Fixed-Distance Hamiltonian Monte Carlo
Supplementary Material

Hadi Mohasel Afshar
CSIRO's Data61
Eveleigh, NSW 2015, Australia
`Hadi.Afshar@data61.csiro.au`

Sally Cripps
CSIRO's Data61 & The University of Sydney
Eveleigh, NSW 2015, Australia
`Sally.Cripps@data61.csiro.au`

In Section 1, we further clarify what we mean (and what we do not mean) by the terms *bias* and *reversibility* to prevent any confusion or misunderstanding.

In Section 2, we present a formal proof for the reversibility of the FDHMC-Leapfrogs.

In Section 3, we present the algorithms which we have used to tune the FDHMC's hyper-parameters.

In Section 4, we study the sensitivity of the performance of FDHMC with respect to the chosen fixed-distance hyper-parameter. We also study the performance of the presented tuning algorithm in finding a reasonable fixed-distance value.

Finally, in Section 5, we plot samples drawn from FDHMC as well as 3 existing static/dynamic HMC algorithms on a multimodal distribution.

1 Clarifying the terminology

Different communities use different terminologies and this, in cases, may cause misunderstandings. In the main paper, we have already defined and used the terms: (a) *biased/unbiased sampling*, in the sense they are usually used by the Machine Learning community, and (b) *reversibility*, in the sense it is often used in the context of the Hamiltonian Monte Carlo sampling (Neal, 2011).

However, these terms have other meanings in different contexts. To resolve any possible confusion, here we explain exactly in what sense we use these terms and in what sense we do not:

Biased/unbiased sampling (Sense 1). An MCMC sampling algorithm is unbiased if the chain, $\{\mathbf{q}^{(c)}\}_{c=1}^{N}$, of samples that it generates, converges to the target, (say, $\pi_{\mathbf{Q}}(\mathbf{q})$), in distribution. That is, when the length of the MCMC chain tends to infinity, $N \to \infty$, then the number of samples that are within any subset, $\mathcal{A}$, of the probability space is proportional to the target probability mass associated with that subset, $\int_{\mathcal{A}} \pi_{\mathbf{Q}}(\mathbf{q}) d\mathbf{q}$. If this condition does not hold, then we say that the sampler is biased i.e. does not converge to the target distribution.

This is the terminology that we have used throughout the paper. However the statisticians often use the term *bias* in another sense:

Biased/unbiased estimation (Sense 2). If the difference between an estimated expected value and the true value of the parameter that is being estimated is 0, then the estimator is unbiased otherwise it is biased.

Clearly if a sampler is unbiased in Sense 1 then it is also unbiased in Sense 2 as well, but the opposite does not always hold. In this paper, we have not used the term *bias* in Sense 2.

Reversibility (of a deterministic mapping) (Sense 1). A deterministic transformation, $\mathcal{F}$, is called *reversible* if it is a *diffeomorphism* i.e. a differentiable mapping that is the inverse of itself:

$$\mathcal{F}(\mathbf{x}) = \mathbf{x}', \quad \text{if and only if,} \quad \mathcal{F}(\mathbf{x}') = \mathbf{x}. \qquad \text{(Reversibility condition)} \tag{1}$$

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

In the literature of the Hamiltonian Monte Carlo sampling, *reversibility* (of dynamics) is often defined as above (Neal, 2011) and throughout the present paper, the term *reversibility* is exclusively used in this sense.

Reversibility (of a Markov Chain) (Sense 2). A Markov chain is called *reversible* or *time reversible* if it satisfies the *detailed balance* condition:

$$\pi_{\mathbf{Q}}(\mathbf{q})P(\mathbf{q} \to \mathbf{q}') = \pi_{\mathbf{Q}}(\mathbf{q}')P(\mathbf{q}' \to \mathbf{q}), \qquad \forall \mathbf{q}, \mathbf{q}'$$

where $P(\mathbf{q} \to \mathbf{q}')$ is the transition kernel probability density from state $\mathbf{q}$ to state $\mathbf{q}'$.

The reversibility of the mapping, $\mathcal{F}$, that is used in an RJMCMC sampler (Sense 1) is required for the detailed balance condition of the MCMC chain (i.e. reversibility in Sense 2). However, to avoid any ambiguity we have not used the term *reversibility* in its second sense.

2 Proof of the reversibility of the Fixed-distance Leapfrog Mechanism (FD-Leapfrogs)

Here is the FD-Leapfrogs algorithm (as presented in the main paper):

FD-Leapfrogs. Given the leapfrog step-size, ϵ, and the total evolution distance, $\mathfrak{D}$, as hyper-parameters, and an input vector, $(\mathbf{q}, \mathbf{p} := \mathbf{p}^{[0]}, \tau)$ where $\mathbf{q} \sim \pi_{\mathbf{Q}}(\mathbf{q})$ (on $\mathbb{R}^n$), $\mathbf{p} \sim \pi_{\mathbf{P}}(\mathbf{p})$ (on $\mathbb{R}^n$), and $\tau \sim$ Unif$(0, \epsilon)$, FD-Leapfrogs, $\mathcal{F} : \mathbb{R}^{2n+1} \to \mathbb{R}^{2n+1}$, maps $(\mathbf{q}, \mathbf{p}, \tau)$ to $(\mathbf{q}', \mathbf{p}', \tau')$ as follows:

