A. Involutive MCMC

A.1. Proof of Proposition 1 (FPE condition)

For the target distribution $p(x)$ and the deterministic proposal $q(x' \mid x) = \delta(x' - f(x))$, we consider the following transition kernel

$$t(x' \mid x) = \delta(x' - f(x)) \min\left\{1, \frac{p(x')}{p(x)}\left|\frac{\partial f}{\partial x}\right|\right\} + \delta(x' - x)\int dx''\, \delta(x'' - f(x))\left(1 - \min\left\{1, \frac{p(x'')}{p(x)}\left|\frac{\partial f(x)}{\partial x}\right|\right\}\right). \tag{33}$$

Then we want to check the fixed-point equation

$$\int dx\; t(x' \mid x)p(x) = p(x'). \tag{34}$$

Substitution of $t(x' \mid x)$ gives

$$\int dx\; \delta(x' - f(x)) \min\left\{p(x), p(x')\left|\frac{\partial f}{\partial x}\right|\right\} + p(x') - \min\left\{p(x'), p(f(x'))\left|\frac{\partial f}{\partial x'}\right|\right\} = p(x') \tag{35}$$

Assuming that $f(x)$ has the inverse $f^{-1}(x)$, we change variables $x = f^{-1}(\widetilde{x})$ and rewrite the previous equation as

$$\int d\widetilde{x}\; \delta(x' - \widetilde{x}) \min\left\{p(f^{-1}(\widetilde{x})), p(x')\left|\frac{\partial f}{\partial x}\right|_{x=f^{-1}(\widetilde{x})}\right\}\left|\frac{\partial f^{-1}}{\partial \widetilde{x}}\right| - \min\left\{p(x'), p(f(x'))\left|\frac{\partial f}{\partial x'}\right|\right\} = 0. \tag{36}$$

Using the chain rule, we have

$$1 = \left|\frac{\partial f(f^{-1}(x))}{\partial x}\right| = \left|\frac{\partial f}{\partial y}\right|_{y=f^{-1}(x)}\left|\frac{\partial f^{-1}}{\partial x}\right|. \tag{37}$$

Thus, we obtain the following condition to satisfy the fixed-point equation

$$\min\left\{p(f^{-1}(x))\left|\frac{\partial f^{-1}}{\partial x}\right|, p(x)\right\} = \min\left\{p(x), p(f(x))\left|\frac{\partial f}{\partial x}\right|\right\}. \tag{38}$$

The same applies for the joint space

$$\min\left\{p(f^{-1}(x,v))\left|\frac{\partial f^{-1}(x,v)}{\partial [x,v]}\right|, p(x,v)\right\} = \min\left\{p(x,v), p(f(x,v))\left|\frac{\partial f(x,v)}{\partial [x,v]}\right|\right\}. \tag{39}$$

Moreover, there is no need to care about the distribution of v' in the fixed point equation

$$\int dx dv dv'\; t(x', v' \mid x, v)p(x,v) = p(x'). \tag{40}$$

Thus, we obtain more general condition

$$\int dv \min\left\{p(f^{-1}(x,v))\left|\frac{\partial f^{-1}(x,v)}{\partial [x,v]}\right|, p(x,v)\right\} = \int dv \min\left\{p(x,v), p(f(x,v))\left|\frac{\partial f(x,v)}{\partial [x,v]}\right|\right\}. \tag{41}$$

Also, note that the condition can be easily rewritten for different acceptance function, e.g., for the Barker's test (Barker, 1965). That is,

$$t(x' \mid x) = \delta(x' - f(x))\left[1 + \frac{p(x)}{p(x')}\left|\frac{\partial f}{\partial x}\right|^{-1}\right]^{-1} + \delta(x' - x)\left(1 - \left[1 + \frac{p(x)}{p(f(x))}\left|\frac{\partial f}{\partial x}\right|^{-1}\right]^{-1}\right). \tag{42}$$

Substituting this kernel into the fixed point equation $\int dx t(x' \mid x)p(x) = p(x')$, and performing a similar algebra, we have

$$\left[\frac{1}{p(x)} + \frac{1}{p(f^{-1}(x))}\left|\frac{\partial f^{-1}}{\partial x}\right|^{-1}\right]^{-1} = \left[\frac{1}{p(x)} + \frac{1}{p(f(x))}\left|\frac{\partial f}{\partial x}\right|^{-1}\right]^{-1} \tag{43}$$

