integrator since two of the updates have the

same variable on both sides:

$$\begin{aligned}\boldsymbol{p}^{(n+1/2)} &= \boldsymbol{p}^{(n)} - \frac{\epsilon}{2}\nabla_{\boldsymbol{x}} H\left(\boldsymbol{x}^{(n)}, \boldsymbol{p}^{(n+1/2)}\right), \\ \boldsymbol{x}^{(n+1)} &= \boldsymbol{x}^{(n)} + \frac{\epsilon}{2}\left[\nabla_{\boldsymbol{p}} H\left(\boldsymbol{x}^{(n)}, \boldsymbol{p}^{(n+1/2)}\right)\right. \\ &\qquad \left. + \nabla_{\boldsymbol{p}} H\left(\boldsymbol{x}^{(n+1)}, \boldsymbol{p}^{(n+1/2)}\right)\right], \\ \boldsymbol{p}^{(n+1)} &= \boldsymbol{p}^{(n+1/2)} - \frac{\epsilon}{2}\nabla_{\boldsymbol{x}} H\left(\boldsymbol{x}^{(n+1)}, \boldsymbol{p}^{(n+1/2)}\right).\end{aligned}$$

The solution of these equations requires matrix inversion during every iteration since $\nabla_{\boldsymbol{x}} H(\boldsymbol{x}, \boldsymbol{p}) = G(\boldsymbol{x})^{-1}\boldsymbol{p}$. Furthermore, the implicit equations are solved by a fixed-point iteration and hence there is a need of computing inverse matrices multiple times. Usually the metric is derived from FI, as explained in more detail in Section 2.3. MCMC chains in Fisher metric behave better compared to any Euclidean metric, but the extensive computational cost and difficulty of computing the metric has prevented wide-spread use of RMHMC. Paquet and Fraccaroa (2018) considered using the Hessian of the target density as the metric tensor as an alternative, but it has the same computational cost.

In Section 4 we will consider in detail a variant of RMHMC, Lagrangian Monte Carlo (Lan et al., 2015), that avoids implicit equations but requires calculation of determinants and Christoffel symbols instead.

2.2 Differential Geometry Preliminaries

Our point of departure is the notion of a *differentiable manifold*. We call a set $\mathcal{M}$ a *differentiable manifold* of dimension m (in short manifold) if together with bijective mappings (also called parametrizations or system of coordinates) $\Xi_i(x_1, \ldots, x_m) : \mathcal{X}_i \subset \mathbb{R}^m \to \mathcal{M}$ where $\mathcal{X}_i$ is a chart, they satisfy,

(a) $\bigcup_i \Xi_i(\mathcal{X}_i) = \mathcal{M}$

(b) For each i, j, $\Xi_i(\mathcal{X}_i) \bigcap \Xi_j(\mathcal{X}_j) \neq \emptyset$ and that $\Xi_i^{-1} \circ \Xi_j$ are differentiable mappings.

The family $(\Xi_i, \mathcal{X}_i)$ is also called a differentiable structure on $\mathcal{M}$, and allow us to extend notions of the differential calculus in Euclidean space to more general spaces such as some abstract set $\mathcal{M}$ (e.g. a family of probability distributions).

One of the aims of differential geometry is to enable characterizing the *rate of change* for computing derivatives on $\mathcal{M}$ intrinsically, without referring to any external coordinate space. For this we need the notion of a *tangent space*. To do so, consider two overlapping curves that trace out different paths on $\mathcal{M}$ but intersect in a unique point $p \in \mathcal{M}$. With the aid of two distinct charts for each path, we denote $\gamma_1 := \Xi(t) : I_1 \subset \mathbb{R} \to \mathcal{M}$ and $\gamma_2 := \Pi(t) : I_2 \subset \mathbb{R} \to \mathcal{M}$. By taking the usual derivatives w.r.t to the variable t at t_1 such that $\gamma_1(t_1) = p$ and for the second curve at t_2 such that $\gamma_2(t_2) = p$, we obtain

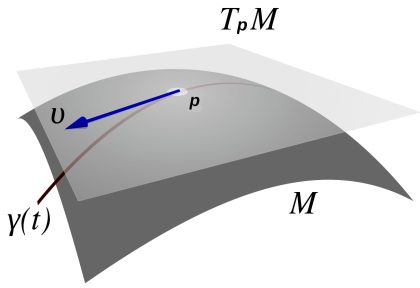

Figure 1: Manifold $\mathcal{M}$ and its tangent space $T_p\mathcal{M}$, with v being a tangent vector of the curve $\gamma(t)$.

$$\dot{\gamma}_1 = \sum_{k=1}^{n} \frac{\mathrm{d}x_k}{\mathrm{d}t}\frac{\partial}{\partial x_k}\Xi \quad \text{and} \quad \dot{\gamma}_2 = \sum_{k=1}^{n} \frac{\mathrm{d}y_k}{\mathrm{d}t}\frac{\partial}{\partial y_k}\Pi.$$

Because of condition (b) in the manifold definition, the set of vectors $\{\partial/\partial_k\, \Xi\}_k$ and $\{\partial/\partial_k\, \Pi\}_k$ span the same linear subspace of $\mathbb{R}^n$ at $p \in \mathcal{M}$, differing only in the basis vectors. Henceforth, we call this linear subspace as *tangent space* at p, in short $T_p\mathcal{M}$. To see this more clearly, define a new chart $\psi = \Xi \circ h : I_3 \to \mathcal{M}$ where $h = \Xi^{-1} \circ \Pi$ and note that

$$\frac{\partial}{\partial y_k}\psi = \frac{\partial}{\partial y_k}\Xi \circ h = \sum_{c=1}^{n} \frac{\mathrm{d}x_c}{\mathrm{d}y_k}\frac{\partial}{\partial x_c}\Xi$$

for $k = 1, \ldots, n$.

Since the Jacobian of transformation h does not vanish for any $p \in \mathcal{M}$, we have $\{\partial/\partial y_k\ \psi\}_k$ and $\{\partial/\partial x_k\ \Xi\}_k$ as the only different basis vectors of the set $T_p\mathcal{M}$. Furthermore, we can define an inner product of elements of the space $T_p\mathcal{M}$ as $g : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$ and then note that g is invariant with respect of different charts of the manifold. Gauss (1902) noted the implications of this alreay in 1827: If we want to study the curvature (how much $\mathcal{M}$ deviates from a Euclidean space, or how it stretches and squeezes locally) of the set $\mathcal{M}$, it is enough to know the metric g – we do not need the exact form of the charts.

2.3 Riemannian Manifolds and Metrics

A *Riemannian manifold* is a manifold which associates for each point $p \in \mathcal{M}$ an inner product g (symmetric, bilinear and positive-definite) for the vectors in $T_p\mathcal{M}$. For a given parametrization Ξ and tangents $v = s_i\ \partial/\partial_i\ \Xi$ and $u = t_i\ \partial/\partial_i\ \Xi$, we have $g(u, v) = \langle u, v\rangle_p = \boldsymbol{s}^\top G(p)\boldsymbol{t}$ where the coefficients of

the metric g are the elements of the positive-definite matrix $G(p)$ given by the inner products

$$G_{i,j}(p) = \left\langle \frac{\partial}{\partial x_i}\Xi, \frac{\partial}{\partial x_j}\Xi \right\rangle_p \quad \text{and} \quad \boldsymbol{s}, \boldsymbol{t} \in \mathbb{R}^n.$$

Such a matrix is called *metric tensor*. In this way Riemannian manifolds can be directly defined by a differentiable structure on a set $\mathcal{M}$ and a positive-definite matrix G at each $p \in \mathcal{M}$, without reference to any specific system of coordinates.

One particular Riemannian metric used broadly in statistics and machine learning uses the Fisher information matrix as the metric tensor (Amari et al., 2019; Girolami and Calderhead, 2011; Lan et al., 2015). In context of MCMC, it provides a metric that accounts for a probabilistic model for data, but that requires computing the expectation of the Hessian that is often difficult (Pawitan, 2001). If the probabilistic model satisfies suitable regularity conditions (Schervish, 2011), we can express the metric as

$$\begin{aligned} G_{i,j}(p) &= \mathbb{E}_Y\left(\frac{\partial}{\partial p_i}\log \pi_Y(Y|p)\frac{\partial}{\partial p_j}\log \pi_Y(Y|p)\right) \\ &= -\mathbb{E}_Y\left(\frac{\partial^2}{\partial p_i \partial p_j}\log \pi_Y(Y|p)\right) \\ &= -\int_\Omega \frac{\partial^2}{\partial p_i \partial p_j}\log \pi_Y(y|p)\pi_Y(y|p)\,\mathrm{d}y. \end{aligned}$$

where Y is a random variable (data yet to be observed), y is the observed data and Ω is the space of all possible data outcomes. We call the resulting metric *Fisher metric* and denote the metric tensor by $G_F(\cdot)$.

FI characterizes the lower bound of the variance of unbiased estimators and it can also be derived from the Kullback-Leibler divergence between two probability distribution of the same family and hence offers interesting theoretical connections, but ultimately the choice has still been primarily justified by good empirical properties (Girolami and Calderhead, 2011; Betancourt, 2017). Finally, it is only applicable for posterior sampling and not for general sampling problems.

3 MONGE PATCH AND METRIC

Our goal is to form a metric that accounts for local curvature of the target distribution, but is (a) computationally efficient and (b) applicable for general target densities, rather than requiring an underlying probabilistic model for forming the metric. We seek for such a metric based on pure geometric principles of hyper-surfaces embedded in higher-dimensional Euclidean spaces (Gauss, 1902; Do Carmo and Flaherty, 1992; Do Carmo, 2017).

Let $\mathcal{M}$ and $\mathcal{N}$ be manifolds of dimension m and n respectively with $m \leq n$. We say $\mathcal{M}$ is an *embedding* if for a differentiable mapping $\varphi : \mathcal{M} \to \mathcal{N}$ the differential $\mathrm{d}\varphi_p(v) : T_p\mathcal{M} \to T_{\varphi(p)}\mathcal{N}$ is injective and φ is a bijection. Consider a target probabilistic model $\boldsymbol{X} \sim \pi_{\boldsymbol{X}}(\cdot)$ from which we would like to obtain samples from and denote its logarithm as $\ell(\boldsymbol{x}) = \log \pi_{\boldsymbol{X}}(\boldsymbol{x}) : \mathcal{X} \subseteq \mathbb{R}^D \to \mathbb{R}$.

We then represent the manifold $\mathcal{M}$ as an embedding using the target distribution to define the embedding in $\mathcal{N}$ (which is a subspace of the $D+1$ dimensional Euclidean space) with φ as the identity function. The embedding is,

$$\Xi(\boldsymbol{x}) = (\boldsymbol{x}, \alpha\ell(\boldsymbol{x})) \in \mathcal{M}$$

and thus $\mathcal{M} = \{(z_1, \ldots, z_{D+1}) =: \Xi(\boldsymbol{x}) \in (\mathcal{X} \times \mathbb{R}) \subset \mathbb{R}^{D+1} : \boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^D\}$ is the embedded manifold via the *scaled Monge patch* Ξ with $\alpha \geq 0$. This extends the Monge parameterization $(\boldsymbol{x}, \ell(\boldsymbol{x}))$ with a parameter α that will be used for controlling the curvature information of the induced metric. Alternatively, we can interpret this as embedding of the logarithm of the tempered distribution $\pi_{\boldsymbol{X}}(\boldsymbol{x})^\alpha$. This embedding is arbitrary in the sense that we have no specific rationale for the choice, but as will be shown next it induces a metric that has several desirable properties.