1. Initial position τ-step: Evolve the position, $\mathbf{q}$, for time τ with fixed momentum, $\mathbf{p}$.

$$\mathbf{q}^{[1]} := \mathbf{q} + \tau\mathbf{p} \tag{2}$$

2. Initial momentum full-step:

$$\mathbf{p}^{[1]} := \mathbf{p} - \epsilon\nabla U(\mathbf{q}^{[1]}) \tag{3}$$

3. Interchanged position and momentum full-steps: For $i = 2, \ldots, k$, update the position and momentum vectors interchangeably as follows:

$$\mathbf{q}^{[i]} := \mathbf{q}^{[i-1]} + \epsilon\mathbf{p}^{[i-1]} \tag{4}$$

$$\mathbf{p}^{[i]} := \mathbf{p}^{[i-1]} - \epsilon\nabla U(\mathbf{q}^{[i]}) \tag{5}$$

where k is the maximum number of leapfrogs such that the total traversed distance does not exceed $\mathfrak{D}$:

$$d^{[k]}(\mathbf{q}, \mathbf{p}, \tau) \leq \mathfrak{D} < d^{[k+1]}(\mathbf{q}, \mathbf{p}, \tau), \tag{6}$$

with,

$$d^{[j]}(\mathbf{q}, \mathbf{p}, \tau) := \tau\|\mathbf{p}\| + \sum_{i=1}^{j-1} \epsilon\|\mathbf{p}^{[i]}\|, \quad \forall j \in \mathbb{N}. \tag{7}$$

4. Final position τ'-step: With momentum $\mathbf{p}^{[k]}$, the remaining distance,

$$\mathfrak{d} := \mathfrak{D} - d^{[k]}(\mathbf{q}, \mathbf{p}, \tau), \tag{8}$$

is traversed in time,

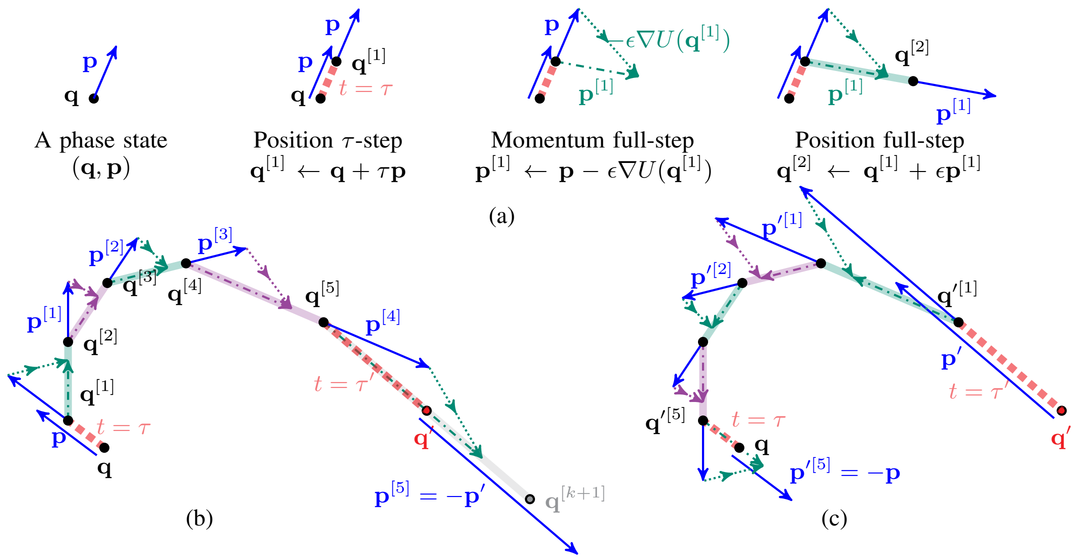
$$\tau' := \frac{\mathfrak{d}}{\|\mathbf{p}^{[k]}\|}, \tag{9}$$

to reach the final position state, $\mathbf{q}'$,

$$\mathbf{q}' := \mathbf{q}^{[k]} + \tau'\mathbf{p}^{[k]}. \tag{10}$$

As such, the fixed traversed distance, $\mathfrak{D}$, is equal to:

$$\begin{aligned}
\mathfrak{D} &= d^{[k]}(\mathbf{q}, \mathbf{p}, \tau) + \mathfrak{d} && \text{, by (8)} \\
&= d^{[k]}(\mathbf{q}, \mathbf{p}, \tau) + \tau'\|\mathbf{p}^{[k]}\| && \text{, by (9)} \\
&= \tau\|\mathbf{p}\| + \sum_{i=1}^{k-1} \epsilon\|\mathbf{p}^{[i]}\| + \tau'\|\mathbf{p}^{[k]}\| && \text{, by (7)}
\end{aligned} \tag{11}$$



(a)

(b)

(c)

Figure 1: Reversibility of FD-leapfrogs: (a) The first 3 steps of FD-leapfrog mechanism. (b) FD-leapfrogs map $(\mathbf{q}, \mathbf{p}, \tau)$ to $(\mathbf{q}', \mathbf{p}', \tau')$. (c) FD-leapfrogs map $(\mathbf{q}', \mathbf{p}', \tau')$ back to $(\mathbf{q}, \mathbf{p}, \tau)$.