Thus, for the Barker's test, the fixed point equation can be reduced to

$$p(f^{-1}(x))\left|\frac{\partial f^{-1}}{\partial x}\right| = p(f(x))\left|\frac{\partial f}{\partial x}\right| \tag{44}$$

A.2. Proof of Proposition 2 (Detailed balance)

We analyse this property of Involutive MCMC by deriving the reverse operator $r(x, v \mid x', v')$, which is defined as

$$t(x', v' \mid x, v)p(x, v) = r(x, v \mid x', v')p(x', v'). \tag{45}$$

By the definition, we have

$$r(x, v \mid x', v') = t(x', v' \mid x, v)\frac{p(x, v)}{p(x', v')} \tag{46}$$

$$r(x, v \mid x', v') = \delta([x', v'] - f(x, v)) \min\left\{\frac{p(x, v)}{p(x', v')}, \left|\frac{\partial f(x, v)}{\partial [x, v]}\right|\right\} + \tag{47}$$

$$+ \delta([x', v'] - [x, v])\left(\frac{p(x, v)}{p(x', v')} - \min\left\{\frac{p(x, v)}{p(x', v')}, \frac{p(f(x, v))}{p(x', v')}\left|\frac{\partial f(x, v)}{\partial [x, v]}\right|\right\}\right) \tag{48}$$

The detailed balance is satisfied in the joint space if $\int_A r(x, v \mid x', v')dxdv = \int_A t(x, v \mid x', v')dxdv$, where A is any non-zero measure volume in the joint space. Remind that

$$t(x, v \mid x', v') = \delta([x, v] - f(x', v')) \min\left\{1, \frac{p(x, v)}{p(x', v')}\left|\frac{\partial f(x', v')}{\partial [x', v']}\right|\right\} + \tag{49}$$

$$+ \delta([x, v] - [x', v'])\left(1 - \min\left\{1, \frac{p(f(x', v'))}{p(x', v')}\left|\frac{\partial f(x', v')}{\partial [x', v']}\right|\right\}\right). \tag{50}$$

For the involutive map f, it is clear that the integrals $\int_A r(x, v \mid x', v')dxdv$ and $\int_A t(x, v \mid x', v')dxdv$ are non-zero around the points $[x, v] = [x', v']$ and $[x, v] = f(x', v')$. Thus, integrating over A_1 that is around $[x, v] = [x', v']$, we have

$$\int_{A_1} r(x, v \mid x', v')dxdv = 1 - \min\left\{1, \frac{p(f(x', v'))}{p(x', v')}\left|\frac{\partial f(x', v')}{\partial [x', v']}\right|\right\} = \int_{A_1} t(x, v \mid x', v')dxdv. \tag{51}$$

Then, integrating over A_2 that is around $[x, v] = f(x', v')$, we have

$$\int_{A_2} r(x, v \mid x', v')dxdv = \int_{f(A_2)} dxdv\ \delta([x', v'] - [x, v]) \cdot \min\left\{\frac{p(f^{-1}(x, v))}{p(x', v')}, \left|\frac{\partial f(y)}{\partial y}\right|_{y=f^{-1}(x,v)}\right\}\left|\frac{\partial f^{-1}(x, v)}{\partial [x, v]}\right| = \tag{52}$$

$$= \int_{f(A_2)} dxdv\ \delta([x', v'] - [x, v]) \cdot \min\left\{\frac{p(f^{-1}(x, v))}{p(x', v')}\left|\frac{\partial f^{-1}(x, v)}{\partial [x, v]}\right|, 1\right\} \tag{53}$$

Since f is an involutive map, then $f^{-1} = f$, and $[x', v']$ lies in $f(A_2)$, where A_2 is an area around $[x, v] = f(x', v')$. Thus, we have

$$\int_{A_2} t(x, v \mid x', v')dxdv = \min\left\{1, \frac{p(f(x', v'))}{p(x', v')}\left|\frac{\partial f(x', v')}{\partial [x', v']}\right|\right\} = \int_{A_2} r(x, v \mid x', v')dxdv \tag{54}$$

