# Wormhole Hamiltonian Monte Carlo

**Shiwei Lan**
Department of Statistics
University of California, Irvine

**Jeffrey Streets**
Department of Mathematics
University of California, Irvine

**Babak Shahbaba**
Department of Statistics
University of California, Irvine

## Abstract

In machine learning and statistics, probabilistic inference involving multimodal distributions is quite difficult. This is especially true in high dimensional problems, where most existing algorithms cannot easily move from one mode to another. To address this issue, we propose a novel Bayesian inference approach based on Markov Chain Monte Carlo. Our method can effectively sample from multimodal distributions, especially when the dimension is high and the modes are isolated. To this end, it exploits and modifies the Riemannian geometric properties of the target distribution to create *wormholes* connecting modes in order to facilitate moving between them. Further, our proposed method uses the regeneration technique in order to adapt the algorithm by identifying new modes and updating the network of wormholes without affecting the stationary distribution. To find new modes, as opposed to rediscovering those previously identified, we employ a novel mode searching algorithm that explores a *residual energy* function obtained by subtracting an approximate Gaussian mixture density (based on previously discovered modes) from the target density function.

## Introduction

In Bayesian inference, it is well known that standard Markov Chain Monte Carlo (MCMC) methods tend to fail when the target distribution is multimodal (Neal 1993; 1996; Celeux, Hurn, and Robert 2000; Neal 2001; Rudoy and Wolfe 2006; Sminchisescu and Welling 2011; Craiu, R., and Y. 2009). These methods typically fail to move from one mode to another since such moves require passing through low probability regions. This is especially true for high dimensional problems with isolated modes. Therefore, despite recent advances in computational Bayesian methods, designing effective MCMC samplers for multimodal distribution has remained a major challenge. In the statistics and machine learning literature, many methods have been proposed address this issue (Neal 1996; 2001; Warnes 2001; Laskey and Myers 2003; Hinton, Welling, and Mnih 2004; Braak 2006; Rudoy and Wolfe 2006; Sminchisescu and Welling 2011; Ahn, Chen, and Welling 2013). However, these methods tend to suffer from the curse of dimensionality (Hinton, Welling, and Mnih 2004; Ahn, Chen, and Welling 2013).

In this paper, we propose a new algorithm, which exploits and modifies the Riemannian geometric properties of the target distribution to create wormholes connecting modes in order to facilitate moving between them. Our method can be regarded as an extension of Hamiltonian Monte Carlo (HMC). Compared to random walk Metropolis, standard HMC explores the target distribution more efficiently by exploiting its geometric properties. However, it too tends to fail when the target distribution is multimodal since the modes are separated by high energy barriers (low probability regions) (Sminchisescu and Welling 2011).

In what follows, we provide an brief overview of HMC. Then, we introduce our method assuming that the locations of the modes are known (either exactly or approximately), possibly through some optimization techniques (e.g., (Kirkpatrick, Gelatt, and Vecchi 1983; Sminchisescu and Triggs 2002)). Next, we relax this assumption by incorporating a mode searching algorithm in our method in order to identify new modes and to update the network of wormholes.

## Preliminaries

Hamiltonian Monte Carlo (HMC) (Duane et al. 1987; Neal 2010) is a Metropolis algorithm with proposals guided by Hamiltonian dynamics. HMC improves upon random walk Metropolis by proposing states that are distant from the current state, but nevertheless have a high probability of acceptance. These distant proposals are found by numerically simulating Hamiltonian dynamics, whose state space consists of its *position*, denoted by the vector $\boldsymbol{\theta}$, and its *momentum*, denoted by a vector $\boldsymbol{p}$. Our objective is to sample from the distribution of $\boldsymbol{\theta}$ with the probability density function (up to some constant) $\pi(\boldsymbol{\theta})$. We usually assume that the auxiliary momentum variable $\boldsymbol{p}$ has a multivariate normal distribution (the same dimension as $\boldsymbol{\theta}$) with mean zero. The covariance of $\boldsymbol{p}$ is usually referred to as the *mass matrix*, $\boldsymbol{M}$, which in standard HMC is usually set to the identity matrix, $\boldsymbol{I}$, for convenience.

Based on $\boldsymbol{\theta}$ and $\boldsymbol{p}$, we define the *potential energy*, $U(\boldsymbol{\theta})$, and the *kinetic energy*, $K(\boldsymbol{p})$. We set $U(\boldsymbol{\theta})$ to minus the log probability density of $\boldsymbol{\theta}$ (plus any constant). For the auxiliary momentum variable $\boldsymbol{p}$, we set $K(\boldsymbol{p})$ to be minus the log probability density of $\boldsymbol{p}$ (plus any constant). The *Hamilto-*

*nian* function is then defined as follows:

$$H(\boldsymbol{\theta}, \boldsymbol{p}) \quad = \quad U(\boldsymbol{\theta}) + K(\boldsymbol{p})$$

The partial derivatives of $H(\boldsymbol{\theta}, \boldsymbol{p})$ determine how $\boldsymbol{\theta}$ and $\boldsymbol{p}$ change over time, according to *Hamilton's equations*,

$$\begin{aligned} \dot{\boldsymbol{\theta}} &= \nabla_{\boldsymbol{p}} H(\boldsymbol{\theta}, \boldsymbol{p}) &=& \quad \boldsymbol{M}^{-1}\boldsymbol{p} \\ \dot{\boldsymbol{p}} &= -\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}, \boldsymbol{p}) &=& -\nabla_{\boldsymbol{\theta}} U(\boldsymbol{\theta}) \end{aligned} \tag{1}$$

Note that $\boldsymbol{M}^{-1}\boldsymbol{p}$ can be interpreted as velocity.