5. Negating the final momentum: $(\mathbf{q}', \mathbf{p}', \tau')$ is returned as $\mathcal{F}(\mathbf{q}, \mathbf{p}, \tau)$, where:

$$\mathbf{p}' := -\mathbf{p}^{[k]} \tag{12}$$

Lemma. *Fixed-distance Leapfrog Mechanism (FD-Leapfrogs) is reversible.*

Proof. We should show that $\mathcal{F}(\mathbf{q}', \mathbf{p}', \tau') = (\mathbf{q}, \mathbf{p}, \tau)$. Firstly, we show that for $i = 1, \ldots, k-1$:

$$\mathbf{q}'^{[i]} = \mathbf{q}^{[k+1-i]}, \qquad \mathbf{p}'^{[i]} = -\mathbf{p}^{[k-i]}. \tag{13}$$

The proof is by mathematical induction. The base case is established for $(i = 1)$:

$$\mathbf{q}'^{[1]} \overset{(2)}{=} \mathbf{q}' + \tau'\mathbf{p}' \overset{(12)}{=} \mathbf{q}' - \tau'\mathbf{p}^{[k]} \overset{(10)}{=} \mathbf{q}^{[k]}, \tag{14}$$

$$\mathbf{p}'^{[1]} \overset{(3)}{=} \mathbf{p}' - \epsilon\nabla U(\mathbf{q}'^{[1]}) \overset{(12),(14)}{=} -\mathbf{p}^{[k]} - \epsilon\nabla U(\mathbf{q}^{[k]}) \overset{(5)}{=} -\mathbf{p}^{[k-1]}. \tag{15}$$

In the induction step, we assume that (13) holds for i, and show that $\mathbf{q}'^{[i+1]} = \mathbf{q}^{[k-i]}$ and $\mathbf{p}'^{[i+1]} = -\mathbf{p}^{[k-i-1]}$:

$$\mathbf{q}'^{[i+1]} \overset{(4)}{=} \mathbf{q}'^{[i]} + \epsilon\mathbf{p}'^{[i]} \overset{(13)}{=} \mathbf{q}^{[k+1-i]} - \epsilon\mathbf{p}^{[k-i]} \overset{(4)}{=} \mathbf{q}^{[k-i]}, \tag{16}$$

$$\mathbf{p}'^{[i+1]} \overset{(5)}{=} \mathbf{p}'^{[i]} - \epsilon\nabla U(\mathbf{q}'^{[i+1]}) \overset{(13),(16)}{=} -\mathbf{p}^{[k-i]} - \epsilon\nabla U(\mathbf{q}^{[k-i]}) \overset{(5)}{=} -\mathbf{p}^{[k-i-1]}. \tag{17}$$

Therefore, by following the (backward) interchanged full-steps, we revisit all the forward states in the reversed order till we reach:

$$\mathbf{q}'^{[k]} = \mathbf{q}^{[1]}, \qquad \mathbf{p}'^{[k]} = -\mathbf{p}^{[0]} := -\mathbf{p}. \tag{18}$$

The interchanged full-steps terminates at the state $(\mathbf{q}'^{[k]}, \mathbf{p}'^{[k]})$ (i.e. the (backward) termination condition (6) is satisfied at $k' = k$).

The reason is that,

$$\mathfrak{D} \overset{(11)}{=} \tau\|\mathbf{p}\| + \sum_{i=1}^{k-1} \epsilon\|\mathbf{p}^{[i]}\| + \tau'\|\mathbf{p}^{[k]}\| = \tau\|\mathbf{p}'^{[k]}\| + \sum_{i=1}^{k-1} \epsilon\|\mathbf{p}'^{[i]}\| + \tau'\|\mathbf{p}'\|, \quad \text{by (18), (13) \& (12)}$$

$$\overset{(7)}{=} \tau\|\mathbf{p}'^{[k]}\| + d^k(\mathbf{q}', \mathbf{p}', \tau') < d^{k+1}(\mathbf{q}', \mathbf{p}', \tau'), \qquad \text{since } \tau < \epsilon.$$