Hence, $t(x', v' \mid x, v)$ satisfies the detailed balance in the joint space. Moreover, that yields the detailed balance on the support of $p(x)$. Indeed, reducing to the samples from $p(x)$, we have the transition kernel

$$\widehat{t}(x \mid x') = \int t(x, v \mid x', v')p(v' \mid x')dv'dv. \tag{55}$$

By definition, the reverse transition kernel is

$$\widehat{r}(x' \mid x) = \widehat{t}(x \mid x')\frac{p(x')}{p(x)} = \frac{p(x')}{p(x)}\int t(x, v \mid x', v')p(v' \mid x')dv'dv. \tag{56}$$

Since $t(x, v \mid x', v')$ satisfies the detailed balance, we have

$$\widehat{r}(x' \mid x) = \frac{p(x')}{p(x)}\int t(x', v' \mid x, v)p(v' \mid x')\frac{p(x, v)}{p(x', v')}dv'dv = \int t(x', v' \mid x, v)p(v \mid x)dv'dv = \widehat{t}(x' \mid x) \tag{57}$$

Hence, $\widehat{t}(x \mid x')$ also satisfies the detailed balance.

A.3. (Murray & Elliott, 2012; Neal, 2012)

Here we formulate the algorithm from the papers (Murray & Elliott, 2012; Neal, 2012). We consider one-dimensional target density $p(x)$ and some transition kernel $q(x' \,|\, x)$ that satisfy the fixed point equation with the target density. For any kernel $q(x' \,|\, x)$ we can define the reverse transition kernel $r(x \,|\, x')$ in terms of so-called generalized detailed balance:

$$r(x \,|\, x')p(x') = q(x' \,|\, x)p(x). \tag{58}$$

Note that the reverse kernel is a correct distribution w.r.t. x, and also satisfy the fixed point equation:

$$\int dx\; r(x \,|\, x') = \frac{1}{p(x')} \int dx\; q(x' \,|\, x)p(x) = 1, \quad \int dx'\; r(x \,|\, x')p(x') = \int dx'\; q(x' \,|\, x)p(x) = p(x). \tag{59}$$

Consider the joint distribution $p(x, u) = p(x)p(u)$, where $p(u) = \text{Uniform}[0, 1]$. For now, assume that at each iteration u is sampled independently from the uniform distribution and the transition kernel is the deterministic function $f(x, v) = [x', v']$ defined as:

$$x' = F^{-1}_{q(\cdot \,|\, x)}(u), \quad u' = F_{r(\cdot \,|\, x')}(x), \tag{60}$$

where F_p is a CDF of a distribution with the density p. To check the measure-preserving condition (1), we need to derive the determinant of the Jacobian of the f. Using the chain rule, we have

$$\frac{\partial u'}{\partial u} = \frac{\partial u'}{\partial x'}\frac{\partial x'}{\partial u}, \quad \frac{\partial u'}{\partial x} = r(x \,|\, x') + \frac{\partial u'}{\partial x'}\frac{\partial x'}{\partial x}. \tag{61}$$

Then the Jacobian is

$$|J| = \left| \frac{\partial x'}{\partial x}\frac{\partial u'}{\partial u} - \frac{\partial x'}{\partial u}\frac{\partial u'}{\partial x} \right| = \frac{\partial x'}{\partial u}\left| \frac{\partial x'}{\partial x}\frac{\partial u'}{\partial x'} - \frac{\partial u'}{\partial x} \right| = \frac{r(x \,|\, x')}{q(x' \,|\, x)}. \tag{62}$$

Now, it is easy to check the measure preserving condition (3) using the definition of the reverse transition kernel.

$$p(f(x, u))\left| \frac{\partial f(x, u)}{\partial [x, u]} \right| = p(x')p(u')\frac{r(x \,|\, x')}{q(x' \,|\, x)} = p(x) = p(x, u). \tag{63}$$

In the paper (Murray & Elliott, 2012), the authors propose to use some dependent random stream d_t to update the auxiliary variable u as $u_t = (u_{t-1} + d_t) \mod 1$, instead of sampling from the uniform. In some cases, it is even possibly to eliminate all the stochasticity by letting d_t be some constant irrational number: $d_t = c$.