In practice, solving Hamiltonian's equations exactly is difficult, so we need to approximate these equations by discretizing time, using some small step size $e$. For this purpose, the *leapfrog* method is commonly used. We can use some number, $L$, of these leapfrog steps, with some stepsize, $e$, to propose a new state in the Metropolis algorithm. This proposal is either accepted or rejected based on the Metropolis acceptance probability.

While HMC explores the target distribution more efficiently than random walk Metropolis, it does not fully exploits its geometric properties. Recently, (Girolami and Calderhead 2011) proposed a new method, called Riemannian Manifold HMC (RMHMC), that improvs the efficiency of standard HMC by automatically adapting to the local structure. To this end, they follow (Amari and Nagaoka 2000) and propose Hamiltonian Monte Carlo methods defined on the Riemannian manifold endowed with metric $\boldsymbol{G}_0(\boldsymbol{\theta})$, which is set to the Fisher information matrix. More specifically, they define Hamiltonian dynamics in terms of a position-specific mass matrix, $\boldsymbol{M}$, set to $\boldsymbol{G}_0(\boldsymbol{\theta})$. The standard HMC method is a special case of RMHMC with $\boldsymbol{G}_0(\boldsymbol{\theta}) = \boldsymbol{I}$. Here, we use the notation $\boldsymbol{G}_0$ to generally refer to a Riemannian metric, which is not necessarily the Fisher information. In the following section, we introduce a natural modification of $\boldsymbol{G}_0$ such that the associated Hamiltonian dynamical system has a much greater chance of moving between isolated modes.

## Wormhole Hamiltonian Monte Carlo

Consider a manifold $\mathcal{M}$ endowed with a generic metric $\boldsymbol{G}_0(\boldsymbol{\theta})$. Given a differentiable curve $\boldsymbol{\theta}(t) : [0, T] \to \mathcal{M}$ one can define the arclength along this curve as

$$\ell(\boldsymbol{\theta}) := \int_0^T \sqrt{\dot{\boldsymbol{\theta}}(t)^\mathsf{T} \boldsymbol{G}_0(\boldsymbol{\theta}(t)) \dot{\boldsymbol{\theta}}(t)} dt \tag{2}$$

Under very general geometric assumptions, which are nearly always satisfied in statistical models, given any two points $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \mathcal{M}$ there exists a curve $\boldsymbol{\theta}(t) : [0, T] \to \mathcal{M}$ satisfying the boundary conditions $\boldsymbol{\theta}(0) = \boldsymbol{\theta}_1, \boldsymbol{\theta}(T) = \boldsymbol{\theta}_2$ whose arclength is minimal among such curves. The length of such a minimal curve defines a distance function on $\mathcal{M}$. In Euclidean space, where $\boldsymbol{G}_0(\boldsymbol{\theta}) \equiv \boldsymbol{I}$, the shortest curve connecting $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ is simply a straight line with the Euclidean length $\|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2$.

As mentioned above, while standard HMC algorithms explore the target distribution more efficiently, they nevertheless fail to move between isolated modes since these modes are separated by high energy barriers (Sminchisescu and Welling 2011). To address this issue, we propose to replace the base metric $\boldsymbol{G}_0$ with a new metric for which the distance between modes is shortened. This way, we can facilitate moving between modes by creating "wormholes" between them.

Let $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$ be two modes of the target distribution. We define a straight line segment, $\boldsymbol{v}_W := \hat{\boldsymbol{\theta}}_2 - \hat{\boldsymbol{\theta}}_1$, and refer to a small neighborhood (tube) of the line segment as a *wormhole*. Next, we define a *wormhole metric*, $\boldsymbol{G}_W(\boldsymbol{\theta})$, in the vicinity of the wormhole. The metric $\boldsymbol{G}_W(\boldsymbol{\theta})$ is an inner product assigning a non-negative real number to a pair of tangent vectors $\boldsymbol{u}, \boldsymbol{w}$: $\boldsymbol{G}_W(\boldsymbol{\theta})(\boldsymbol{u}, \boldsymbol{w}) \in \mathbb{R}^+$. To shorten the distance in the direction of $\boldsymbol{v}_W$, we project both $\boldsymbol{u}, \boldsymbol{w}$ to the plane normal to $\boldsymbol{v}_W$ and then take the Euclidean inner product of those projected vectors. We set $\boldsymbol{v}_W^* = \boldsymbol{v}_W / \|\boldsymbol{v}_W\|$ and define a *pseudo wormhole metric* $\boldsymbol{G}_W^*$ as follows:

$$\begin{aligned} \boldsymbol{G}_W^*(\boldsymbol{u}, \boldsymbol{w}) &:= \langle \boldsymbol{u} - \langle \boldsymbol{u}, \boldsymbol{v}_W^* \rangle \boldsymbol{v}_W^*, \boldsymbol{w} - \langle \boldsymbol{w}, \boldsymbol{v}_W^* \rangle \boldsymbol{v}_W^* \rangle \\ &= \boldsymbol{u}^\mathsf{T} [\boldsymbol{I} - \boldsymbol{v}_W^* (\boldsymbol{v}_W^*)^\mathsf{T}] \boldsymbol{w} \end{aligned}$$

Note that $\boldsymbol{G}_W^* := \boldsymbol{I} - \boldsymbol{v}_W^* (\boldsymbol{v}_W^*)^\mathsf{T}$ is semi-positive definite (degenerate at $\boldsymbol{v}_W^* \neq 0$). We modify this metric to make it positive definite, and define the *wormhole metric* $\boldsymbol{G}_W$ as follows:

$$\boldsymbol{G}_W = \boldsymbol{G}_W^* + \varepsilon \boldsymbol{v}_W^* (\boldsymbol{v}_W^*)^\mathsf{T} = \boldsymbol{I} - (1 - \varepsilon) \boldsymbol{v}_W^* (\boldsymbol{v}_W^*)^\mathsf{T} \tag{3}$$

where $0 < \varepsilon \ll 1$ is a small positive number.