As such, according to (9), the final position τ''-step is carried out for the duration $\tau'' := \frac{\tau\|\mathbf{p}'^{[k]}\|}{\|\mathbf{p}'^{[k]}\|} = \tau$, Therefore, by (10) and (18), it ends in $\mathbf{q}'' := \mathbf{q}'^{[k]} + \tau''\mathbf{p}'^{[k]} = \mathbf{q}^{[1]} - \tau\mathbf{p} = \mathbf{q}$. Finally by negating the final momentum, $(\mathbf{q}'' := \mathbf{q}, \mathbf{p}'' := \mathbf{p}, \tau'' := \tau)$ is returned. □

```
Algorithm 1: FDHMC-GIVEN-STEPSIZE-AND-DISTANCE $\left(\mathbf{q}^{(c)}, U, \epsilon, \mathfrak{D}\right)$
Input: $\mathbf{q}^{(c)}$, current state $\in \mathcal{X} \subset \mathbb{R}^n$;
       $U : \mathcal{X} \to \mathbb{R}$, Potential energy function (neg log of the target density, $\pi_{\mathbf{Q}}$);
       $\epsilon \in \mathbb{R}$, leapfrog step-size;
       $\mathfrak{D} \in \mathbb{R}$, total evolution distance.

▷ The following line is equivalent to: $\mathbf{p}^{(c)} \sim \pi_{\mathbf{P}}(\mathbf{p}) \propto \|\mathbf{p}\| \exp(-\frac{\|\mathbf{p}\|^2}{2})$
$\mathbf{s} \sim \mathcal{N}(\mathbf{0}_n, \mathbf{I}_{n \times n})$; $m \sim \chi(m; n+1)$; $\mathbf{p}^{(c)} \leftarrow m \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}$    ▷ where $\chi(m; n+1) \propto m^n \exp(-m^2/2)$
$\tau^{(c)} \sim \text{Unif}(0, \epsilon)$    ▷ duration of the initial fixed-momentum evolution
$\mathbf{q} \leftarrow \mathbf{q}^{(c)} + \tau^{(c)} \cdot \mathbf{p}^{(c)}$    ▷ initial evolution of the position vector for time $\tau^{(c)}$
$\mathfrak{d} = \mathfrak{D} - \tau^{(c)} \cdot \|\mathbf{p}^{(c)}\|$    ▷ This is the remaining distance
$\mathbf{p} \leftarrow \mathbf{p}^{(c)} - \epsilon \nabla U(\mathbf{q})$    ▷ initial evolution of the momentum vector for time $\epsilon$

▷ while a position full-step evolution is possible without exceeding the remaining distance, $\mathfrak{d}$:
while $\epsilon \cdot \|\mathbf{p}\| < \mathfrak{d}$ do
    $\mathbf{q} \leftarrow \mathbf{q} + \epsilon \cdot \mathbf{p}$    ▷ position full-step
    $\mathfrak{d} \leftarrow \mathfrak{d} - \epsilon \cdot \|\mathbf{p}\|$    ▷ update the remaining distance
    $\mathbf{p} \leftarrow \mathbf{p} - \epsilon \nabla U(\mathbf{q})$    ▷ momentum full step

$\tau^{(p)} \leftarrow \frac{\mathfrak{d}}{\|\mathbf{p}\|}$;   $\mathbf{q}^{(p)} \leftarrow \mathbf{q} + \tau^{(p)} \cdot \mathbf{p}$;   $\mathbf{p}^{(p)} \leftarrow -\mathbf{p}$;    ▷ constructing the final proposal
$\alpha \leftarrow \min\left\{1, \exp\left(U(\mathbf{q}^{(c)}) + \|\mathbf{p}^{(c)}\|^2/2 - U(\mathbf{q}^{(p)}) - \|\mathbf{p}^{(p)}\|^2/2\right)\right\}$    ▷ acceptance probability
if $u \sim \text{Unif}(0,1) < \alpha$ then return $\left(\mathbf{q}^{(p)}, \alpha\right)$ else return $\left(\mathbf{q}^{(c)}, \alpha\right)$
```

3 Automated Tuning of the Fixed-Distance Hamiltonian Monte Carlo (FDHMC)

In this section we present algorithms to automatically tune the hyper-parameters of the FDHMC algorithm, i.e. the leapfrog step-size, ϵ, and the fixed evolution distance, $\mathfrak{D}$. The same automated tuning is used in all the experiments that are presented in the main text.

Basic FDHMC. Algorithm 1, is the same FDHMC algorithm that is presented in the main text, except that here,

- The acceptance probability is computed in terms of the *Hamiltonian*s,

$$\begin{aligned}\alpha &= \min\{1, \exp(H^{(c)} - H^{(p)})\} \\ &= \min\left\{1, \exp\left(U(\mathbf{q}^{(c)}) + \frac{\|\mathbf{p}^{(c)}\|^2}{2} - U(\mathbf{q}^{(p)}) - \frac{\|\mathbf{p}^{(p)}\|^2}{2}\right)\right\},\end{aligned}$$

  rather than the direct form. This is recommended since the logarithmic computations lead to more numerical stability.
- The hyper-parameters, ϵ and $\mathfrak{D}$, are passed as (extra) inputs and the acceptance probability, α, is returned as the second output. These minor modifications are required so that we can use this algorithm as a module inside Algorithms 2 and 4 to tune the hyper-parameters.

Tuning the leapfrog step-size. Algorithm 2 tunes the FDHMC leapfrog step-size, ϵ, by dual averaging (Nesterov, 2009; Hoffman and Gelman, 2014). The input values of this Algorithm are: (a) an initial MCMC state, $\mathbf{q}^{(0)}$, (b) the target potential energy function, (c) an initial leapfrog step-size, ϵ_0, and (d) the fixed evolution distance, $\mathfrak{D}$. In this Algorithm, we set the hyper-parameter $\mu = \log(\epsilon_0)$ and the other hyper-parameters, i.e. $\log \overline{\epsilon_0}$, $\overline{H}_0$, γ and κ are set to the values proposed by Hoffman and Gelman (2014). We have not tried other configurations since the performance of the dual-averaging algorithm with the above choice of hyper-parameters already seemed satisfactory on all our experimental models.