For a tangent $v = \sum_{i=1}^n s_i\, \partial/\partial x_i\, \Xi(\boldsymbol{x}) \in T_p\mathcal{M}$ where

$$\frac{\partial}{\partial x_i}\Xi(\boldsymbol{x}) = \Big(\underbrace{0, \ldots, 1, \ldots, 0}_{i^{th}\text{ position}}, \alpha\frac{\partial}{\partial x_i}\ell(\boldsymbol{x})\Big),$$

we obtain that $\mathrm{d}\varphi_p(v)$ is injective $\forall p$. Therefore, as defined previously for tangents $u, v \in T_p\mathcal{M}$ the metric induce by this embedding becomes,

$$\begin{aligned} & g_M(u,v) = \boldsymbol{s}^\top G_M(p)\boldsymbol{t} \\ &= \boldsymbol{s}^\top \begin{bmatrix} \sum_{d=1}^D \frac{\partial \Xi_d}{\partial x_1}\frac{\partial \Xi_d}{\partial x_1} & \cdots & \sum_{d=1}^D \frac{\partial \Xi_d}{\partial x_1}\frac{\partial \Xi_d}{\partial x_D} \\ \vdots & \ddots & \vdots \\ \sum_{d=1}^D \frac{\partial \Xi_d}{\partial x_D}\frac{\partial \Xi_d}{\partial x_1} & \cdots & \sum_{d=1}^D \frac{\partial \Xi_d}{\partial x_D}\frac{\partial \Xi_d}{\partial x_D} \end{bmatrix} \boldsymbol{t} \\ &= \boldsymbol{s}^\top \begin{bmatrix} 1 + \alpha^2 \frac{\partial}{\partial x_1}\ell(\boldsymbol{x})^2 & \cdots & \alpha^2 \frac{\partial}{\partial x_1}\ell(\boldsymbol{x})\frac{\partial}{\partial x_n}\ell(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ \alpha^2 \frac{\partial}{\partial x_n}\ell(\boldsymbol{x})\frac{\partial}{\partial x_1}\ell(\boldsymbol{x}) & \cdots & 1 + \alpha^2 \frac{\partial}{\partial x_n}\ell(\boldsymbol{x})^2 \end{bmatrix} \boldsymbol{t} \end{aligned}$$

where each $\boldsymbol{x}$ in the support of our target (or the domain of the target distribution) uniquely determines a specific point p on the manifold $\mathcal{M}$. Hence we denote the metric tensor in matrix form as

$$G_M(\boldsymbol{x}) = I_D + \alpha^2 \nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top, \tag{1}$$

slightly abusing the notation to express it directly in terms of $\boldsymbol{x}$. The matrix (1) is symmetric and positive-definite and hence the pair $(\mathcal{M}, g_M)$ is a Riemannian manifold. We call the resulting metric the *Monge metric*.

3.1 Interpretation

The local geometric properties of manifolds have two important quantities with direct interpretation: the *first fundamental form* which relates to lengths of the curves on $\mathcal{M}$ and the *second fundamental form* that relates to the curvature, i.e., how much the manifold locally deviates from the Euclidean space (or the tangent plane). Both the Fisher metric and the Monge metric are connected to the first fundamental form as they tell us a way to measure lengths of curves on $\mathcal{M}$. In some statistics literature the Fisher metric has been linked with the idea of curvature, see Calderhead (2012), Girolami and Calderhead (2011) and Paquet and Fraccaroa (2018), due to the its definition as the expected value of the Hessian matrix. However, the Monge metric has natural geometric reasoning as it additionally has a direct notion of curvature due to the clear manifestation of Hessian matrix of ℓ in the second fundamental form on the embedded Riemannian manifold $\mathcal{M}$.

The second fundamental form $g_* = \langle \ddot{\gamma}_1(\boldsymbol{x}(t_1)), N(\boldsymbol{x}) \rangle$ is formally defined as the inner product between the acceleration of a curve on the manifold, $\ddot{\gamma}_1(\boldsymbol{x}(t_1))$, and the normal vector

$$N(\boldsymbol{x}) = -\frac{\left(\alpha \frac{\partial}{\partial x_1}\ell(\boldsymbol{x}), \ldots, \alpha \frac{\partial}{\partial x_D}\ell(\boldsymbol{x}), 1\right)}{\sqrt{1+\alpha^2||\nabla \ell(\boldsymbol{x})||^2}},$$

for every $p \in \mathcal{M}$. After algebraic manipulation and cancelling the terms of $\dot{\gamma}_1(\boldsymbol{x}(t_1))$ orthogonal to $N(\boldsymbol{x}(t_1))$ and dropping the notation of the argument t_1, we obtain (see Pressley, 2010; Do Carmo, 2017)

$$g_* = \boldsymbol{s}^\top \left\{ \left\langle \frac{\partial^2}{\partial x_i \partial x_j} \Xi(\boldsymbol{x}), N(\boldsymbol{x}) \right\rangle \right\}_{i,j} \boldsymbol{s}$$

Thus we can see that

$$g_* = \boldsymbol{s}^\top \frac{\alpha}{c} \underbrace{\begin{bmatrix} \frac{\partial^2}{\partial x_1 \partial x_2}\ell(\boldsymbol{x}) & \cdots & \frac{\partial^2}{\partial x_1 \partial x_D}\ell(\boldsymbol{x}) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2}{\partial x_D \partial x_1}\ell(\boldsymbol{x}) & \cdots & \frac{\partial^2}{\partial x_D \partial x_D}\ell(\boldsymbol{x}) \end{bmatrix}}_{H(\boldsymbol{x})} \boldsymbol{s},$$

where $c = \sqrt{1+\alpha^2||\nabla \ell(\boldsymbol{x})||^2}$ and $H(\boldsymbol{x}) = \nabla^2 \ell(\boldsymbol{x})$ is the Hessian matrix of the logarithm of the target distribution. The curvature of the Monge metric, as measured by the second fundamental form, is hence a scaled version of the Hessian that encodes local scaling and stretching information. This provides an intuitive and natural interpretation for the metric, even though it was induced by a seemingly arbitrary embedding.

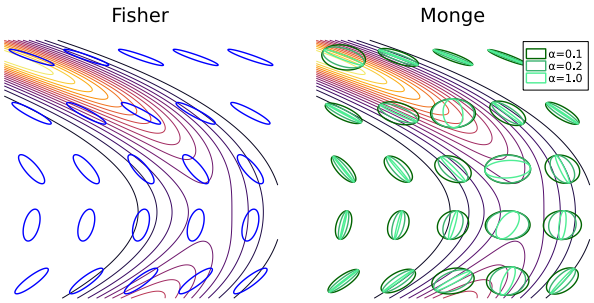

Figure 2: Illustration of the metric on the Rosenbrock distribution. The Monge metric captures the shape of the distribution in similar manner as the Fisher metric. The α parameter controls the embedding and scales the metric; with small α (dark green) it is close to Euclidean and with large α (light green) we get very elongated metric for areas of high curvature.

The Monge metric is derived from a different perspective than the Fisher metric, but they are related. For the case where the logarithm of the target distribution is $\ell(\boldsymbol{x}) = \log \pi_Y(Y|\,\boldsymbol{x})$ and $\pi_Y(\cdot|\,\boldsymbol{x})$ is a model that defines the random generating mechanism of the data, we obtain $\alpha^{-2}\,\mathbb{E}_Y\left(G_M(\boldsymbol{x})\right) = \alpha^{-2} I_D + G_F(\boldsymbol{x})$ by computing the expectation of the negative Hessian over Y. That is, in expectation the metric can be seen as biased or regularized estimator for FI, so that inverse α controls the regularization.

Figure 2 illustrates the Monge and Fisher metrics for a banana-shaped posterior ($\pi_{\boldsymbol{X}}(x_1, x_2) \propto \prod_i \mathcal{N}(y_i|x_1 + x_2^2, \sigma_y^2)\mathcal{N}(x_1|0, \sigma^2)\mathcal{N}(x_2|0, \sigma^2)$ with $\sigma_y^2 = \sigma^2 = 0.5$ and $n = 10$ observations y_i). Here the Fisher metric is constant w.r.t. to the x_1 coordinate (Bornn and Cornebise, 2011), whereas the Monge metric is bivariate. The Monge metric becomes identity at the mode, and flattens towards spherical Euclidean metric for $\alpha \to 0$. Outside the mode it behaves similarly to the Fisher metric, but for large α is more elongated. The question of optimal α is an empirical one.

3.2 Computation

Fast Inverse and Determinants The metric (1) has efficient inverse via Sherman-Morrison lemma as

$$G_M(\boldsymbol{x})^{-1} = I_D - \alpha^2 \frac{\nabla \ell(\boldsymbol{x}) \nabla \ell(\boldsymbol{x})^\top}{1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2}$$

with $\mathcal{O}(D^2)$ complexity. Similarly, the determinant is

$$\det G_M(\boldsymbol{x}) = 1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2$$

with linear complexity. Both are significant improvements over $\mathcal{O}(D^3)$ for general operations in the original LMC formulation (Lan et al., 2015).

Fast Christoffel symbols The *Christoffel symbols* $\Gamma^k_{i,j}(\boldsymbol{x})$ measure the magnitude of the basis vector $\partial/\partial x_k \Xi$ in the rate of change of the vector $\partial/\partial x_j \Xi$ at the direction of $\partial/\partial x_i \Xi$ for every point $\Xi(\boldsymbol{x}) = p$ of the manifold. Formally the Christoffel symbols are defined as the coefficients of the *Levi-Civita connection* of the Riemannian manifold.

They are required for some algorithms operating on Riemannian manifolds and, if not obtained in closed-form, their computation might incur a significant computational cost in general case. Since $G_M(\boldsymbol{x})$ is obtained using an embedding Ξ, we can re-write the Christoffel symbols following (Do Carmo and Flaherty, 1992, page 56, equation (10)) as

$$\begin{aligned}\Gamma^k_{i,j}(\boldsymbol{x}) &= \sum_{l=1}^{D} G^{-1}_{k,l}(\boldsymbol{x}) \left\langle \frac{\partial^2}{\partial x_i \partial x_j}\Xi, \frac{\partial}{\partial x_l}\Xi \right\rangle \\ &= \frac{\alpha^2}{1+\alpha^2\left\|\nabla \ell(\boldsymbol{x})\right\|^2} \frac{\partial}{\partial x_k}\ell(\boldsymbol{x}) \frac{\partial^2}{\partial x_i \partial x_j}\ell(\boldsymbol{x})\end{aligned}$$

using the elements $\frac{\partial^2}{\partial x_i \partial x_j}\ell(\boldsymbol{x})$ of the second fundamental form. Even though the metric tensor only requires the gradients, we see that second-order derivatives are needed for computing the Christoffel symbols.

4 LMC ON MONGE PATCHES

The Monge metric is general and applicable for several geometric MCMC methods. We demonstrate it here for Lagrangian Monte Carlo (Lan et al., 2015), providing detailed derivations in the Supplement.