To see that the wormhole metric $\boldsymbol{G}_W$ in fact shortens the distance between $\hat{\boldsymbol{\theta}}_1$ and $\hat{\boldsymbol{\theta}}_2$, consider a simple case where $\boldsymbol{\theta}(t)$ follows a straight line: $\boldsymbol{\theta}(t) = \boldsymbol{\theta}_1 + \boldsymbol{v}_W t, t \in [0, 1]$. In this case, the distance under $\boldsymbol{G}_W$ is

$$\mathrm{dist}(\hat{\boldsymbol{\theta}}_1, \hat{\boldsymbol{\theta}}_2) = \int_0^1 \sqrt{\boldsymbol{v}_W^\mathsf{T} \boldsymbol{G}_W \boldsymbol{v}_W} dt = \sqrt{\varepsilon} \|\boldsymbol{v}_W\| \ll \|\boldsymbol{v}_W\|$$

which is much smaller than the Euclidean distance.

Next, we define the overall metric, $\boldsymbol{G}$, for the whole parameter space of $\boldsymbol{\theta}$ as a weighted sum of the base metric $\boldsymbol{G}_0$ and the wormhole metric $\boldsymbol{G}_W$,

$$\boldsymbol{G}(\boldsymbol{\theta}) = (1 - \mathfrak{m}(\boldsymbol{\theta})) \boldsymbol{G}_0(\boldsymbol{\theta}) + \mathfrak{m}(\boldsymbol{\theta}) \boldsymbol{G}_W \tag{4}$$

where $\mathfrak{m}(\boldsymbol{\theta}) \in (0, 1)$ is a mollifying function designed to make the wormhole metric $\boldsymbol{G}_W$ influential in the vicinity of the wormhole only. In this paper, we choose the following mollifier:

$$\mathfrak{m}(\boldsymbol{\theta}) := \exp\{-(\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_1\| + \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_2\| - \|\hat{\boldsymbol{\theta}}_1 - \hat{\boldsymbol{\theta}}_2\|)/F\} \tag{5}$$

where the *influence factor* $F > 0$, is a free parameter that can be tuned to modify the extent of the influence of $\boldsymbol{G}_W$: decreasing $F$ makes the influence of $\boldsymbol{G}_W$ more restricted around the wormhole. The resulting metric leaves the base metric almost intact outside of the wormhole, while making the transition of the metric from outside to inside smooth. Within the wormhole, the trajectories are mainly guided in the wormhole direction $\boldsymbol{v}_W^*$: $\boldsymbol{G}(\boldsymbol{\theta}) \approx \boldsymbol{G}_W$, so $\boldsymbol{G}(\boldsymbol{\theta})^{-1} \approx \boldsymbol{G}_W{}^{-1}$ has the dominant eigen-vector $\boldsymbol{v}_W^*$ (with eigen-value $1/\varepsilon \gg 1$), thereafter $\boldsymbol{v} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{G}(\boldsymbol{\theta})^{-1})$ tends to be directed in $\boldsymbol{v}_W^*$.

We use the modified metric (4) in RMHMC and refer to the resulting algorithm as Wormhole Hamiltonian Monte

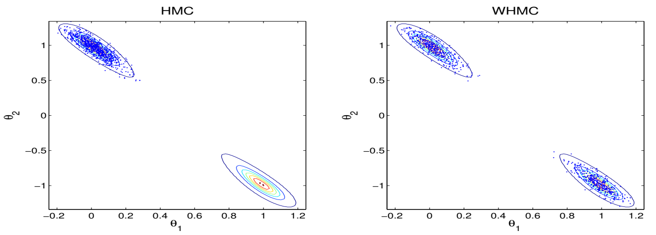


Figure 1: Comparing HMC and WHMC in terms of sampling from a two-dimensional posterior distribution with two isolated modes (Welling and Teh 2011).

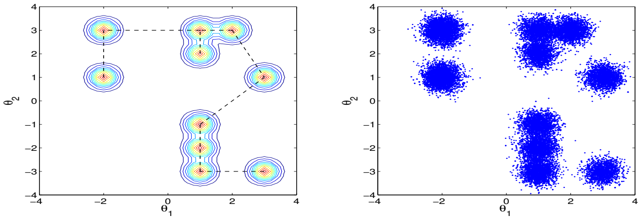


Figure 2: Sampling from a mixture of 10 Gaussian distributions with dimension $D = 100$ using WHMC with a vector field $\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{v})$ to enforce moving between modes in higher dimensions. Dashed lines show the minimal spanning tree.