Finding an initial step-size. Algorithm 3 shows the heuristic by which we choose the initial step-size, ϵ_0: We firstly choose a step-size $\epsilon^* = 1$ and then repeatedly double or halve the value of ϵ^*

```
Algorithm 2: TUNE-FDHMC-LEAPFROGSTEP $(\mathbf{q}^{(c)}, U, \epsilon, \mathfrak{D})$
Input: $\mathbf{q}^{(0)} \in \mathbb{R}^n$, initial state;
       $U : \mathbb{R}^n \to \mathbb{R}$, Potential energy function ($-$log of the target density, $\pi_{\mathbf{Q}}$);
       $\epsilon_0 \in \mathbb{R}$, initial leapfrog step-size;
       $\mathfrak{D} \in \mathbb{R}$, total evolution distance.
Hyper-parameters: $M^{\text{adapt}} = 200$, no. samples based on which the leapfrog step-size is tuned;
                  $\delta = 0.65$, target mean proposal acceptance probability;
                  $\mu = \log(\epsilon_0)$; $\log \bar{\epsilon} = 0$; $\bar{H} = 0$; $\gamma = 0.05$; $t_0 = 10$; $\kappa = 0.75$.

$\epsilon \leftarrow \epsilon_0$; $\mathbf{q} \leftarrow \mathbf{q}^{(0)}$
for $m = 1$ to $M^{\text{adapt}}$ do
    $(\mathbf{q}, \alpha) \leftarrow$ FDHMC-GIVEN-STEPSIZE-AND-DISTANCE $(\mathbf{q}, U, \epsilon, \mathfrak{D})$
    $\bar{H} \leftarrow \left(1 - \frac{1}{m+t_0}\right) \bar{H} + \frac{1}{m+t_0}(\delta - \alpha)$
    $\epsilon \leftarrow \exp\left(\mu - \frac{\sqrt{m}}{\gamma} \bar{H}\right)$
    $\log \bar{\epsilon} \leftarrow m^{-\kappa} \log(\epsilon) + (1 - m^{-\kappa}) \log \bar{\epsilon}$
return $\exp(\log \bar{\epsilon})$
```

```
Algorithm 3: LARGEEPSILON $(\mathbf{q}, U, \bar{m})$
Input: $\mathbf{q} \in \mathbb{R}^n$, a state;
       $U : \mathbb{R}^n \to \mathbb{R}$, Potential energy function ($-$log of the target density, $\pi_{\mathbf{Q}}$);
       $\bar{m}$, momentum magnitude.

$\mathbf{s} \sim \mathcal{N}(\mathbf{0}_n, \mathbf{I}_{n \times n})$; $\mathbf{p} \leftarrow \bar{m} \cdot \frac{\mathbf{s}}{\|\mathbf{s}\|}$
$\epsilon \leftarrow 1$
$(\mathbf{q}', \mathbf{p}') \leftarrow$ LEAPFROG$(\mathbf{q}, \mathbf{p}, \epsilon)$
if $\exp\left(U(\mathbf{q}) + \|\mathbf{p}\|^2/2 - U(\mathbf{q}') - \|\mathbf{p}'\|^2/2\right) > 0.5$ then $a \leftarrow 1$ else $a \leftarrow -1$
while $\left[\exp\left(U(\mathbf{q}) + \|\mathbf{p}\|^2/2 - U(\mathbf{q}') - \|\mathbf{p}'\|^2/2\right)\right]^a > 2^{-a}$ do
    $\epsilon \leftarrow 2^a \epsilon$
    $(\mathbf{q}', \mathbf{p}') \leftarrow$ LEAPFROG$(\mathbf{q}, \mathbf{p}, \epsilon)$
return $\epsilon$

LEAPFROG$(\mathbf{q}, \mathbf{p}, \epsilon)$ :
    $\mathbf{p}^* \leftarrow \mathbf{p} - \frac{\epsilon}{2} \nabla U(\mathbf{q})$; $\mathbf{q}' \leftarrow \mathbf{q} + \epsilon \mathbf{p}^*$; $\mathbf{p}' \leftarrow \mathbf{p}^* - \frac{\epsilon}{2} \nabla U(\mathbf{q}')$
    return $(\mathbf{q}', \mathbf{p}')$
```

till the acceptance probability of the Langevin proposal (i.e. a single HMC leapfrog step with random momentum) crosses 0.5.

This is similar to the heuristic suggested by Hoffman and Gelman (2014) except that we further stabilise the outcome by choosing a momentum which has an average magnitude. That is, $\bar{m} = \mathbb{E}_{m \sim \chi(n+1)}(m) = \sqrt{2}\Gamma(n/2+1)/\Gamma((n+1)/2)$.[1]

Choosing the evolution distance. In Algorithm 4, the fixed distance is also set automatically. This leads to an FDHMC sampler where no manual hyper-parameter tuning is required.

The proposed tuning method is based on the following heuristic:

---

[1] This fraction can be further simplified for even and odd values of n but this is not necessary because the mean value of a Chi distribution can simply be obtained from the existing libraries e.g. in Python:

$\bar{m} =$ `scipy.stats.chi.stats(df=n + 1, moments='m').`

```
Algorithm 4: AUTO-TUNED-FDHMC $\left(\mathbf{q}^{(0)}, U, N\right)$
Input: $\mathbf{q}^{(0)} \in \mathbb{R}^n$, initial state;
       $U : \mathbb{R}^n \to \mathbb{R}$, Potential energy function ($-$log of the target density, $\pi_{\mathbf{Q}}$);
       $N \in \mathbb{R}$, no. samples to be returned.
Hyper-parameters: $N^{\text{adapt}} = 500$, no. samples by which the fixed distance, $\mathfrak{D}$, is tuned;
                  $c = 10$, a hyper-parameter for choosing a sufficiently large distance.
▷ I. Tuning FDHMC's fixed distance and leapfrog step-size:
$\bar{m} \leftarrow \sqrt{2}\frac{\Gamma\left(\frac{n}{2}+1\right)}{\Gamma\left(\frac{n+1}{2}\right)}$    ▷ expected momentum magnitude: $\mathbb{E}_{m\sim\chi(n+1)}(m)$
$\epsilon^* \leftarrow$ LARGEEPSILON$(\mathbf{q}^{(0)}, U, \bar{m})$    ▷ sufficiently large but reasonable initial step-size
$\mathfrak{D}^* \leftarrow c \cdot \bar{m} \cdot \epsilon^*$    ▷ sufficiently large distance
▷ Tune the leapfrog step-size of an auxiliary FDHMC sampler associated with fixed distance $\mathfrak{D}^*$:
$\epsilon_1 \leftarrow$ TUNE-FDHMC-LEAPFROGSTEP$(\mathbf{q}^{(0)}, U, \epsilon^*, \mathfrak{D}^*)$
$d^{sum} \leftarrow 0$    ▷ to compute the total distance between $N^{\text{adapt}}$ successive samples
for $i = 1$ to $N^{\text{adapt}}$ do
    $(\mathbf{q}, -) \leftarrow$ FDHMC-GIVEN-STEPSIZE-AND-DISTANCE $\left(\mathbf{q}^{(0)}, U, \epsilon_1, \mathfrak{D}^*\right)$
    $d^{sum} \leftarrow d^{sum} + \|\mathbf{q} - \mathbf{q}^{(0)}\|$
    $\mathbf{q}^{(0)} \leftarrow \mathbf{q}$
$\mathfrak{D} \leftarrow \frac{d^{sum}}{N^{adapt}}$    ▷ final tuned fixed-distance
$\epsilon \leftarrow$ TUNE-FDHMC-LEAPFROGSTEP$(\mathbf{q}, U, \epsilon_1, \mathfrak{D})$    ▷ final tuned leapfrog step-size
▷ II. Drawing $N$ Samples:
for $i = 1$ to $N$ do
    $(\mathbf{q}^{(i)}, -) \leftarrow$ FDHMC-GIVEN-STEPSIZE-AND-DISTANCE $\left(\mathbf{q}^{(i-1)}, U, \epsilon, \mathfrak{D}\right)$
return $\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(N)}$
```

If $\mathfrak{D}$ is sufficiently large, its correlation with the expected distance of the successive samples, $\mathbb{E}[\|\mathbf{q}^{(i)} - \mathbf{q}^{(i-1)}\|]$, is low. In this case, the latter value, $\mathbb{E}[\|\mathbf{q}^{(i)} - \mathbf{q}^{(i-1)}\|]$, depends more on the geometry of the target density function, $\pi_{\mathbf{Q}}(\mathbf{q})$.

As such, in Algorithm 4

1. An auxiliary FDHMC sampler is build that is associated with a sufficiently large fixed-distance:
$$\mathfrak{D}^* = 10 \cdot \bar{m} \cdot \epsilon^*,$$
where $\bar{m}$ is the expected FDHMC momentum magnitude and ϵ^* is a large step-size that is determined by Algorithm 3.
2. $N^{\text{adapt}} = 500$ samples, $\mathbf{q}^{(i)}$, are drawn from the auxiliary sampler to approximate the expected distance between the successive samples. We set the tuned fixed-distance, $\mathfrak{D}$, to this value:
$$\mathfrak{D} \leftarrow \mathbb{E}[\|\mathbf{q}^{(i)} - \mathbf{q}^{(i-1)}\|] \approx \sum_{i=1}^{N^{\text{adapt}}} \|\mathbf{q}^{(i)} - \mathbf{q}^{(i-1)}\|/(N^{\text{adapt}}).$$
3. Having the final fixed-distance, $\mathfrak{D}$, we also tune the step-size, ϵ by Algorithm 2.

Comparison with static and dynamic variations of HMC. For tuning FDHMC, in total we take 900 samples (200 samples to tune the step-size of the auxiliary sampler by dual averaging, $N^{\text{adapt}} = 500$ samples to choose the final $\mathfrak{D}$ and another 200 samples to tune the final step-size). For tuning static HMC and the two variations of NUTS (i.e. Dyanamic Slice HMC (DSHMC) and and Dynamic Multinomial HMC (DMHMC)) by dual averaging, we take 200 samples. Note that the time and computation resources that are spent on tuning FDHMC is less than time/computation required to tune NUTS samplers because drawing an FDHMC sample is in average much faster.

On each model and for each algorithm we run 50 separate MCMC chains that after an initial 200 burn-in draws, return 1000 samples (50,000 in total).

Model	dim	Tuned fixed distance
MVN	10	2.57052 ± 0.12061
MVN	30	2.29134 ± 0.04483
MVN	100	2.22110 ± 0.00386
MVN	300	2.03197 ± 0.00308
FNNL	5	3.97946 ± 0.35447
FNNL	10	4.77171 ± 0.39750
FNNL	50	5.07557 ± 0.43943
FNNL	100	5.43784 ± 0.83692
AusCr	15	4.51966 ± 0.46677
SPECT	23	6.50499 ± 0.18321
GrCr	25	0.80260 ± 0.01084

Table 1: Expected value of the parameter $\mathfrak{D}$ and $\pm 95\%$ confidence interval that is tuned by running Algorithm 4 on 50 MCMC chains.