Lagrangian Monte Carlo As explained in Section 2, RMHMC involves two implicit equations that require fixed-point iterations and hence multiple inversions of the metric tensor during every update. An explicit integrator can be developed by switching to *Lagrangian dynamics* and working with *velocity* $\boldsymbol{v} = G(\boldsymbol{x})^{-1}\boldsymbol{p}$ instead of the momentum, resulting in Riemannian manifold Lagrangian Monte Carlo (Lan et al., 2015). The energy functional and dynamics are

$$\begin{aligned} E(\boldsymbol{x},\boldsymbol{v}) &= -\log \pi_{\boldsymbol{X}}(\boldsymbol{x}) - \tfrac{1}{2}\log|G(\boldsymbol{x})| + \tfrac{1}{2}\boldsymbol{v}^\top G(\boldsymbol{x})\boldsymbol{v}, \\ \dot{\boldsymbol{x}} &= \boldsymbol{v}, \\ \dot{\boldsymbol{v}} &= -\frac{1}{2}G(\boldsymbol{x})^{-1}\big[2\partial_{\boldsymbol{x}}G(\boldsymbol{x}) - (\partial_{\boldsymbol{x}} vec\ G(\boldsymbol{x}))^\top\big](\dot{\boldsymbol{x}}\otimes\dot{\boldsymbol{x}}) \\ &\quad - G(\boldsymbol{x})^{-1}\nabla\phi(\boldsymbol{x}), \end{aligned} \tag{2}$$

where the notation $\partial_{\boldsymbol{x}}A = [\partial_{x_1}A \cdots \partial_{x_D}A]$. Observe that the k^{th} row of the matrix $\frac{1}{2}G(\boldsymbol{x})^{-1}\big[2\partial_{\boldsymbol{x}}G(\boldsymbol{x}) - (\partial_{\boldsymbol{x}} vec\ G(\boldsymbol{x}))^\top\big]$ is $(vec\ \Gamma^k)^\top$ where $\Gamma^k_{i,j}(\boldsymbol{x}) = \frac{1}{2}\sum_l G^{k,l}(\frac{\partial}{\partial x_i}G_{i,j} + \frac{\partial}{\partial x_j}G_{i,l} - \frac{\partial}{\partial x_l}G_{i,j})$ are the Christoffel symbols (See Arvanitidis et al., 2018, for similar formulation). The matrix elements $G_{i,j}$ and $G^{i,j}$ are the elements of the matrix $G(\boldsymbol{x})$ and its inverse respectively, and $\nabla\phi(\boldsymbol{x}) = -\nabla\log\pi_{\boldsymbol{X}}(\boldsymbol{x}) + \frac{1}{2}\nabla\log\det G(\boldsymbol{x})$.

The explicit integrator repeats L_F times the updates

$$\begin{aligned} \boldsymbol{v}^{(n+1/2)} &= A^{-1}_{n,n}\left[\boldsymbol{v}^{(n)} - \frac{\varepsilon}{2}G(\boldsymbol{x}^{(n)})^{-1}\nabla\phi(\boldsymbol{x}^{(n)})\right] \\ \boldsymbol{x}^{(n+1)} &= \boldsymbol{x}^{(n)} + \varepsilon\,\boldsymbol{v}^{(n+1/2)} \\ \boldsymbol{v}^{(n+1)} &= A^{-1}_{n+1,n+1/2}\Big[\boldsymbol{v}^{(n+1/2)} \\ &\qquad - \frac{\varepsilon}{2}G(\boldsymbol{x}^{(n+1)})^{-1}\nabla\phi(\boldsymbol{x}^{(n+1)})\Big] \end{aligned} \tag{3}$$

where $\Omega(\boldsymbol{x},\boldsymbol{v})$ is a matrix whose (i,j) element is given by $\sum_k v_k\Gamma^i_{k,j}(\boldsymbol{x})$ and $A_{n_1,n_2} = I_D + \frac{\varepsilon}{2}\Omega(\boldsymbol{x}^{(n_1)},\boldsymbol{v}^{(n_2)})$.

The integrator is not volume-preserving and we need determinant adjustment for the acceptance probability $\alpha_{LMC} = \min\{1, \exp(-E_{\text{diff}})|\det J|\}$ where $E_{\text{diff}} = E(\boldsymbol{x}^{(L_F+1)},\boldsymbol{v}^{(L_F+1)}) - E(\boldsymbol{x}^{(1)},\boldsymbol{v}^{(1)})$ and

$$\begin{aligned}\det J = \prod_{n=1}^{L_F}\Bigg(&\frac{\det\big(G(\boldsymbol{x}^{(n+1)}) - \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n+1)},\boldsymbol{v}^{(n+1)})\big)}{\det\big(G(\boldsymbol{x}^{(n+1)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n+1)},\boldsymbol{v}^{(n+1/2)})\big)} \\ &\times \frac{\det\big(G(\boldsymbol{x}^{(n)}) - \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n)},\boldsymbol{v}^{(n+1/2)})\big)}{\det\big(G(\boldsymbol{x}^{(n)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n)},\boldsymbol{v}^{(n)})\big)}\Bigg),\end{aligned}$$

where the matrix $\tilde{\Omega}(\boldsymbol{x},\boldsymbol{v}) = G(\boldsymbol{x})\Omega(\boldsymbol{x},\boldsymbol{v})$.

LMC does not require fixed-point iterations and hence only needs two inversions per step, but the computational advantage is lost due to computation of the determinants and the $\mathcal{O}(D^3)$ Christoffel symbols. The complexity of both RMHMC and LMC in a general metric is $\mathcal{O}(D^3)$, and their relative speed depends on the problem.

LMC in Monge Metric In the Monge metric $G_M(\boldsymbol{x})$ the energy becomes

$$\begin{aligned}E(\boldsymbol{x},\boldsymbol{v}) = &-\ell(\boldsymbol{x}) - \tfrac{1}{2}\log(1+\alpha^2\left\|\nabla\ell(\boldsymbol{x})\right\|^2) \\ &+ \tfrac{1}{2}\left\|\boldsymbol{v}\right\|^2 + \tfrac{\alpha^2}{2}\left\langle\nabla\ell(\boldsymbol{x}),\boldsymbol{v}\right\rangle^2.\end{aligned}$$

In the dynamical system (2) we retain $\dot{\boldsymbol{x}} = \boldsymbol{v}$ and for the velocity we have

$$\begin{aligned}\dot{\boldsymbol{v}} = &-\frac{\alpha^2}{1+\alpha^2\left\|\nabla\ell(\boldsymbol{x})\right\|^2}\big(\nabla\ell(\boldsymbol{x})\ (vec\ H(\boldsymbol{x}))^\top\big)(\boldsymbol{v}\otimes\boldsymbol{v}) \\ &- G_M(\boldsymbol{x})^{-1}\left(\frac{\alpha^2 H(\boldsymbol{x})}{1+\alpha^2\left\|\nabla\ell(\boldsymbol{x})\right\|^2} - I_D\right)\nabla\ell(\boldsymbol{x}).\end{aligned}$$

For an initial velocity $\boldsymbol{v}$ and initial position $\boldsymbol{x}$ these keep the energy constant. The Hessian $H(\boldsymbol{x})$ appears here due to the Christoffel symbols, even though

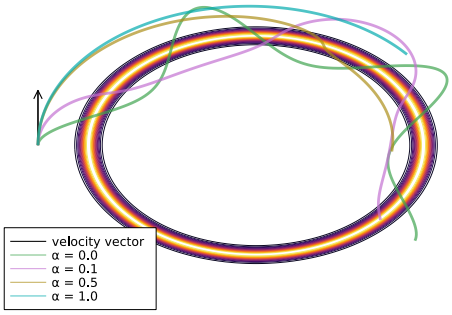

Figure 3: Geodesic paths of Lagrangian dynamics in Monge metrics of different α for fixed initial velocity. Note that $\alpha = 0$ means Euclidean metric.

the metric only involves gradients. We illustrate the geodesics for various α in Figure 3, computed for a ring distribution. For $\alpha = 0$ the metric reduces to Euclidean and the geodesic paths fluctuate around the typical set (see Betancourt, 2017, for detailed discussion), whereas for large α they resemble clear orbits.

After fairly extensive simplification, the update equations (3) in Monge metric can be written as in Table 1. The full derivation and a pseudo-code for the algorithm is provided in the Supplement. These updates are somewhat complicated, but free of matrix-matrix products and free of explicit matrix inversions. The proposal acceptance probability simplifies in a similar manner using

$$\det\left(G(\boldsymbol{x}) \pm \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x},\boldsymbol{v})\right) = 1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2 \pm \frac{\alpha^2 \varepsilon}{2}\langle \nabla \ell(\boldsymbol{x}), H(\boldsymbol{x})\,\boldsymbol{v}\rangle.$$

The computation for one pass of the numerical integrator is dominated by the formation of the gradient vector ($\mathcal{O}(D)$) and the Hessian matrix ($\mathcal{O}(D^2)$). Since they are called twice in each loop of the numerical integrator, the cost is dominated by $2L_F(\mathcal{O}(D) + \mathcal{O}(D^2))$ operations. The overall complexity is hence quadratic in D, not cubic as with the Fisher metric.

5 EXPERIMENTS

We evaluate LMC in Monge metric (LMC-Monge) in two example problems, a funnel distribution and posterior inference for logistic regression, but note that Figures 2 and 3 already demonstrated the metric in two other contexts. We compare against competing methods in Euclidean and Fisher metrics (when applicable). The experiments were ran on Intel i5-8250@1.6GHz laptop CPU. All experimental details and some additional illustrations are provided in the Supplement. The methods were implemented in `Julia` (Bezanson et al., 2017) and the implementation is available at `https://github.com/mahaa2/EmbeddedLMC`, providing both the inference algorithm itself as well as scripts for re-creating some of the experiments.

5.1 Funnel Distribution

We first show the metric helps in exploring areas of strong curvature, using the funnel distrubution by Neal (2003). The D-dimensional funnel is given by

$$\pi_{\boldsymbol{X}}(\boldsymbol{x}, a) = \prod_{i=1}^{D} \mathcal{N}(x_i|0, \text{softplus}(a))\mathcal{N}(a|\mu, \sigma_a^2), \quad (4)$$

where the marginal distribution of a is $\mathcal{N}(a|\mu, \sigma_a^2)$ and hence we can easily evaluate the quality of the marginal. We set $\mu = 0.0$ and $\sigma_a^2 = 15.0$, and use 60.000 samples. To illustrate the metric we use $\alpha = 1$ with accurate numeric integration with small step-length ϵ and L_F growing from 8 to 130 when increasing the dimensionality, adjusted by visual inspection.

Figure 4 demonstrates how LMC-Monge provides samples from the correct distribution but HMC in Euclidean metric does not, even when using the more advanced NUTS algorithm (Hoffman and Gelman, 2014) as implemented in `Turing.jl` (Ge et al., 2018). Both samplers have low autocorrelation, seen by observing the sampling chains, and hence the problems of the Euclidean sampler could easily go unnoticed in practice. Fisher metric is here not applicable since we are not conducting posterior inference, but rather sampling from the distribution itself for given parameters.

Figure 5 investigates the quality as a function of D, measured by approximating KL divergence between the true marginal and the MCMC approximation with $\sum_k [\log P(A_k)/\tilde{Q}(A_k)]P(A_k)$, where A_k is a histogram bin. LMC-Monge retains good accuracy for all D whereas NUTS gets progressive worse.

5.2 Logistic Regression

Having established the metric can explore well, we turn the attention to performance. We replicate the logistic regression experiment of Lan et al. (2015) on their largest data sets, using 20,000 samples (warm-up of 5,000). We also otherwise match their empirical setup, and in particular select ϵ and L_F to obtain acceptance probability in the range of 0.6-0.9 for each method. We evaluate the efficiency using the standard effective sample size (ESS) measure, but note that high ESS does not guarantee correct sampling.