Carlo (WHMC). Figure 1 compares WHMC to standard HMC based on the following illustrative example appeared in the paper by (Welling and Teh 2011):

$$\begin{aligned} \theta_d &\sim \mathcal{N}(\theta_d, \sigma_d^2), \quad d = 1, 2. \\ x_i &\sim \frac{1}{2}\mathcal{N}(\theta_1, \sigma_x^2) + \frac{1}{2}\mathcal{N}(\theta_1 + \theta_2, \sigma_x^2). \end{aligned}$$

Here, we set $\theta_1 = 0, \theta_2 = 1, \sigma_1^2 = 10, \sigma_2^2 = 1, \sigma_x^2 = 2$, and generate 1000 data points from the above model. In Figure 1, the dots show the posterior samples of $(\theta_1, \theta_2)$ given the simulated data. While HMC is trapped in one mode, WHMC moves easily between the two modes. For this example, we set $\boldsymbol{G}_0 = \boldsymbol{I}$ to make WHMC comparable to standard HMC. Further, we use 0.03 and 0.3 for $\varepsilon$ and $F$ respectively.

For more than two modes, we can construct a network of wormholes by connecting any two modes with a wormhole. Alternatively, we can create a wormhole between neighboring modes only. In this paper, we define the neighborhood using a *minimal spanning tree* (Kleinberg and Tardos 2005).

The above method could suffer from two potential shortcomings in higher dimensions. First, the effect of wormhole metric could diminish fast as the sampler leaves one mode towards another mode. Second, such mechanism, which modifies the dynamics in the existing parameter space, could interfere with the native HMC dynamics in the neighborhood of a wormhole, possibly preventing the sampler from properly exploring areas around the modes as well as some low probability regions.

To address the first issue, we add an external vector field to enforce the movement between modes. More specifically, we define a vector field, $\boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{v})$, in terms of the position parameter $\boldsymbol{\theta}$ and the velocity vector $\boldsymbol{v} = \boldsymbol{G}(\boldsymbol{\theta})^{-1}\boldsymbol{p}$ as follows:

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{v}) &:= \exp\{-V(\boldsymbol{\theta})/(DF)\} U(\boldsymbol{\theta}) \langle \boldsymbol{v}, \boldsymbol{v}_W^* \rangle \boldsymbol{v}_W^* \\ &= \mathfrak{m}(\boldsymbol{\theta}) \langle \boldsymbol{v}, \boldsymbol{v}_W^* \rangle \boldsymbol{v}_W^* \end{aligned}$$

with mollifier $\mathfrak{m}(\boldsymbol{\theta}) := \exp\{-V(\boldsymbol{\theta})/(DF)\}$, where $D$ is the dimension, $F > 0$ is the influence factor, and $V(\boldsymbol{\theta})$ is a vicinity function indicating the Euclidean distance from the line segment $\boldsymbol{v}_W$,

$$V(\boldsymbol{\theta}) := \langle \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_1, \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_2 \rangle + |\langle \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_1, \boldsymbol{v}_W^* \rangle| |\langle \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_2, \boldsymbol{v}_W^* \rangle| \tag{6}$$

The resulting vector field has three properties: 1) it is confined to a neighborhood of each wormhole, 2) it enforces the movement along the wormhole, and 3) its influence diminishes at the end of the wormhole when the sampler reaches another mode. Such a vector field acts as an external driving force on the sampler. In high dimensions, this approach works more effectively than the wormhole metric discussed above.

After adding the vector field, we modify the Hamiltonian equation governing the evolution of $\boldsymbol{\theta}$ as follows:

$$\dot{\boldsymbol{\theta}} = \boldsymbol{v} + \boldsymbol{f}(\boldsymbol{\theta}, \boldsymbol{v}) \tag{7}$$

We also need to adjust the Metropolis acceptance probability accordingly since the transformation is not volume preserving. (More details are provided in the supplementary file.) Figure 2 illustrates this approach based on sampling from a mixture of 10 Gaussian distributions with dimension $D = 100$.

To address the second issue, we allow the wormholes to pass through an extra auxiliary dimension to avoid their interference with the existing HMC dynamics in the given parameter space. In particular we introduce an auxiliary variable $\theta_{D+1} \sim \mathcal{N}(0, 1)$ corresponding to an auxiliary dimension. We use $\tilde{\boldsymbol{\theta}} := (\boldsymbol{\theta}, \theta_{D+1})$ to denote the position parameters in the resulting $D + 1$ dimensional space $\mathcal{M}^D \times \mathbb{R}$. $\theta_{D+1}$ can be viewed as random noise independent of $\boldsymbol{\theta}$ and contributes $\frac{1}{2}\theta_{D+1}^2$ to the total potential energy. Correspondingly, we augment velocity $\boldsymbol{v}$ with one extra dimension, denoted as $\tilde{\boldsymbol{v}} := (\boldsymbol{v}, v_{D+1})$. At the end of the sampling, we project $\tilde{\boldsymbol{\theta}}$ to the original parameter space and discard $\theta_{D+1}$.

We refer to $\mathcal{M}^D \times \{-h\}$ as the *real world*, and call $\mathcal{M}^D \times \{+h\}$ the *mirror world*. Here, $h$ is half of the distance between the two worlds, and it should be in the same scale as the average distance between the modes. For most of the examples discussed here, we set $h = 1$. Figure 3 illustrates how the two worlds are connected by networks of wormholes. When the sampler is near a mode $(\hat{\boldsymbol{\theta}}_1, -h)$ in the real world, we build a wormhole network by connecting it to all the modes in the mirror world. Similarly, we connect the corresponding mode in the mirror world, $(\hat{\boldsymbol{\theta}}_1, +h)$, to all the modes in the real world. Such construction allows the sampler to jump from one mode in the real world to the same mode in the mirror world and vice versa. This way, the algorithm can effectively sample from the vicinity of a mode, while occasionally jumping from one mode to another.