The approximation of the Effective sample size (ESS) is based on the details suggested by Hoffman and Gelman (2014). (a) For each dimension $i = 1, \ldots, n$, the univariate ESS that is associated with function $f = q_i$ is approximated and the minimum of these n values is returned. (b) The estimate of the auto-correlation spectrum is truncated when the auto-correlations first dip below 0.05. (c) To estimate the true means and variances of q_i, we rely on an extra chain of 500,000 reference samples. In the case of MVN and FNNL models, the reference samples are directly drawn from the model. In the case of the three Bayesian Logistic Regression models, direct sampling is not possible and the estimation is based on a separate chain of 500,000 FDHMC samples.

4 Quantitative Analysis

In this section we study the sensitivity of the performance of FDHMC to the fixed-distance parameter, $\mathfrak{D}$, as well as the effectiveness of the proposed heuristic algorithm in tuning this parameter.

4.1 ESS/gradient versus fixed-distance

We repeat the experiments of the main text but instead of automated tuning, we set $\mathfrak{D}$ to a value in $\{1, 2, 3, 4, 5, 6, 7, 8\}$. For each configuration we run multiple independent MCMC chains.[2]

For each chain, the step-size is tuned by dual averaging as before (using Algorithm 2 with $M^{\text{adapt}} = 200$ samples) and ESS per gradient is approximated based on 1000 samples. The resulting ESS per gradient versus the chosen fixed-distance is plotted in Figure 2. In these plots, each circle represents an MCMC chain. It can be seen that the sensitivity of performance to $\mathfrak{D}$ as well as the range of its optimal choices varies significantly and is model dependent.

4.2 Automated tuning of the fixed-distance

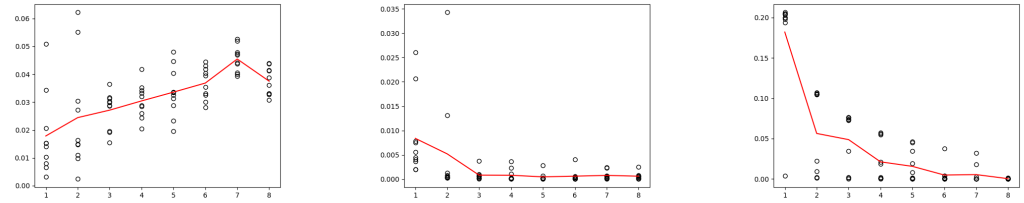
Table 1 depicts the expected chosen fixed-distance parameter (by running Algorithm 4 with $N^{\text{adapt}} = 500$ samples to tune $\mathfrak{D}$) for the experimental models of the main text $\pm 95\%$ confidence interval (based on 50 independent MCMC chains). Comparing the results of this table with the plots of Figure 2 suggests that the proposed automated parameter tuning performs reasonably well and in the majority of the experimental models, chooses parameter values that are not far from the optimal range.

5 Exploring multimodal distribution

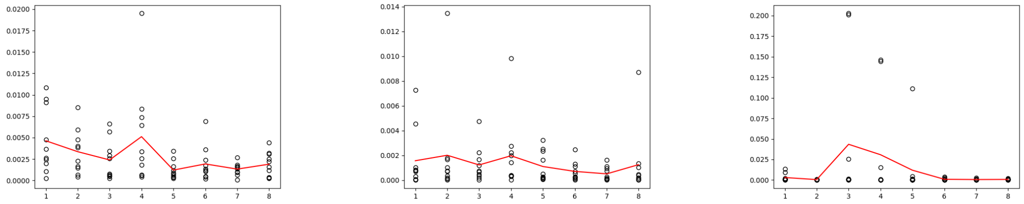
In this section we plot samples taken by FDHMC, Dyanamic Slice HMC (DSHMC) (Hoffman and Gelman, 2014) and Dynamic Multinomial HMC (DMHMC) (Betancourt, 2017) from a multimodal distribution.

---

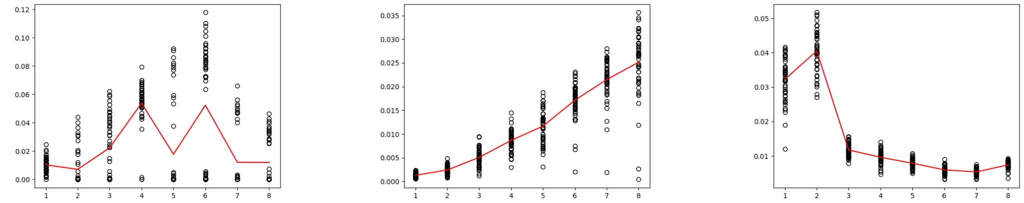
[2] We have run 10 MCMC chains for the configurations associated with MVN and FNNL models and 50 MCMC chains for the configurations associated with the three Bayesian Logistic Regression models.



(a) MVN (dim=10) (b) MVN (dim=30) (c) MVN (dim=100)



(d) FNNL (dim=10) (e) FNNL (dim=50) (f) FNNL (dim=100)



(g) AusCr (h) SPECT (i) GrCr

Figure 2: ESS/grad versus the fixed-distance, $\mathfrak{D}$, in FDHMC sampling.

FDHMC is automatically tuned by Algorithm 4 (with default hyper-parameters) while DSHMC and DMHMC are tuned by the default dual averaging step-size adaptor that is available in *Mici* probabilistic programming language (Graham, 2019).