Table 1: Numerical integration updates for LMC in the Monge metric. See Supplement for derivations.

$$
\begin{aligned}
\boldsymbol{v}^{(n+1/2)} &= \left[I_D - \frac{\nabla\ell(\boldsymbol{x}^{(n)})\big(\nabla\ell(\boldsymbol{x}^{(n)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n)})^\top H(\boldsymbol{x}^{(n)})\big)}{\big[\nabla\ell(\boldsymbol{x}^{(n)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n)})^\top H(\boldsymbol{x}^{(n)})\big]^\top \nabla\ell(\boldsymbol{x}^{(n)}) + \frac{1}{\alpha^2}} \right] \\
&\quad \times \left\{ \left[\left(\alpha^2 \nabla\ell(\boldsymbol{x}^{(n)})^\top \boldsymbol{v}^{(n)} + \frac{\varepsilon}{2}\right) I_D - \frac{\varepsilon\alpha^2}{2 + 2\alpha^2 \left\|\nabla\ell(\boldsymbol{x}^{(n)})\right\|^2} H(\boldsymbol{x}^{(n)}) \right] \nabla\ell(\boldsymbol{x}^{(n)}) + \boldsymbol{v}^{(n)} \right\} \\
\boldsymbol{x}^{(n+1)} &= \boldsymbol{x}^{(n)} + \varepsilon\, \boldsymbol{v}^{(n+1/2)} \\
\boldsymbol{v}^{(n+1)} &= \left[I_D - \frac{\nabla\ell(\boldsymbol{x}^{(n+1)})\big(\nabla\ell(\boldsymbol{x}^{(n+1)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n+1/2)})^\top H(\boldsymbol{x}^{(n+1)})\big)}{\big[\nabla\ell(\boldsymbol{x}^{(n+1)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n+1/2)})^\top H(\boldsymbol{x}^{(n+1)})\big]^\top \nabla\ell(\boldsymbol{x}^{(n+1)}) + \frac{1}{\alpha^2}} \right] \\
&\quad \times \left\{ \left[\left(\alpha^2 \nabla\ell(\boldsymbol{x}^{(n+1)})^\top \boldsymbol{v}^{(n+1/2)} + \frac{\varepsilon}{2}\right) I_D - \frac{\varepsilon\alpha^2}{2 + 2\alpha^2 \left\|\nabla\ell(\boldsymbol{x}^{(n+1)})\right\|^2} H(\boldsymbol{x}^{(n+1)}) \right] \nabla\ell(\boldsymbol{x}^{(n+1)}) + \boldsymbol{v}^{(n+1/2)} \right\}
\end{aligned}
$$

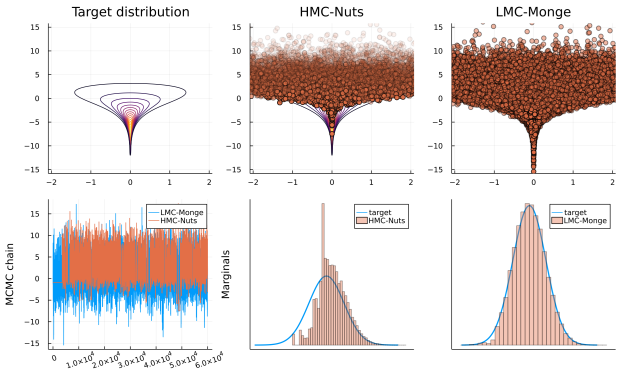

Figure 4: 1D funnel. LMC in Monge metric (right) explores the target distribution well, whereas HMC in Euclidean metric (middle) does not, even though both chains mix well (bottom left).

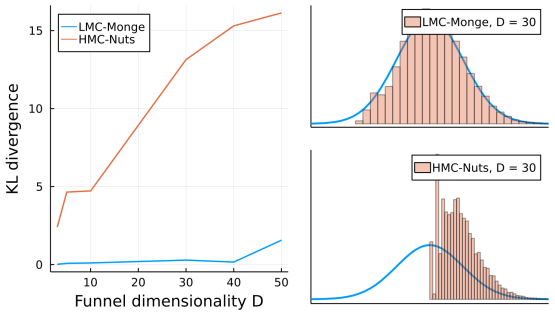

Figure 5: (Left) KL divergence between the true marginal $\mathcal{N}(a)$ and its estimate as a function of problem dimensionality. (Right) Marginals for $D = 30$.

Table 2 compares LMC-Monge against three baselines (using implementation and parameter settings of Lan et al. (2015)) that differ in terms of the metric: LMC and RHMC in Fisher metrics, and standard HMC in spherical Euclidean metric. The Riemannian methods have clearly higher ESS compared to the Euclidean HMC, and Monge metric behaves similarly to the Fisher metric but is faster. This validates our main claim. For completeness, we also show the results for NUTS in Euclidean metric even though direct comparison is not fair due to adaptive choice of L_F and ϵ that also helps in achieving high ESS.

Figure 6 shows the effect of the control parameter, using 3000 samples after warm-up of 500. The optimal choice depends on the data and often very small α are best, but we note that this does not necessarily mean the metric would be particularly close to Euclidean as the magnitude of $\nabla\ell(\boldsymbol{x})$ and $H(\boldsymbol{x})$ can also be large.

6 DISCUSSION

Augmented MCMC is the workhorse of probabilistic programming. Geometric MCMC algorithms offer theoretical advantages for complex distribution, but have slow updates and are rarely used in practice. We set out to resolve this problem, by providing a new Riemannian metric that still accounts for local curvature but is faster. The Monge metric, a natural metric for a Monge patch embedding, can be easily computed for every density based on gradients alone and has efficient inverse and determinant. Besides LMC, it could be used e.g. with the explicitly symplectic integrator for RMHMC (Cobb et al., 2019) or with manifold-adjusted Langevin Monte Carlo (Girolami and Calderhead, 2011).

We demonstrated the basic properties of the metric, but significant practical steps remain on the path to validating it in routine use. A practical tool for probabilistic programming or arbitrary sampling tasks would require a high-quality implementation and au-

Table 2: Logistic regression. The best method (ESS/sec) with constant L_F is indicated by boldface, and boldface italics marks cases where NUTS with adaptive L_F is overall the best. AP is average acceptance probability.

Data	Method	AP	ESS (min, mean, median)	time(s)	min(ESS)/s	mean(ESS)/s
Heart	LMC-Monge ($\alpha = 0.01$)	0.79	(15000, 15000, 15000)	34	**441**	**441**
$N = 270$	LMC-Fisher	0.76	(10347, 10848, 10724)	63	164	172
$D = 14$	RMHMC-Fisher	0.72	(6263, 7391, 7430)	90	69	82
	HMC-Euclidean	0.71	(378, 1164, 2624)	7	55	170
	HMC-Nuts	*0.94*	(*13804*, *14777*, *15000*)	*15*	***927***	***1055***
German	LMC-Monge ($\alpha = 0.01$)	0.81	(13390, 14949, 15000)	71	**194**	**210**
$N = 1000$	LMC-Fisher	0.70	(13762, 14932, 15000)	202	68	74
$D = 22$	RMHMC-Fisher	0.75	(14885, 14995, 15000)	252	49	59
	HMC-Euclidean	0.73	(766, 4803, 15000)	69	11	69
	HMC-Nuts	*0.70*	(*14168*, *14960*, *15000*)	*40*	***350***	***374***
Australian	LMC-Monge ($\alpha = 0.01$)	0.82	(1259, 12932, 15000)	52	24	**249**
$N = 690$	LMC-Fisher	0.75	(9636, 10464, 10443)	100	**96**	104
$D = 15$	RMHMC-Fisher	0.72	(7824, 9237, 9055)	134	58	69
	HMC-Euclidean	0.74	(1225, 4440, 10691)	18	65	246
	HMC-Nuts	*0.99*	(*1227*, *11715*, *15000*)	*54*	*22*	*216*

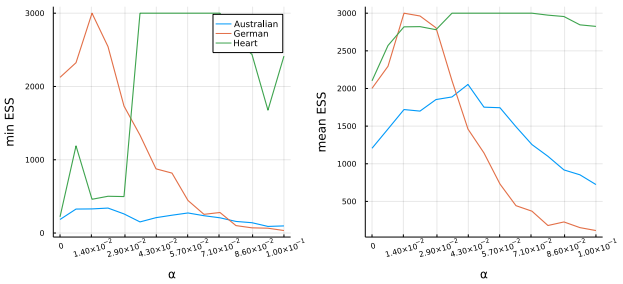

Figure 6: Relationship between α and sample efficiency. The optimal α depends on the problem.

tomatic means for adapting the controls parameter ϵ, L_F and α. We expect extension of the NUTS to Riemannian metrics (Betancourt, 2013) to help by offering automatic choice of the integration length, and α could be adapted during warm-up similar to how the Euclidean metric tensor M is often adapted, for instance based on gradient magnitudes. Finally, we see a need for more detailed theoretical analysis of the metric, e.g. along the lines of Brosse et al. (2018).

Acknowledgments

Hartmann and Klami were supported by the Academy of Finland (grant 345811 and Flagship programme: Finnish Center for Artificial Intelligence, FCAI), and Business Finland (MINERAL project).

Mark Girolami was supported by Engineering and Physical Sciences Research Council Grants [EP/R034710/1, EP/R018413/1, EP/R004889/1, EP/P020720/1] and a Royal Academy of Engineering Research Chair.

We thank ST John for software tips and Luiz Hartmann for discussions on topics in differential geometry.

References

Shun-ichi Amari, Ryo Karakida, and Masafumi Oizumi. Fisher information and natural gradient learning in random deep networks. In *22nd International Conference on Artificial Intelligence and Statistics*, pages 694–702. PMLR, 2019.

Georgios Arvanitidis, Lars Kai Hansen, and Søren Hauberg. Latent space oddity: on the curvature of deep generative models. In *International Conference on Learning Representations*, 2018.

Alexandros Beskos, Mark Girolami, Shiwei Lan, Patrick E. Farrell, and Andrew M. Stuart. Geometric mcmc for infinite-dimensional inverse problems. *Journal of Computational Physics*, 335:327–351, 2017. doi: 10.1016/j.jcp.2016.12.041.

Michael Betancourt. A general metric for Riemannian manifold Hamiltonian Monte Carlo. In Frank Nielsen and Frédéric Barbaresco, editors, *Geometric Science of Information*, pages 327–334, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

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

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.

Luke Bornn and Julien Cornebise. Discussion on "Riemann manifold Langevin and Hamiltonian Monte Carlo methods" by m. girolami and b. calderhead. *Journal of the Royal Statistical Society, Series B*, 73 (2):174–177, 2011.

Nicolas Brosse, Eric Durmus, Alainand Moulines, and Sotirios Sabanis. The tamed unadjusted langevin algorithm. *Stochastic Processes and their Applica-*

tions, page S0304414918305635, 2018. ISSN 0304-4149. doi: 10.1016/j.spa.2018.10.002.

Ben Calderhead. *Differential geometric MCMC methods and applications*. PhD thesis, University of Glasgow, 2012.

Oksana A. Chkrebtii, David A. Campbell, Ben Calderhead, and Mark A. Girolami. Bayesian solution uncertainty quantification for differential equations. *Bayesian Analysis*, 11(4):1239–1267, 12 2016.

Adam D. Cobb, Atılım Güneş Baydin, Andrew Markham, and Stephen J. Roberts. Introducing an explicit symplectic integration scheme for Riemannian manifold Hamiltonian Monte Carlo, 2019.

Manfredo P. Do Carmo. *Differential Geometry of Curves and Surfaces*. Dover Publications, 2nd edition, 2017.

Manfredo P. Do Carmo and Francis Flaherty. *Riemannian Geometry*. Mathematics. Theory & applications. Birkhäuser, 1st edition, 1992.