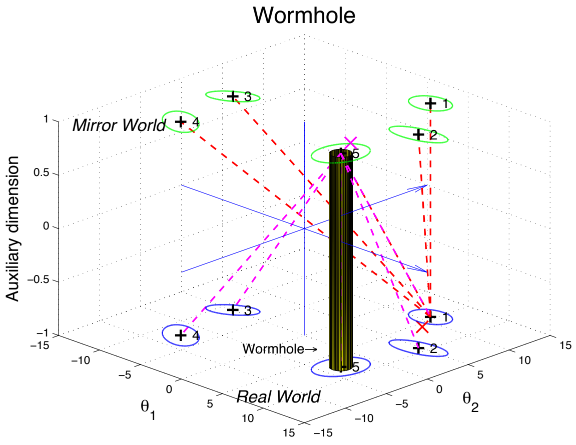


Figure 3: Illustrating a wormhole network connecting the real world to the mirror world ($h = 1$). As an example, the cylinder shows a wormhole connecting mode 5 in the real world to its mirror image. The dashed lines show two sets of wormholes. The red lines shows the wormholes when the sampler is close to mode 1 in the real world, and the magenta lines show the wormholes when the sampler is close to mode 5 in the mirror world.

The attached supplementary file provides the details of our algorithm (Algorithm 1), along with the proof of convergence and its implementation in MATLAB.

## Mode Searching After Regeneration

So far, we assumed that the locations of modes are known. This is of course not a realistic assumption in many situations. In this section, we relax this assumption by extending our method to search for new modes proactively and to update the network of wormholes dynamically. In general, however, allowing such adaptation to take place infinitely often will disturb the stationary distribution of the chain, rendering the process no longer Markov (Gelfand and Dey 1994; Gilks, Roberts, and Sahu 1998). To avoid this issue, we use the *regeneration* method discussed by (Nummelin 1984; Mykland, Tierney, and Yu 1995; Gilks, Roberts, and Sahu 1998; Brockwell and Kadane 2005).

Informally, a regenerative process "starts again" probabilistically at a set of times, called *regeneration times* (Brockwell and Kadane 2005). At regeneration times, the transition mechanism can be modified based on the entire history of the chain up to that point without disturbing the consistency of MCMC estimators.

### Identifying Regeneration Times

The main idea behind finding regeneration times is to regard the transition kernel $T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)$ as a mixture of two kernels, $Q$ and $R$ (Nummelin 1984; Ahn, Chen, and Welling 2013),

$$T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t) = S(\boldsymbol{\theta}_t)Q(\boldsymbol{\theta}_{t+1}) + (1 - S(\boldsymbol{\theta}_t))R(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)$$

where $Q(\boldsymbol{\theta}_{t+1})$ is an *independence kernel*, and the *residual kernel* $R(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)$ is defined as follows:

$$R(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)=\begin{cases} \dfrac{T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t) - S(\boldsymbol{\theta}_t)Q(\boldsymbol{\theta}_{t+1})}{1 - S(\boldsymbol{\theta}_t)}, & S(\boldsymbol{\theta}_t) \in [0, 1) \\ 1, & S(\boldsymbol{\theta}_t) = 1 \end{cases}$$

$S(\boldsymbol{\theta}_t)$ is the mixing coefficient between the two kernels such that

$$T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t) \geq S(\boldsymbol{\theta}_t)Q(\boldsymbol{\theta}_{t+1}), \forall \boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1} \tag{8}$$

Now suppose that at iteration $t$, the current state is $\boldsymbol{\theta}_t$. To implement this approach, we first generate $\boldsymbol{\theta}_{t+1}$ using the original transition kernel $\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t \sim T(\cdot|\boldsymbol{\theta}_t)$. Then, we sample $B_{t+1}$ from a Bernoulli distribution with probability

$$r(\boldsymbol{\theta}_t, \boldsymbol{\theta}_{t+1}) = \frac{S(\boldsymbol{\theta}_t)Q(\boldsymbol{\theta}_{t+1})}{T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)} \tag{9}$$

If $B_{t+1} = 1$, a regeneration has occurred, then we discard $\boldsymbol{\theta}_{t+1}$ and sample it from the independence kernel $\boldsymbol{\theta}_{t+1} \sim Q(\cdot)$. At regeneration times, we redefine the dynamics using the past sample path.

Ideally, we would like to evaluate regeneration times in terms of WHMC's transition kernel. In general, however, this is quite difficult for such Metropolis algorithm. On the other hand, regenerations are easily achieved for the independence sampler (i.e., the proposed state is independent from the current state) as long as the proposal distribution is close to the target distribution (Gilks, Roberts, and Sahu 1998). Therefore, we can specify a hybrid sampler that consists of the original proposal distribution (WHMC) and the independence sampler, and adapt both proposal distributions whenever a regeneration is obtained on an independence-sampler step (Gilks, Roberts, and Sahu 1998). In our method, we systematically alternate between WHMC and the independence sampler while evaluating regeneration times based on the independence sampler only.