A.4. Proof of Trick 2 (Mixture of involutions)

We remind that in the trick we consider the joint distribution $p(x, v, a) = p(x, v)p(a \,|\, x, v)$, and the family of involutions $f_a(x, v)$, i.e. $f_a(f_a(x, v)) = [x, v]$. To make the calculations more concise, we denote the tuple $[x, v]$ as y. Then the transition kernel for the distribution $p(y, a) = p(x, v, a)$ is

$$\begin{aligned} t(y', a' \,|\, y, a) =& \delta([y', a'] - [f_a(y), a]) \min\left\{1, \frac{p(f_a(y))p(a \,|\, f_a(y))}{p(y)p(a \,|\, y)}\left|\frac{\partial f_a(y)}{\partial y}\right|\right\} + \\ &+ \delta([y', a'] - [y, a])\left(1 - \min\left\{1, \frac{p(f_a(y))p(a \,|\, f_a(y))}{p(y)p(a \,|\, y)}\left|\frac{\partial f_a(y)}{\partial y}\right|\right\}\right). \end{aligned} \tag{64}$$

Putting this transition kernel into the fixed point equation ($\int t(y', a' \,|\, y, a)p(y, a)dyda = p(y', a')$), we have

$$\begin{aligned} &\int dyda\; \delta([y', a'] - [f_a(y), a]) \min\left\{p(y, a), p(f_a(y))p(a \,|\, f_a(y))\left|\frac{\partial f_a(y)}{\partial y}\right|\right\} + \\ &+ p(y', a') - \int dyda\; \delta([y', a'] - [y, a]) \min\left\{p(y, a), p(f_a(y))p(a \,|\, f_a(y))\left|\frac{\partial f_a(y)}{\partial y}\right|\right\} = p(y', a'). \end{aligned} \tag{65}$$

From the last equation, we immediately obtain the equation

$$\min\left\{p(f_{a'}^{-1}(y'),a')\left|\frac{\partial f_{a'}^{-1}(y')}{\partial y'}\right|,p(y',a')\right\}=\min\left\{p(y',a'),p(f_{a'}(y'),a')\left|\frac{\partial f_{a'}(y')}{\partial y'}\right|\right\},\tag{66}$$

which solutions in the space of f_a include all involutive functions: $f_a(y)=f_a^{-1}(y)$.

To demonstrate that we must not change the variable a let's try to apply some smooth function g to propose a new a. Then equation (65) becomes

$$\begin{aligned}&\int dyda\;\delta([y',a']-[f_a(y),g(a)])\min\left\{p(y,a),p(f_a(y),g(a))\left|\frac{\partial f_a(y)}{\partial y}\right|\left|\frac{\partial g(a)}{\partial a}\right|\right\}+\\&+p(y',a')-\int dyda\;\delta([y',a']-[y,a])\min\left\{p(y,a),p(f_a(y),g(a))\left|\frac{\partial f_a(y)}{\partial y}\right|\left|\frac{\partial g(a)}{\partial a}\right|\right\}=p(y',a'),\end{aligned}\tag{67}$$

which yields the much stronger condition:

$$\begin{aligned}&\int da\;\delta(a'-g(a))\min\left\{p(f_a^{-1}(y'),a)\left|\frac{\partial f_a^{-1}(y')}{\partial y'}\right|,p(y',g(a))\left|\frac{\partial g(a)}{\partial a}\right|\right\}=\\&=\min\left\{p(y',a'),p(f_{a'}(y'),g(a'))\left|\frac{\partial f_{a'}(y')}{\partial y'}\right|\left|\frac{\partial g(a')}{\partial a'}\right|\right\}\end{aligned}\tag{68}$$

$$\begin{aligned}&\min\left\{p(f_{g^{-1}(a')}^{-1}(y'),g^{-1}(a'))\left|\frac{\partial f_{g^{-1}(a')}^{-1}(y')}{\partial y'}\right|\left|\frac{\partial g^{-1}(a')}{\partial a'}\right|,p(y',a')\right\}=\\&=\min\left\{p(y',a'),p(f_{a'}(y'),g(a'))\left|\frac{\partial f_{a'}(y')}{\partial y'}\right|\left|\frac{\partial g(a')}{\partial a'}\right|\right\}\end{aligned}\tag{69}$$