Simon Duane, A.D. Kennedy, Brian J. Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195:216–222, 1987.

Youhan Fang, J. M. Sanz-Serna, and Robert D. Skeel. Compressible generalized hybrid Monte Carlo. *The Journal of Chemical Physics*, 140, 2014.

Karl F. Gauss. General investigations of curved surfaces in 1827 and 1825. *Nature*, 66:316–317, 1902.

Hong Ge, Kai Xu, and Zoubin Ghahramani. Turing: a language for flexible probabilistic inference. In *International Conference on Artificial Intelligence and Statistics*, pages 1682–1690, 2018.

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

Peter J. Green, Krzysztof Łatuszyński, Marcelo Pereyra, and Christian P. Robert. Bayesian computation: a summary of the current state, and samples backwards and forwards. *Statistics and Computing*, 25:835–862, 2015. ISSN 0960-3174,1573-1375.

Matthew D. Hoffman and Andrew Gelman. The No-U-Turn Sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15(47):1593–1623, 2014.

Shiwei Lan, Vasileios Stathopoulos, Babak Shahbaba, and Mark Girolami. Markov chain Monte Carlo from Lagrangian dynamics. *Journal of Computational and Graphical Statistics*, 24:357–378, 2015. ISSN 1061-8600,1537-2715.

Radford Neal, Steve Brooks, Andrew Gelman, Galin L. Jones, and Xiao-Li Meng. MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*, 2(11):2, 2011.

Radford M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.

Barrett O'Neill. *Elementary Differential Geometry*. Elsevier, 2nd edition, 2006.

Omiros Papaspiliopoulos, Gareth O. Roberts, and Martin Sköld. A General Framework for the Parametrization of Hierarchical Models. *Statistical Science*, 22(1):59 – 73, 2007. doi: 10.1214/088342307000000014.

Ulrich Paquet and Marco Fraccaroa. An efficient implementation of Riemannian manifold Hamiltonian Monte Carlo for Gaussian process models. Technical report, Technical University of Denmark, Lyngby, Denmark, 2018.

Yudi Pawitan. *In All Likelihood: Statistical Modelling and Inference Using Likelihood*. Oxford University Press, USA, 2001. ISBN 0198507658,9780198507659.

Andrew Pressley. *Elementary Differential Geometry*. Springer Undergraduate Mathematics Series. Springer, 2 edition, 2010.

G. O. Roberts and O. Stramer. Langevin diffusions and Metropolis-Hastings algorithms. *Methodology and Computing in Applied Probability*, 4:337–357, 2002.

Gareth O. Roberts and Richard L. Tweedie. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2:341–363, 1996. ISSN 1350-7265.

Mark J. Schervish. *Theory of Statistics*. Springer Series in Statistics, 2011.

Tatiana Xifara, Chris Sherlock, Samuel Livingstone, Simon Byrne, and Mark Girolami. Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters*, 91:14–19, 2014. ISSN 0167-7152.

Supplementary Material: Lagrangian Manifold Monte Carlo on Monge Patches

A OVERVIEW

This Supplementary material provides additional derivations and details for the article *Lagrangian Manifold Monte Carlo on Monge Patches*. Sections B, C and D provide the derivations to complement Sections 3 and 4 of the main paper, whereas Section E provides the full experimental details, additional result plots and experiments.

B DERIVATIONS AND ADDITIONAL FORMULATIONS

In this section, we present derivations and mathematical simplifications that verify statements provided in the main paper and that are required for derivation of the LMC Monge update rules provided in Section C. Throughout this section, we consider shortened notation $L_\alpha(\boldsymbol{x}) = 1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2$ whenever convenient.

Christoffel symbols Section 3.2 provided a compact closed-form expression for the Christoffel symbols in the Monge metric. Starting with the formal definition of a D-dimensional manifold and particularizing for the case of our proposed embedding, we thus have

$$\begin{aligned}
\Gamma^k_{i,j}(\boldsymbol{x}) &= \sum_{l=1}^{D} G^{-1}_{k,l}(\boldsymbol{x}) \left\langle \frac{\partial^2}{\partial x_i \partial x_j} \Xi, \frac{\partial}{\partial x_l} \Xi \right\rangle = \alpha^2 \sum_{l=1}^{D} G^{-1}_{k,l}(\boldsymbol{x}) \ \frac{\partial}{\partial x_l} \ell(\boldsymbol{x}) \frac{\partial^2}{\partial x_i \partial x_j} \ell(\boldsymbol{x}) \\
&= \alpha^2 \sum_{l=1}^{D} \left(\delta_{k,l} - \alpha^2 \frac{\frac{\partial}{\partial x_k} \ell(\boldsymbol{x}) \frac{\partial}{\partial x_l} \ell(\boldsymbol{x})}{1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2} \right) \frac{\partial}{\partial x_l} \ell(\boldsymbol{x}) \frac{\partial^2}{\partial x_i \partial x_j} \ell(\boldsymbol{x}) \\
&= \alpha^2 \left(1 - \alpha^2 \frac{\left\|\nabla \ell(\boldsymbol{x})\right\|^2}{1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2} \right) \frac{\partial}{\partial x_k} \ell(\boldsymbol{x}) \frac{\partial^2}{\partial x_i \partial x_j} \ell(\boldsymbol{x}) \\
&= \frac{\alpha^2}{1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2} \frac{\partial}{\partial x_k} \ell(\boldsymbol{x}) \frac{\partial^2}{\partial x_i \partial x_j} \ell(\boldsymbol{x}),
\end{aligned}$$

which corresponds to the expression provided in the main paper. Since the Christoffel symbols are symmetric over the indices i, j, we can further express them in full matrices as

$$\Gamma^k = \frac{\alpha^2}{1 + \alpha^2 \left\|\nabla \ell(\boldsymbol{x})\right\|^2} \frac{\partial}{\partial x_k} \ell(\boldsymbol{x}) \ H(\boldsymbol{x})$$

for $k = 1, \ldots, D$, where $H(\boldsymbol{x}) = \nabla \nabla \ell(\boldsymbol{x})$ is the Hessian matrix of the log target distribution.

Matrix $\Omega(\boldsymbol{x}, \boldsymbol{v})$ The LMC updates (Eq. (3) in main paper) depend on the matrix $\Omega(\boldsymbol{x}, \boldsymbol{v})$, which is a matrix whose (i, j) element is given by $\sum_k v_k \Gamma^i_{k,j}(\boldsymbol{x})$. In full matrix form this simplifies to

$$\begin{aligned}
\Omega &= \begin{bmatrix} \boldsymbol{v}^\top \Gamma(\boldsymbol{x})^1_{\bullet,1} & \cdots & \boldsymbol{v}^\top \Gamma(\boldsymbol{x})^1_{\bullet,D} \\ \vdots & \ddots & \vdots \\ \boldsymbol{v}^\top \Gamma(\boldsymbol{x})^D_{\bullet,1} & \cdots & \boldsymbol{v}^\top \Gamma(\boldsymbol{x})^D_{\bullet,D} \end{bmatrix} = (I_D \otimes \boldsymbol{v}^\top) \begin{bmatrix} \Gamma^1(\boldsymbol{x}) \\ \vdots \\ \Gamma^D(\boldsymbol{x}) \end{bmatrix} = (I_D \otimes \boldsymbol{v}^\top) \frac{\alpha^2}{L_\alpha(\boldsymbol{x})} \begin{bmatrix} \frac{\partial}{\partial x_1} \ell(\boldsymbol{x}) H(\boldsymbol{x}) \\ \vdots \\ \frac{\partial}{\partial x_D} \ell(\boldsymbol{x}) H(\boldsymbol{x}) \end{bmatrix} \\
&= \frac{\alpha^2}{L_\alpha(\boldsymbol{x})} (I_n \otimes \boldsymbol{v}^\top)(\nabla \ell \otimes H(\boldsymbol{x})) = \frac{\alpha^2}{L_\alpha(\boldsymbol{x})} \nabla \ell(\boldsymbol{x}) \otimes (\boldsymbol{v}^\top H(\boldsymbol{x})) = \frac{\alpha^2}{L_\alpha(\boldsymbol{x})} \nabla \ell(\boldsymbol{x}) \ (\boldsymbol{v}^\top H(\boldsymbol{x})).
\end{aligned}$$

Building on this, the matrix $\tilde{\Omega}(\boldsymbol{x}, \boldsymbol{v}) = G(\boldsymbol{x})\ \Omega(\boldsymbol{x}, \boldsymbol{v})$ required for the determinant adjustment and the inverses in the numerical integrator updates reduces to

$$\begin{aligned}\tilde{\Omega}(\boldsymbol{x}, \boldsymbol{v}) &= \big(I_D + \alpha^2 \nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top\big)\tfrac{\alpha^2}{L_\alpha(\boldsymbol{x})}\nabla\ell(\boldsymbol{x})\ (\boldsymbol{v}^\top H(\boldsymbol{x})) \\ &= \alpha^2 \left(\frac{L_\alpha(\boldsymbol{x}) - 1}{L_\alpha(\boldsymbol{x})}\nabla\ell(\boldsymbol{x})\ (\boldsymbol{v}^\top H(\boldsymbol{x})) + \frac{1}{L_\alpha(\boldsymbol{x})}\nabla\ell(\boldsymbol{x})\ (\boldsymbol{v}^\top H(\boldsymbol{x}))\right) \\ &= \alpha^2\ \nabla\ell(\boldsymbol{x})\ (\boldsymbol{v}^\top H(\boldsymbol{x})).\end{aligned}$$

Both of these will be required for simplifying the update rules in the Monge metric.

Determinant For the determinant $\det\big(G((\boldsymbol{x}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}, \boldsymbol{v})\big)$, that is necessary in the Metropolis-Hasting acceptance probability rule, we use the Sherman-Morrison matrix lemma to get

$$\begin{aligned}\det G((\boldsymbol{x}) \pm \tfrac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}, \boldsymbol{v}) &= \det G(\boldsymbol{x}) \det I_D \pm \tfrac{\varepsilon}{2}\Omega(\boldsymbol{x}, \boldsymbol{v}) \\ &= L_\alpha(\boldsymbol{x}) \det I_D \pm \tfrac{\varepsilon}{2}\tfrac{\alpha^2}{L_\alpha(\boldsymbol{x})}\nabla\ell(\boldsymbol{x})\, \boldsymbol{v}^\top H \\ &= L_\alpha(\boldsymbol{x})\big(1 \pm \tfrac{\varepsilon}{2}\tfrac{\alpha^2}{L_\alpha(\boldsymbol{x})}\langle\nabla\ell, H(\boldsymbol{x})\, \boldsymbol{v}\rangle\big) \\ &= 1 + \alpha^2 \left\|\nabla\ell(\boldsymbol{x})\right\|^2 \pm \tfrac{\varepsilon}{2}\alpha^2\langle\nabla\ell(\boldsymbol{x}), H(\boldsymbol{x})\, \boldsymbol{v}\rangle. \end{aligned} \tag{5}$$