To this end, we follow (Ahn, Chen, and Welling 2013) and specify our independence sampler as a mixture of Gaussians located at the previously identified modes. More specifically, $T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t)$, $S(\boldsymbol{\theta}_t)$ and $Q(\boldsymbol{\theta}_{t+1})$ are defined as follows to satisfy (8):

$$T(\boldsymbol{\theta}_{t+1}|\boldsymbol{\theta}_t) = q(\boldsymbol{\theta}_{t+1}) \min\left\{1, \frac{\pi(\boldsymbol{\theta}_{t+1})/q(\boldsymbol{\theta}_{t+1})}{\pi(\boldsymbol{\theta}_t)/q(\boldsymbol{\theta}_t)}\right\} \tag{10}$$

$$S(\boldsymbol{\theta}_t) = \min\left\{1, \frac{c}{\pi(\boldsymbol{\theta}_t)/q(\boldsymbol{\theta}_t)}\right\} \tag{11}$$

$$Q(\boldsymbol{\theta}_{t+1}) = q(\boldsymbol{\theta}_{t+1}) \min\left\{1, \frac{\pi(\boldsymbol{\theta}_{t+1})/q(\boldsymbol{\theta}_{t+1})}{c}\right\} \tag{12}$$

where $q(\cdot)$ is the independence proposal kernel, which is specified using a mixture of Gaussians with means fixed at the $k$ known modes prior to regeneration. The covariance matrix for each mixture component is set to the inverse observed Fisher information (i.e., Hessian) evaluated at the mode. The relative weights are initialized as $1/k$ and updated at regeneration times according to the number to times each mode has been visited. Algorithm 2 in the supplementary file shows the steps for this method.

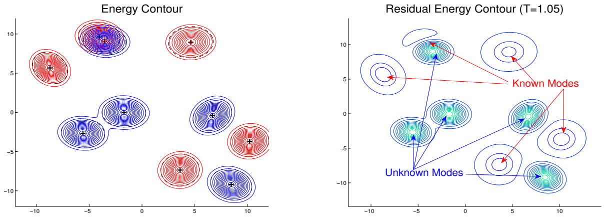


Figure 4: Left panel: True energy function (red: known modes, blue: unknown modes). Right panel: Residual energy function at $T = 1.05$.

## Identifying New Modes

When the chain regenerates, we can search for new modes, modify the transition kernel by including newly found modes in the mode library, and update the wormhole network accordingly. This way, starting with a limited number of modes (identified by some preliminary optimization method), WHMC could discover unknown modes on the fly without affecting the stationarity of the chain.

To search for new modes after regeneration, as opposed to frequently rediscovering the known ones, we propose to remove/down-weight the known modes using the history of the chain up to the regeneration time and run an optimization algorithm on the resulting *residual density*, or equivalently, on the corresponding *residual energy* (i.e., minus log of density). To this end, we fit a mixture of Gaussians with the best knowledge of modes (locations, Hessians and relative weights) prior to the regeneration. The *residual density* function could be simply defined as $\pi_{\mathbf{r}}(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}) - q(\boldsymbol{\theta})$ with the corresponding *residual potential energy* as follows,

$$U_{\mathbf{r}}(\boldsymbol{\theta}) = \log(\pi_{\mathbf{r}}(\boldsymbol{\theta}) + c) = -\log(\pi(\boldsymbol{\theta}) - q(\boldsymbol{\theta}) + c)$$

where the constant $c > 0$ is used to make the term inside the log function positive. To avoid completely flat regions (e.g., when a Gaussian distribution provides a good approximation around the mode), which could cause gradient-based optimization methods to fail, we could use the following *tempered residual potential energy* instead:

$$U_{\mathbf{r}}(\boldsymbol{\theta}, T) = -\log\left(\pi(\boldsymbol{\theta}) - \exp\left(\frac{1}{T}\log q(\boldsymbol{\theta})\right) + c\right)$$

where $T$ is the temperature. Figure 4 illustrates this concept.

When the optimizer finds new modes, they are added to the existing mode library, and the wormhole network is updated accordingly.

## Empirical Results

In this section, we evaluate the performance of our method, henceforth called Wormhole Hamiltonian Monte Carlo (WHMC), using three examples. The first example involves sampling from mixtures of Gaussian distributions with varying number of modes and dimensions. In this example, which is also discussed by (Ahn, Chen, and Welling 2013), the locations of modes are assumed to be known. The second example, which was originally proposed by (Ihler et al. 2005), involves inference regarding the locations of sensors in a network. For our third example, we also use mixtures of Gaussian distributions, but this time we assume that the locations of modes are unknown.

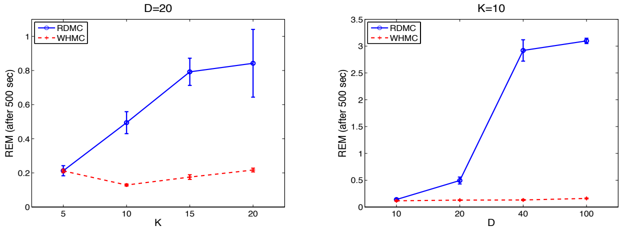


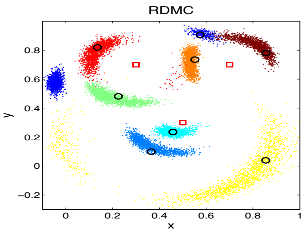
Figure 5: Comparing WHMC to RDMC using $K$ mixtures of $D$-dimensional Gaussians. Left panel: REM (along with 95% confidence interval based on 10 MCMC chains) for varying number of mixture components, $K = 5, 10, 15, 20$, with fixed dimension, $D = 20$. Right panel: REM (along with 95% confidence interval based on 10 MCMC chains) for varying number of dimensions, $D = 10, 20, 40, 100$, with fixed number of mixture components, $K = 10$.

We evaluate our method's performance by comparing it to Regeneration Darting Monte Carlo (RDMC) (Ahn, Chen, and Welling 2013), which is one of the most recent algorithms designed for sampling from multimodal distributions based on the Darting Monte Carlo (DMC) (Sminchisescu and Welling 2011) approach. DMC defines high density regions around the modes. When the sampler enters these regions, a jump between the regions will be attempted. RDMC enriches the DMC method by using the regeneration approach (Mykland, Tierney, and Yu 1995; Gilks, Roberts, and Sahu 1998). However, these methods tend to fail in high dimensional spaces where modes are isolated, small and hard to hit.