The target is the following mixture of four Normal distributions (M4N):

$$\pi(x_1, \ldots, x_d) = \pi(\mathbf{x}) = \frac{1}{4} \sum_{i=1}^{4} \mathcal{N}(\mathbf{x}; \mathbf{m}_i, \sigma^2) \tag{19}$$

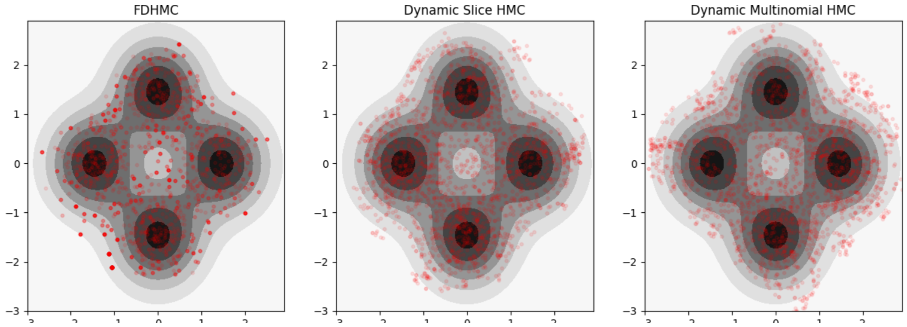
where the means of the modes are chosen to be

$$\mathbf{m_1} = [-1.5, 0, 0, \ldots, 0]^\top, \ \mathbf{m_2} = [+1.5, 0, 0, \ldots, 0]^\top,$$
$$\mathbf{m_3} = [0, -1.5, 0, \ldots, 0]^\top, \ \mathbf{m_4} = [0, +1.5, 0, \ldots, 0]^\top$$

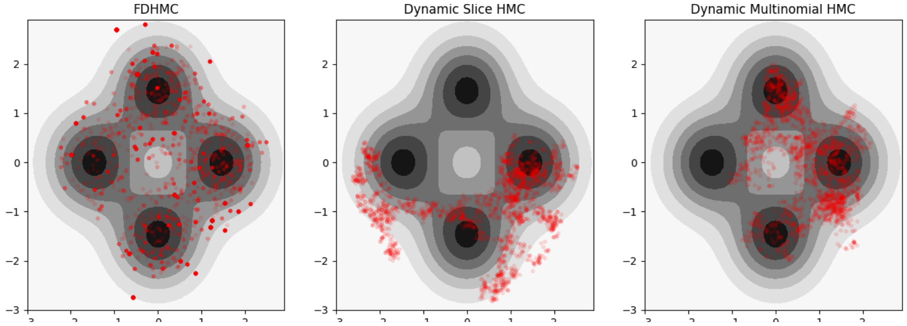
We set $\sigma^2 = 0.5$ and let the dimension $d \in \{2, 10, 20\}$.

2K samples are drawn from this model and the first two elements (x_1, x_2) of the samples are plotted in Figure 3. The conditional distribution, $\pi(x_1, x_2 | x_3 = 0, \ldots, x_d = 0)$, is plotted in the background to indicate the approximate location of the modes. It can be seen that as the dimension increases, the rejection rate of FDHMC proposals gradually increases and the samples look sparser. However, even in the high dimensional model, the transition between the four modes of the distribution is not rare.

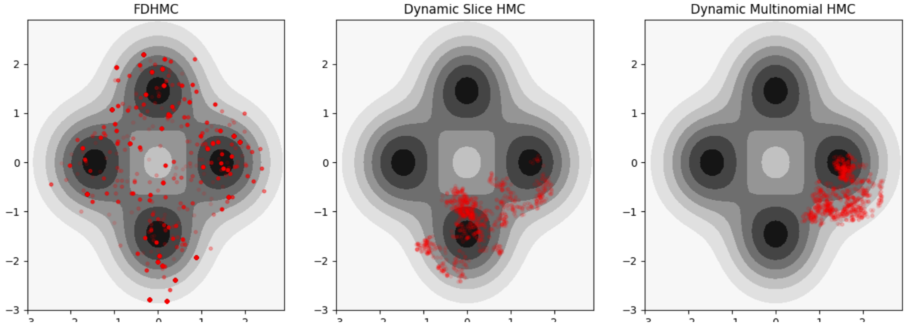
In comparison, in high dimensions, DSHMC and DMHMC has more difficulty in switching between the distribution modes. This observation matches our expectation, because the expected magnitude of FDHMC's momentum vectors is higher than the expected momentum magnitude of the existing variations of HMC. Therefore we expect that with large momentums, FDHMC's state passes through the potential energy barriers easier and therefore transits between the distribution modes more frequently.



(a) dim=2



(b) dim=10



(c) dim=20

Figure 3: 2K draws from a) 2D (b) 10D and (c) 20D symmetric mixture of 4 normal distributions (M4N) with automatically tuned FDHMC (left), Dynamic slice HMC (middle) and Dynamic Multinomial HMC (right).

References

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. *arXiv preprint arXiv:1701.02434*, 2017.

Matthew Graham. Graham: Mici: Markov chain Monte Carlo (MCMC) methods for Python. URL: `https://git.io/mici.py`, 2019.

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(1):1593–1623, 2014.

Radford M. Neal. MCMC using Hamiltonian dynamics. In Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng, editors, *Handbook of Markov Chain Monte Carlo*, pages 113–162. Chapman & Hall, 2011.

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