Inverses The inverse matrices required in the first and third updating equation of the velocity vector of the explicitly numerical integrator are also simplified using the same matrix lemma. We have

$$\begin{aligned}\Big(G(\boldsymbol{x}) + \tfrac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}, \boldsymbol{v})\Big)^{-1} &= \left(I_D + \alpha^2\ \nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top + \frac{\varepsilon\alpha^2}{2}\nabla\ell(\boldsymbol{x})\ (\boldsymbol{v}^\top H(\boldsymbol{x}))\right)^{-1} \\ &= \Big(I_D + \alpha^2\nabla\ell(\boldsymbol{x})\big(\nabla\ell(\boldsymbol{x})^\top + \tfrac{\varepsilon}{2}\, \boldsymbol{v}^\top H(\boldsymbol{x})\big)\Big)^{-1} \\ &= I_D - \frac{\nabla\ell(\boldsymbol{x})\ \big(\nabla\ell(\boldsymbol{x})^\top + \frac{\varepsilon}{2}\, \boldsymbol{v}^\top H(\boldsymbol{x})\big)}{\big(\nabla\ell(\boldsymbol{x})^\top + \frac{\varepsilon}{2}\, \boldsymbol{v}^\top H(\boldsymbol{x})\big)\nabla\ell(\boldsymbol{x}) + \frac{1}{\alpha^2}}. \end{aligned} \tag{6}$$

Gradient of potential energy The potential energy for LMC is $\phi(\boldsymbol{x}) = -\log\pi_{\boldsymbol{X}}(\boldsymbol{x}) + \frac{1}{2}\log\det G(\boldsymbol{x})$ and we need the gradient of that. The first term is obvious and the latter can be computed using

$$\begin{aligned}\frac{\partial}{\partial x_i}\log\det G(\boldsymbol{x}) &= tr\left(G^{-1}(\boldsymbol{x})\frac{\partial}{\partial x_i}G(\boldsymbol{x})\right) \\ &= \alpha^2\ tr\left[\left(I_D - \alpha^2\frac{\nabla\ell(\boldsymbol{x})\ \nabla\ell(\boldsymbol{x})^\top}{L_\alpha(\boldsymbol{x})}\right) 2\left(\frac{\partial}{\partial x_i}\nabla\ell(\boldsymbol{x})\right)\nabla\ell(\boldsymbol{x})^\top\right] \\ &= \frac{2\alpha^2}{L_\alpha(\boldsymbol{x})}\ \left[tr(H_i(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top L_\alpha(\boldsymbol{x})) - \alpha^2 tr(\nabla\ell(\boldsymbol{x})\ \nabla\ell(\boldsymbol{x})^\top H_i(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top)\right] \\ &= 2\alpha^2\nabla\ell(\boldsymbol{x})^\top H_i(\boldsymbol{x})\Big(1 - \alpha^2\frac{\nabla\ell(\boldsymbol{x})^\top\nabla\ell(\boldsymbol{x})}{L_\alpha(\boldsymbol{x})}\Big) = 2\alpha^2\Big(1 - \alpha^2\frac{\nabla\ell(\boldsymbol{x})^\top\nabla\ell(\boldsymbol{x})}{L_\alpha(\boldsymbol{x})}\Big)\nabla\ell(\boldsymbol{x})^\top H_i(\boldsymbol{x}) \\ &= 2\alpha^2\frac{\nabla\ell(\boldsymbol{x})^\top}{L_\alpha(\boldsymbol{x})}H_i(\boldsymbol{x}),\end{aligned}$$

where $H_i(\boldsymbol{x})$ is the i^{th} row (or column) of the Hessian matrix. Hence the gradient of the second term in the potential energy is given by

$$\nabla\log\det G = \frac{2\alpha^2}{L_\alpha(\boldsymbol{x})}H(\boldsymbol{x})\ \nabla\ell(\boldsymbol{x}). \tag{7}$$

Relationship to Fisher metric Section 3.1 established a relationship between Fisher and Monge metrics. For the specific case where $\ell(\boldsymbol{x}) = \log\pi_Y(Y|\,\boldsymbol{x})$ corresponds to a data generating distribution over some random variable Y, we can compute the expectation of the Monge metric over Y. We can then write

$$\mathbb{E}_Y\left(G_M(\boldsymbol{x})\right) = I_D + \alpha^2\,\mathbb{E}_Y\left(\nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top\right),$$

where the latter term corresponds to the definition of Fisher metric $G_F(\boldsymbol{x})$ as expressed in Section 2. Consequently, we can interpret the expected Monge metric as a biased (and scaled) estimator for the Fisher metric, for instance writing

$$\mathbb{E}_Y\left(G_M(\boldsymbol{x})\right) = I_D + \alpha^2 G_F(\boldsymbol{x})$$

or equivalently

$$G_F(\boldsymbol{x}) = \alpha^{-2}\left[\mathbb{E}_Y\left(G_M(\boldsymbol{x})\right) - I_D\right].$$

C CLOSED-FORM EXPLICIT NUMERICAL INTEGRATOR

Having established the required computational elements in the previous section, we proceed to derivation of the update rules for the explicit numerical integrator for LMC-Monge. Full pseudo-code for the resulting algorithm is given in Algorithm 1 and reference implementation in `Julia` is provided at `https://github.com/mahaa2/EmbeddedLMC`. The code also includes scripts for re-creating some of the experiments.

The LMC algorithm assumes that step-size ε, number of steps L_F and the metric control parameter α are provided as inputs, together with some initial value for $\boldsymbol{x}^{(1)}$ (the previous sample). The integrator proposed by Lan et al. (2015) then repeats the following steps L_F times:

$$\begin{aligned}
\boldsymbol{v}^{(n+1/2)} &= \Big[G(\boldsymbol{x}^{(n)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n)}, \boldsymbol{v}^{(n)})\Big]^{-1}\Big[G(\boldsymbol{x}^{(n)})\,\boldsymbol{v}^{(n)} - \frac{\varepsilon}{2}\nabla\phi(\boldsymbol{x}^{(n)})\Big] \\
\boldsymbol{x}^{(n+1)} &= \boldsymbol{x}^{(n)} + \varepsilon\,\boldsymbol{v}^{(n+1/2)} \\
\boldsymbol{v}^{(n+1)} &= \Big[G(\boldsymbol{x}^{(n+1)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n+1)}, \boldsymbol{v}^{(n+1/2)})\Big]^{-1}\Big[G(\boldsymbol{x}^{(n+1)})^{-1}\,\boldsymbol{v}^{(n+1/2)} - \frac{\varepsilon}{2}\nabla\phi(\boldsymbol{x}^{(n+1)})\Big].
\end{aligned} \tag{8}$$

To express these updates in the Monge metric, we will use the simplifications described in the previous section. The update for the position $\boldsymbol{x}$ does not depend on the metric, whereas the two updates for the velocity $\boldsymbol{v}$ are analogous, requiring the same algebraic changes. Consequently, we only write the first update explicitly, using (7) to compute the gradient of the energy and (6) to compute the inverse. This results the expression provided also in Table 1 of the main paper:

$$\begin{aligned}
\boldsymbol{v}^{(n+1/2)} &= \Big[G(\boldsymbol{x}^{(n)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n)}, \boldsymbol{v}^{(n)})\Big]^{-1}\Big[G(\boldsymbol{x}^{(n)})\,\boldsymbol{v}^{(n)} - \frac{\varepsilon}{2}\nabla\phi(\boldsymbol{x}^{(n)})\Big] \\
&= \left[I_D - \frac{\nabla\ell(\boldsymbol{x}^{(n)})\big(\nabla\ell(\boldsymbol{x}^{(n)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n)})^\top H(\boldsymbol{x}^{(n)})\big)}{\big[\nabla\ell(\boldsymbol{x}^{(n)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n)})^\top H(\boldsymbol{x}^{(n)})\big]^\top \nabla\ell(\boldsymbol{x}^{(n)}) + \frac{1}{\alpha}}\right] \\
&\qquad \times \left\{\left[\big(\alpha^2\nabla\ell(\boldsymbol{x}^{(n)})^\top\,\boldsymbol{v}^{(n)} + \frac{\varepsilon}{2}\big)I_D - \frac{\alpha^2\varepsilon}{2L_\alpha}H(\boldsymbol{x}^{(n)})\right]\nabla\ell(\boldsymbol{x}^{(n)}) + \boldsymbol{v}^{(n)}\right\}.
\end{aligned}$$

The integrator is not volume-preserving (see Lan et al., 2015). Thus the proposal' acceptance probability needs the determinant adjustment and becomes

$$\alpha_{LMC} = \min\Big\{1, \exp\big(-E(\boldsymbol{x}^{(L_F+1)}, \boldsymbol{v}^{(L_F+1)}) + E(\boldsymbol{x}^{(1)}, \boldsymbol{v}^{(1)})\big)\,|\det J|\Big\}$$

where the energy function E is defined as,

$$\begin{aligned}
E(\boldsymbol{x}, \boldsymbol{v}) &= -\ell(\boldsymbol{x}) - \tfrac{1}{2}\log\det G(\boldsymbol{x}) + \tfrac{1}{2}\,\boldsymbol{v}^\top G(\boldsymbol{x})\,\boldsymbol{v} \\
&= -\ell(\boldsymbol{x}) - \tfrac{1}{2}\log(1 + \alpha^2\,\|\nabla\ell(\boldsymbol{x})\|^2) + \tfrac{1}{2}\,\|\boldsymbol{v}\|^2 + \frac{\alpha^2}{2}\,\langle\nabla\ell(\boldsymbol{x}), \boldsymbol{v}\rangle^2
\end{aligned}$$

and the determinant adjustment becomes

$$\begin{aligned}
\det J &= \prod_{n=1}^{L_F} \frac{\det\big(G(\boldsymbol{x}^{(n+1)}) - \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n+1)}, \boldsymbol{v}^{(n+1)})\big)}{\det\big(G(\boldsymbol{x}^{(n+1)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n+1)}, \boldsymbol{v}^{(n+1/2)})\big)} \frac{\det\big(G(\boldsymbol{x}^{(n)}) - \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n)}, \boldsymbol{v}^{(n+1/2)})\big)}{\det\big(G(\boldsymbol{x}^{(n)}) + \frac{\varepsilon}{2}\tilde{\Omega}(\boldsymbol{x}^{(n)}, \boldsymbol{v}^{(n)})\big)} \\
&= \prod_{n=1}^{L_F} \frac{L_\alpha(\boldsymbol{x}^{(n+1)}) - \frac{\alpha^2\varepsilon}{2}\langle\nabla\ell(\boldsymbol{x}^{(n+1)}), H(\boldsymbol{x}^{(n+1)})\,\boldsymbol{v}^{(n+1)}\rangle}{L_\alpha(\boldsymbol{x}^{(n+1)}) + \frac{\alpha^2\varepsilon}{2}\langle\nabla\ell(\boldsymbol{x}^{(n+1)}), H(\boldsymbol{x}^{(n+1)})\,\boldsymbol{v}^{(n+1/2)}\rangle} \frac{L_\alpha(\boldsymbol{x}^{(n)}) - \frac{\alpha^2\varepsilon}{2}\langle\nabla\ell(\boldsymbol{x}^{(n)}), H(\boldsymbol{x}^{(n)})\,\boldsymbol{v}^{(n+1/2)}\rangle}{L_\alpha(\boldsymbol{x}^{(n)}) + \frac{\alpha^2\varepsilon}{2}\langle\nabla\ell(\boldsymbol{x}^{(n)}), H(\boldsymbol{x}^{(n)})\,\boldsymbol{v}^{(n)}\rangle},
\end{aligned}$$

using the simplification provided in (5).