We compare the two methods (i.e., WHMC and RDMC) in terms of Relative Error of Mean (REM) proposed by (Ahn, Chen, and Welling 2013). The value of REM at time $t$ is $\mathrm{REM}(t) = \|\overline{\theta(t)} - \theta^*\|_1/\|\theta^*\|_1$, which summarizes the error in approximating the expectation of variables across all dimensions. Here, $\theta^*$ is the true mean and $\overline{\theta(t)}$ is the mean estimated by MCMC samples collected up to time $t$. We examine $\mathrm{REM}(t)$ as a function of $t$ until a pre-specified time representing a given computational budget (Ahn, Chen, and Welling 2013; Anoop Korattikara 2014). Because RDMC uses standard HMC algorithm with a flat metric, we also use the baseline metric $\boldsymbol{G}_0 \equiv \boldsymbol{I}$ to make the two algorithms comparable. Our approach, however, can be easily extended to other metrics such as Fisher information.

## Mixture of Gaussians with Known Modes

First, we evaluate the performance of our method based on sampling from $K$ mixtures of $D$-dimensional Gaussian distributions with *known* modes. (We relax this assumption later.) The means of these distributions are randomly generated from $D$-dimensional uniform distributions such that



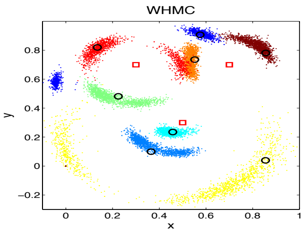


Figure 6: Posterior samples for sensor locations using RDMC (left) and WHMC (right). Squares show the locations of reference sensors, point clouds show the marginal distributions, and circles show the corresponding modes.

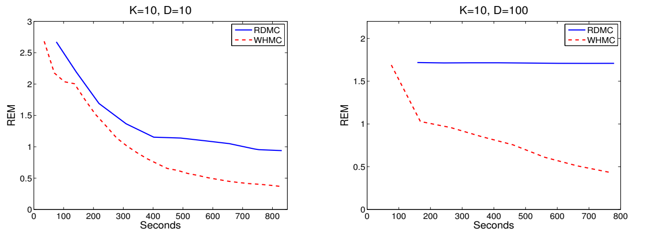


Figure 7: Comparing WHMC to RDMC in terms of REM using $K = 10$ mixtures of $D$-dimensional Gaussians with $D = 20$ (left panel) and $D = 100$ (right panel).

the average pairwise distances remains around 20. The corresponding covariance matrices are constructed in a way that mixture components have different density functions. Simulating samples from the resulting $D$ dimensional mixture of $K$ Gaussians is challenging because the modes are far apart and the high density regions have different shapes.

The left panel of Figure 5 compares the two methods for varying number of mixture components, $K = 5, 10, 15, 20$, with fixed dimension ($D = 20$). The right panel shows the results for varying number of dimensions, $D = 10, 20, 40, 100$, with fixed number of mixture components ($K = 10$). For both scenarios, we stop the two algorithms after 500 seconds and compare their REM. As we can see, WHMC has substantially lower REM compared to RDMC, especially when the number of modes and dimensions increase. As we can see, in dimensions above 20, RDMC is trapped in a subset of modes.

## Sensor Network Localization

For our second example, we use a problem previously discussed by (Ihler et al. 2005) and (Ahn, Chen, and Welling 2013). We assume that $N$ sensors are scattered in a planar region with $2d$ locations denoted as $\{x_i\}_{i=1}^N$. The distance $Y_{ij}$ between a pair of sensors $(x_i, x_j)$ is observed with probability $\pi(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2R^2))$. If the distance is in fact observed ($Y_{ij} > 0$), then $Y_{ij}$ follows a Gaussian distribution $\mathcal{N}(\|x_i - x_j\|, \sigma^2)$ with small $\sigma$; otherwise $Y_{ij} = 0$. That is,

$$Z_{ij} = I(Y_{ij} > 0)|x \sim \text{Binom}(1, \pi(x_i, x_j))$$
$$Y_{ij}|Z_{ij} = 1, x \sim \mathcal{N}(\|x_i - x_j\|, \sigma^2)$$

where $Z_{ij}$ is a binary indicator set to 1 if the distance between $x_i$ and $x_j$ is observed.

Given a set of observations $Y_{ij}$ and prior distribution of $x$, which is assumed to be uniform in this example, it is of interest to infer the posterior distribution of all the sensor locations. Following (Ahn, Chen, and Welling 2013), we set $N = 8, R = 0.3, \sigma = 0.02$, and add three additional base sensors with known locations to avoid ambiguities of translation, rotation, and negation (mirror symmetry). The locations of sensors form a multimodal distribution ($D = 16$).

Figure 6 shows the posterior samples based on the two methods. As we can see, RDMC very rarely visits one of the modes (shown in red in the top middle part); whereas, WHMC generates enough samples from this mode to make it discernible. As a result, WHMC converges to a substantially lower REM (0.02 vs. 0.13) after 500 seconds.

## Mixture of Gaussians with Unknown Modes

We now evaluate our method's performance in terms of searching for new modes and updating the network of wormholes. For this example, we simulate a mixture of 10 $D$-dimensional Gaussian distributions, with $D = 10, 100$, and compare our method to RDMC. While RDMC runs four parallel HMC chains initially to discover a subset of modes and to fit a truncated Gaussian distribution around each identified mode, we run four parallel optimizers (different starting points) using the BFGS method. At regeneration times, each chain of RDMC uses the Dirichlet process mixture model to fit a new truncated Gaussian around modes and possibly identify new modes. We on the other hand run the BFGS algorithm based on the residual energy function (with $T = 1.05$) to discover new modes for each chain. Figure 7 shows WHMC reduces REM much faster than RDMC for both $D = 10$ and $D = 100$. Here, the recorded time (horizontal axis) accounts for the computational overhead for adapting the transition kernels. For $D = 10$, our method has a substantially lower REM compared to RDMC. For $D = 100$, while our method identifies new modes over time and reduces REM substantially, RDMC fails to identify new modes so as a result its REM remains high over time.

## Conclusions and Discussion

We have proposed a new algorithm for sampling from multimodal distributions. Using empirical results, we have shown that our method performs well in high dimensions.

Although the examples discussed here use a flat base metric $\boldsymbol{I}$, with the computational complexity of $\mathcal{O}(D)$, our method can be easily extended to more informative base metric, such as Fisher information with the computational complexity of $\mathcal{O}(D^3)$, to adapt to the local geometry.

## Acknowledgements

This research is supported by NSF grants IIS-1216045 and DMS-1201569 and NIH grant R01-AI107034.

## References

Ahn, S.; Chen, Y.; and Welling, M. 2013. Distributed and adaptive darting Monte Carlo through regenerations. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AI Stat)*.

Amari, S., and Nagaoka, H. 2000. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical monographs*. Oxford University Press.

Anoop Korattikara, Yutian Chen, M. W. 2014. Austerity in mcmc land: Cutting the metropolis-hastings budget. In Xing, E. P., and Jebara, T., eds., *Proceedings of the 31th International Conference on Machine Learning (ICML-14)*, volume 32, 181–189.

Braak, C. J. F. T. 2006. A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces. *Statistics and Computing* 16(3):239–249.

Brockwell, A. E., and Kadane, J. B. 2005. Identification of regeneration times in mcmc simulation, with application to adaptive schemes. *Journal of Computational and Graphical Statistics* 14:436–458.

Celeux, G.; Hurn, M.; and Robert, C. P. 2000. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association* 95:957–970.

Craiu, R. V.; R., J.; and Y., C. 2009. Learn from thy neighbor: Parallel-chain and regional adaptive mcmc. *Journal of the American Statistical Association* 104(488):1454–1466.

Duane, S.; Kennedy, A. D.; Pendleton, B. J.; and Roweth, D. 1987. Hybrid Monte Carlo. *Physics Letters B* 195(2):216 – 222.

Gelfand, A. E., and Dey, D. K. 1994. Bayesian model choice: Asymptotic and exact calculation. *Journal of the Royal Statistical Society. Series B.* 56(3):501–514.

Gilks, W. R.; Roberts, G. O.; and Sahu, S. K. 1998. Adaptive markov chain monte carlo through regeneration. *Journal of the American Statistical Association* 93(443):pp. 1045–1054.

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

Hinton, G. E.; Welling, M.; and Mnih, A. 2004. Wormholes improve contrastive divergence. In *Advances in Neural Information Processing Systems 16*.

Ihler, A. T.; III, J. W. F.; Moses, R. L.; and Willsky, A. S. 2005. Nonparametric belief propagation for self-localization of sensor networks. *IEEE Journal on Selected Areas in Communications* 23(4):809–819.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by Simulated Annealing. *Science, Number 4598, 13 May 1983* 220(4598):671–680.

Kleinberg, J., and Tardos, E. 2005. *Algorithm Design*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Laskey, K. B., and Myers, J. W. 2003. Population Markov Chain Monte Carlo. *Machine Learning* 50:175–196.

Mykland, P.; Tierney, L.; and Yu, B. 1995. Regeneration in markov chain samplers. *Journal of the American Statistical Association* 90(429):pp. 233–241.

Neal, R. M. 1993. *Probabilistic Inference Using Markov Chain Monte Carlo Methods*. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto.

Neal, R. M. 1996. *Bayesian Learning for Neural Networks*. Secaucus, NJ, USA: Springer-Verlag New York, Inc.

Neal, R. M. 2001. Annealed importance sampling. *Statistics and Computing* 11(2):125–139.

Neal, R. M. 2010. MCMC using Hamiltonian dynamics. In Brooks, S.; Gelman, A.; Jones, G.; and Meng, X. L., eds., *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC.

Nummelin, E. 1984. *General Irreducible Markov Chains and Non-Negative Operators*, volume 83 of *Cambridge Tracts in Mathematics*. Cambridge University Press.

Rudoy, D., and Wolfe, P. J. 2006. Monte carlo methods for multi-modal distributions. In *Signals, Systems and Computers, 2006. ACSSC '06. Fortieth Asilomar Conference on*, 2019–2023.

Sminchisescu, C., and Triggs, B. 2002. Building roadmaps of local minima of visual models. In *In European Conference on Computer Vision*, 566–582.

Sminchisescu, C., and Welling, M. 2011. Generalized darting monte carlo. *Pattern Recognition* 44(10-11).

Warnes, G. R. 2001. The normal kernel coupler: An adaptive Markov Chain Monte Carlo method for efficiently sampling from multi-modal distributions. Technical Report Technical Report No. 395, University of Washington.

Welling, M., and Teh, Y. W. 2011. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the International Conference on Machine Learning*.