Algorithm 1: Explicit Lagrangian Monte Carlo via embedding using the Monge patch

Result: A sample from the distribution $\pi_{\boldsymbol{X}}(\cdot)$

Inputs : $\nabla\ell$, H, ε, L_F, α and $\boldsymbol{x}^{(1)}$;

Sample new velocity $\boldsymbol{v}^{(1)} = \sqrt{G^{-1}(\boldsymbol{x}^{(1)})}\boldsymbol{z}$ where $z \sim \mathcal{N}(0, I_D)$;

Calculate current $E_1 = E(\boldsymbol{x}^{(1)}, \boldsymbol{v}^{(1)})$;

$\Delta \log\det = 0$;

for $n = 1, \ldots, L_F$ **do**

$\Delta \log\det = \Delta \log\det - \log\left|(1 + \frac{\alpha^2\varepsilon}{2L_\alpha(\boldsymbol{x}^{(n)})}\nabla\ell(\boldsymbol{x}^{(n)})^\top H(\boldsymbol{x}^{(n)})\,\boldsymbol{v}^{(n)})\right|$;

\# update velocity explicitly with a half-step;

$$\boldsymbol{v}^{(n+1/2)} = \left[I_D - \frac{\nabla\ell(\boldsymbol{x}^{(n)})\big(\nabla\ell(\boldsymbol{x}^{(n)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n)})^\top H(\boldsymbol{x}^{(n)})\big)}{\left[\nabla\ell(\boldsymbol{x}^{(n)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n)})^\top H(\boldsymbol{x}^{(n)})\right]^\top \nabla\ell(\boldsymbol{x}^{(n)}) + \frac{1}{\alpha^2}}\right] \times \left\{\left[\big(\alpha^2\nabla\ell(\boldsymbol{x}^{(n)})^\top\,\boldsymbol{v}^{(n)} + \frac{\varepsilon}{2}\big)I_D - \frac{\alpha^2\varepsilon}{2L_\alpha(\boldsymbol{x}^{(n)})}H(\boldsymbol{x}^{(n)})\right]\nabla\ell(\boldsymbol{x}^{(n)}) + \boldsymbol{v}^{(n)}\right\}$$

$\Delta \log\det = \Delta \log\det + \log\left|(1 - \frac{\alpha^2\varepsilon}{2L_\alpha(\boldsymbol{x}^{(n)})}\nabla\ell(\boldsymbol{x}^{(n)})^\top H(\boldsymbol{x}^{(n)})\,\boldsymbol{v}^{(n+1/2)})\right|$;

\# update position with a full-step;

$$\boldsymbol{x}^{(n+1)} = \boldsymbol{x}^{(n)} + \varepsilon\,\boldsymbol{v}^{(n+1/2)}$$

$\Delta \log\det = \Delta \log\det - \log\left|(1 + \frac{\alpha^2\varepsilon}{2L_\alpha(\boldsymbol{x}^{(n+1)})}\nabla\ell(\boldsymbol{x}^{(n+1)})^\top H(\boldsymbol{x}^{(n+1)})\,\boldsymbol{v}^{(n+1/2)})\right|$;

\# update velocity explicitly with a half-step;

$$\boldsymbol{v}^{(n+1)} = \left[I_D - \frac{\nabla\ell(\boldsymbol{x}^{(n+1)})\big(\nabla\ell(\boldsymbol{x}^{(n+1)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n+1/2)})^\top H(\boldsymbol{x}^{(n+1)})\big)}{\left[\nabla\ell(\boldsymbol{x}^{(n+1)})^\top + \frac{\varepsilon}{2}(\boldsymbol{v}^{(n+1/2)})^\top H(\boldsymbol{x}^{(n+1)})\right]^\top \nabla\ell(\boldsymbol{x}^{(n+1)}) + \frac{1}{\alpha^2}}\right] \times \left\{\left[\big(\alpha^2\nabla\ell(\boldsymbol{x}^{(n+1)})^\top\,\boldsymbol{v}^{(n+1/2)} + \frac{\varepsilon}{2}\big)I_D - \frac{\alpha^2\varepsilon}{2L_\alpha(\boldsymbol{x}^{(n)})}H(\boldsymbol{x}^{(n+1)})\right]\nabla\ell(\boldsymbol{x}^{(n+1)}) + \boldsymbol{v}^{(n+1/2)}\right\}$$

$\Delta \log\det = \Delta \log\det + \log\left|(1 - \frac{\varepsilon\alpha^2}{2L_\alpha(\boldsymbol{x}^{(n+1)})}\nabla\ell(\boldsymbol{x}^{(n+1)})^\top H(\boldsymbol{x}^{(n+1)})\,\boldsymbol{v}^{(n+1)})\right|$;

end

Calculate proposed $E_{L_F} = E(\boldsymbol{x}^{(L_F+1)}, \boldsymbol{v}^{(L_F+1)})$;

Calculate *logRatio* $= -E_1 + E_{L_F} + \Delta \log\det$;

Sample $u \sim U(0, 1)$;

if *logRatio* $> u$ **then**

Accept $(\boldsymbol{x}^{(L_F+1)}, \boldsymbol{v}^{(L_F+1)})$ as the current sample

else

Reject $(\boldsymbol{x}^{(L_F+1)}, \boldsymbol{v}^{(L_F+1)})$ and keep $(\boldsymbol{x}^{(1)}, \boldsymbol{v}^{(1)})$ as the current sample

end

D METRIC-TENSOR SQUARE ROOT AND VELOCITY SAMPLING

To sample from the multivariate Gaussian $\boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, G(\boldsymbol{x})^{-1})$, we need the square root matrix $\sqrt{G^{-1}(\boldsymbol{x})} = A$ such that $G^{-1}(\boldsymbol{x}) = AA^\top$. Since the inverse matrix of the metric is also formed by the outer-product of the gradients, it is possible to find the square root matrix with cost $\mathcal{O}(D^2)$, instead of the standard Cholesky decomposition with computational cost of $\mathcal{O}(D^3)$.

For inverse matrix of the metric-tensor $G_M(\boldsymbol{x})$, let the square root matrix be of the form $A = I_D + tuu^\top$, where $t \in \mathbb{R}$. Then we have

$$AA^\top = I_D + (2t + t^2 \left\|u\right\|^2)uu^\top$$

and we want that

$$G_M(\boldsymbol{x})^{-1} = I_D - \alpha^2 \frac{\nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top}{1 + \alpha^2 \left\|\nabla\ell(\boldsymbol{x})\right\|^2} = I_D + (2t + t^2 \left\|\ell(\boldsymbol{x})\right\|^2)\nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top$$

which is equivalent as finding the roots of the quadratic equation in t

$$\left\|\nabla\ell(\boldsymbol{x})\right\|^2 t^2 + 2t + \frac{\alpha^2}{1 + \alpha^2 \left\|\nabla\ell(\boldsymbol{x})\right\|^2} = 0,$$

whose positive root is given by

$$t_+ = -\frac{1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2} + \frac{1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2 \sqrt{1 + \alpha^2 \left\|\nabla\ell(\boldsymbol{x})\right\|^2}}.$$

Setting $t = t_+$, $u = \nabla\ell(\boldsymbol{x})$ in A and rearranging, we get

$$\sqrt{G_M(\boldsymbol{x})^{-1}} = I_D + \frac{1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2}\left(\frac{1}{L_\alpha(\boldsymbol{x})^{\frac{1}{2}}} - 1\right)\nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top.$$

Around the local modes of the log target distribution, the metric-tensor reduces to the Euclidean metric. In these cases, the scalar value

$$c(\left\|\nabla\ell(x)\right\|^2) = \frac{1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2}\left(\frac{1}{L_\alpha(\boldsymbol{x})^{\frac{1}{2}}} - 1\right)$$

might cause instability in computer implementations since the norm of the gradient will be zero. To address this computational issue, we obtain the limit of $c(\cdot)$ when the gradient approaches the zero vector. That is,

$$\begin{aligned}
\lim_{\left\|\nabla\ell(x)\right\|^2 \to 0} \frac{1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2}\left(\frac{1}{(1 + \alpha^2 \left\|\nabla\ell(x)\right\|^2)^{\frac{1}{2}}} - 1\right) &= \lim_{\left\|\nabla\ell(x)\right\|^2 \to 0} \frac{\frac{1}{(1+\alpha^2\left\|\nabla\ell(x)\right\|^2)^{\frac{1}{2}}} - 1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2} \frac{\frac{1}{(1+\alpha^2\left\|\nabla\ell(x)\right\|^2)^{\frac{1}{2}}} + 1}{\frac{1}{(1+\alpha^2\left\|\nabla\ell(x)\right\|^2)^{\frac{1}{2}}} + 1} \\
&= \lim_{\left\|\nabla\ell(x)\right\|^2 \to 0} \frac{\frac{1}{1+\alpha^2\left\|\nabla\ell(x)\right\|^2} - 1}{\left\|\nabla\ell(\boldsymbol{x})\right\|^2 \left(\frac{1}{(1+\alpha^2\left\|\nabla\ell(x)\right\|^2)^{\frac{1}{2}}} + 1\right)} \\
&= \lim_{\left\|\nabla\ell(x)\right\|^2 \to 0} -\frac{\alpha^2}{(1 + \alpha^2 \left\|\nabla\ell(x)\right\|^2)\left(\frac{1}{(1+\alpha^2\left\|\nabla\ell(x)\right\|^2)^{\frac{1}{2}}} + 1\right)} \\
&= -\frac{\alpha^2}{2},
\end{aligned}$$

so that for $\left\|\nabla\ell(\boldsymbol{x})\right\|^2 \approx 0$ we approximate the metric-tensor square root as

$$\sqrt{G_M(\boldsymbol{x})^{-1}} \approx I_D - \frac{\alpha^2}{2}\nabla\ell(\boldsymbol{x})\nabla\ell(\boldsymbol{x})^\top.$$

Table 3: Parameter settings for the funnel experiments.

D	1	3	5	10	30	40	50
ε	0.2	0.2	0.09	0.04	0.025	0.02	0.017
L_F	9	9	25	100	150	180	250

E EXPERIMENT DETAILS

In this section we provide all computational details for the empirical experiments and demonstrations shown in the main paper, with some additional visualizations.

E.1 Ring Probability Distribution (Figure 3)

Figure 3 plotted geodesic curves of LMC-Monge for different α. The ring distribution used here was defined as follows. Let the random variables $R \sim \mathcal{N}(\mu, \sigma^2)$ and $\Theta \sim U[0, 2\pi]$. We now define new random variables

$$X = R\cos(\Theta) \iff R = \sqrt{X^2 + Y^2}$$
$$Y = R\sin(\Theta) \qquad \Theta = \arctan(Y/X)$$

Since the above transformation is one-to-one and smooth, by the Jacobian method of transformation of random variables we get the distribution

$$\pi_{X,Y}(x, y) = \mathcal{N}\big(\sqrt{x^2 + y^2}|\mu, \sigma^2\big)/(2\pi\ \sqrt{x^2 + y^2}),$$

where the Jacobian $\big|\partial(r, \theta)/\partial(x, y)\big| = 1/\sqrt{x^2 + y^2}$. The parameter μ controls the radius of the ring measured from the origin and the parameters σ^2 the thickness of the ring.

We used $\mu = 12$ and $\sigma^2 = 0.12$ for Figure 3. The geodesic trajectories were integrated using $\varepsilon = 0.03$ and $L_F = 200$ for one sample path, in order to guarantee smooth paths with minimal integration error.

E.2 Funnel (Section 5.1)

For the funnel probability distribution in Section 5.1 we used $\alpha = 1$ for all dimensionalities D and the parameters ε and L_F are provided in Table 3. These were set based on manual inspection following the basic principle of using smaller ϵ and larger L_F for the more complex cases, but the results are not sensitive to the exact choices. For HMC-Nuts we used the classical method from Hoffman and Gelman (2014) as implemented in `Turing.jl` using the command `NUTS{SliceTS, ClassicNoUTurn}(LeapFrog(stepsize))`, but note that other variants of the NUTS algorithm behaved in a very similar manner. Using `MassMatrixAdaptor()` to fine-tune the Euclidean metric tensor M made convergence faster, but did not help improving the exploration. We set the initial step size using `find_good_stepsize()`.

The initial values for all cases were given by $\boldsymbol{x}^{(1)} = 5\mathbb{1}_D$, where $\mathbb{1}_D$ is the vector composed by D unitary elements. Figure 5 in the main paper showed the KL divergence as function of D and illustrated two margins for $D \in \{3, 5, 10, 30, 40, 50\}$. For completeness, we plot the marginals for all D in Figure 7 to show that the difference between LMC-Monge and NUTS is consistent.

E.3 Logistic Regression (Section 5.2)

All the binary classification tasks in the main text use the logit link function to model the probability parameter. The un-normalised posterior distribution is given by

$$\pi(\boldsymbol{\theta} \,|\, \boldsymbol{y}) \propto \prod_{i=1}^{n} \left(\frac{\exp(\mathbf{x}_i^\top \boldsymbol{\theta})}{1 + \exp(\mathbf{x}_i^\top \boldsymbol{\theta})}\right)^{y_i} \left(\frac{1}{1 + \exp(\mathbf{x}_i^\top \boldsymbol{\theta})}\right)^{n - y_i} \mathcal{N}(\boldsymbol{\theta} \,|\, \mathbf{0}, 100 \times I_D).$$

where each $y_i \in \{0, 1\}$ and $\mathbf{x}_i$ is a vector of covariates (or inputs). The number n in the sample-size and D is the number of parameters in the model. This formulation, including the prior choice, matches the one used by Lan et al. (2015).

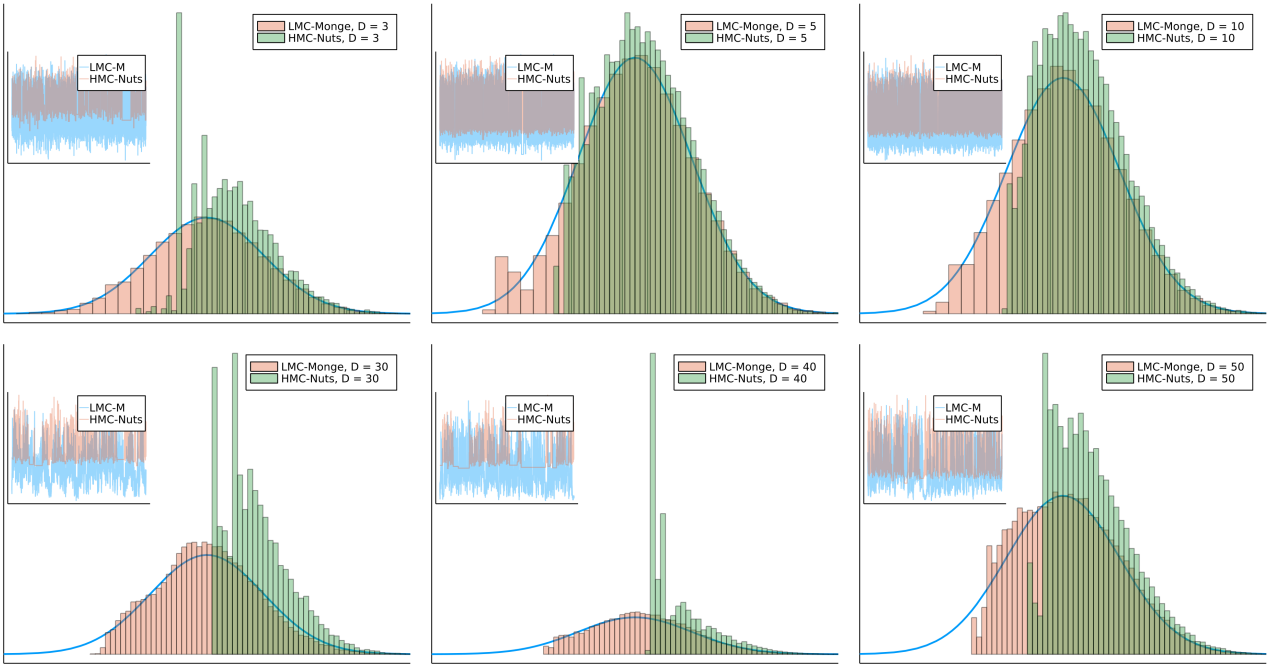

Figure 7: Histograms and MCMC chains for the LMC with Monge metrics and HMC-nuts. The geometric MCMC tends to be better in reaching corners with high curvature compared to HMC-Nuts, leading to a better estimate of the marginal distribution of the parameter a. The blue lines depict the true marginal distribution.

For a sample from a Markov chain $(X_i)_{i=1,\ldots,N}$. The effective sample size (ESS) was computed as

$$ESS = \frac{N}{1 + 2\sum_{t=1}^{N-2} \hat{\rho}_t}$$

where

$$\hat{\rho}_t = \frac{1}{N-t} \sum_{r=1}^{N-t} (X_r - \bar{X})(X_{r+t} - \bar{X})$$

using the implementation from the package `MCMCDiagnostics.jl`.

For the LMC-Monge we chose ε and L_F by trial and error following the same principle that Lan et al. (2015) used for the original LMC-Fisher, aiming for acceptance probability between 0.6 and 0.9. For LMC-Fisher, RMHMC-Fisher and HMC-Euclidean, we used the values provided in the Matlab implementations in `https://bitbucket.org/geomstatcomp/lagrangian-monte-carlo/src/master/`, satisfying the same acceptance probability thresholds. For HMC-Nuts we again used the `find_good_stepsize()` function to set ε, but it failed to converge due to too large-step size. We fixed this by trial and error, ending up using a smaller ε. Note that HMC-Nuts automatically adapts the step length during the algorithm and hence the acceptance probability differs from the aimed range, and NUTS has no parameter L_F as the integration length is determined by the algorithm. Table 4 lists the final values used for all methods for the experiment reported in Table 2. For the experiment reported in Figure 6, we used $L_F = 7$ and $\varepsilon = 0.09$ for all MCMC runs and data-sets.

E.4 Squiggle Probability Distribution (Additional experiment)

Here we define the squiggle probability distribution and provide extra empirical evidence for quality of the proposed algorithm based on the embedding. Let the vector of random variables $Y = (Y_1, Y_2) \sim \mathcal{N}(\mathbf{0}, \Sigma)$. Define the new vector $X = (X_1, X_2) = (Y_1, Y_2 - \sin(aY_1))$. For Jacobian method of transformation of random variables

Table 4: Parameter settings for the logistic regression experiment.

Data	Method	ε	L_F
Heart	LMC-Monge ($\alpha = 0.01$)	0.085	7
	LMC-Fisher	0.75	5
	RMHMC-Fisher	0.75	5
	HMC-Euclidean	0.18	25
	HMC-Nuts	0.066	NA
German	LMC-Monge ($\alpha = 0.01$)	0.05	5
	LMC-Fisher	0.8	5
	RMHMC-Fisher	0.67	6
	HMC-Euclidean	0.063	64
	HMC-Nuts	0.066	NA
Australian	LMC-Monge ($\alpha = 0.01$)	0.085	6
	LMC-Fisher	0.75	6
	RMHMC-Fisher	0.75	6
	HMC-Euclidean	0.11	40
	HMC-Nuts	0.066	NA

we need the inverse mapping given by $(Y_1, Y_2) = (X_1, X_2 + \sin(aX_1))$. Hence the joint distribution in X reads,

$$\pi_X(x_1, x_2|a, \Sigma) = \mathcal{N}(\boldsymbol{y}(x_1, x_2)|\,\mathbf{0}, \Sigma)|\det J_{\boldsymbol{x}\to\boldsymbol{y}}| = \mathcal{N}(\boldsymbol{y}(x_1, x_2)|\,\mathbf{0}, \Sigma) \tag{9}$$

since $|\det J_{\boldsymbol{x}\to\boldsymbol{y}}| = 1$. The parameter $a \geq 0$. In the experiment we vary $a \in \{0.5, 1.0, 2.0\}$, $\Sigma = [10\ 0.01;\ 0.01\ 0.001]$ with initial point $\boldsymbol{x}_0 = [1, -1]^\top$. The chain size is $N = 60000$. For the LMC-monge the step-size ε was $0.07, 0.07, 0.025$, the leapfrog steps L_F were $13, 13, 35$ and $\alpha = 1.0$. Those were chosen again by the inspection of the MCMC chain's convergence. For the HMC-Nuts, we set it similarly as before and we also used `find_good_stepsize` function to set the initial ε at $\boldsymbol{x}_0$. See Figure 8 for the visualisation of the results.

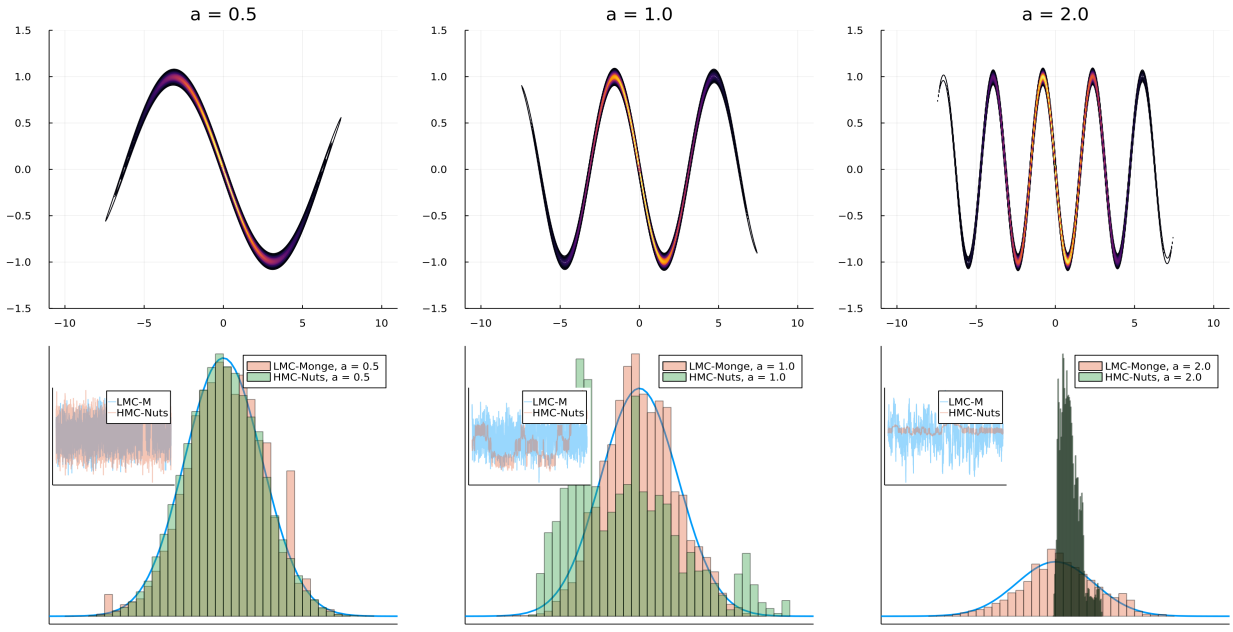

Figure 8: The first row depicts the forms of (9) with varying a. The larger the value of a is, the longer is its sinusoidal form. In the second row, histograms and MCMC chains for LMC and NUTS are shown. LMC algorithm tends to be better in exploring the typical sets when compared to HMC-Nuts, leading to a better estimate of the marginal distribution $\pi_X(x_1)$, in blue.