Looking for some solutions of this equation, we see that the involutivity of g ($g(a)=g^{-1}(a)$) is not enough anymore. Now, we also need $f_{g^{-1}(a)}^{-1}(y)=f_a(y)$. By the assumption, f_a is an involution; hence, we must guarantee $f_{g^{-1}(a)}(y)=f_a(y)$. Thus, we end up with $g^{-1}(a)=g(a)=a$, what forces g to be the identity mapping. Actually, we can guarantee $f_{g^{-1}(a)}^{-1}(y)=f_a(y)$ with non-trivial g if f is not an involution. We describe the latter in Trick 3.

The detailed balance for kernel (64) follows directly from Proposition 2, as well as the detailed balance for the collapsed kernel to the support of $p(y)$. To bring more intuition here, one can consider the simple case of independent a: $p(a\,|\,y)=p(a)$, then the kernel $t(y'\,|\,y)$ can be considered as a linear mixture, where each kernel is reversible:

$$\begin{aligned}t(y'\,|\,y)=\int da\;p(a)\Bigg[&\delta(y'-f_a(y))\min\left\{1,\frac{p(f_a(y))}{p(y)}\left|\frac{\partial f_a(y)}{\partial y}\right|\right\}+\\&+\delta(y'-y)\left(1-\min\left\{1,\frac{p(f_a(y))}{p(y)}\left|\frac{\partial f_a(y)}{\partial y}\right|\right\}\right)\Bigg].\end{aligned}\tag{70}$$

The general case $p(y,a)=p(a\,|\,y)p(y)$ is called state-depended mixture by (Geyer, 2003).

B. Special cases of Involutive MCMC

B.1. Metropolis-Hastings algorithm

Algorithm 2 The Metropolis-Hastings algorithm

input density of target distribution $\hat{p}(x) \propto p(x)$
input proposal distribution $q(x' \mid x)$
  initialize x
  for $i = 0 \dots n$ **do**
    sample proposal point $x' \sim q(x' \mid x)$
    $P = \min\{1, \frac{\hat{p}(x')q(x \mid x')}{\hat{p}(x)q(x' \mid x)}\}$
    $x_i = \begin{cases} x', \text{ with probability } P \\ x, \text{ with probability } (1-P) \end{cases}$
    $x \leftarrow x_i$
  end for
output $\{x_0, \dots, x_n\}$

To see that the MH algorithm is an instance of iMCMC, let's define the joint distribution as $p(x, v) = q(v \mid x)p(x)$ and the deterministic map as $f(x, v) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}$ (note that it is an involution). For that case, we can write iMCMC transition kernel as

$$t(x', v' \mid x, v) = \delta([x', v'] - [v, x]) \min\left\{1, \frac{p(x', v')}{p(x, v)}\right\} + \delta([x', v'] - [x, v])\left(1 - \min\left\{1, \frac{p(v, x)}{p(x, v)}\right\}\right) \tag{71}$$

Then we substitute the last equation into the reduced transition kernel

$$\widehat{t}(x' \mid x) = \int dv dv' t(x', v' \mid x, v) q(v \mid x) \tag{72}$$

$$t(x' \mid x, v) = \int dv'\, t(x', v' \mid x, v) = \delta(x' - v) \min\left\{1, \frac{p(x', x)}{p(x, v)}\right\} + \delta(x' - x)\left(1 - \min\left\{1, \frac{p(v, x)}{p(x, v)}\right\}\right) \tag{73}$$

$$\widehat{t}(x' \mid x) = \int dv\, t(x' \mid x, v) q(v \mid x) = q(x' \mid x) \min\left\{1, \frac{p(x')q(x \mid x')}{p(x)q(x' \mid x)}\right\} + \tag{74}$$

$$+ \delta(x' - x) \int dv\, q(v \mid x)\left(1 - \min\left\{1, \frac{p(v)q(x \mid v)}{p(x)q(v \mid x)}\right\}\right) = q_{\text{MH}}(x' \mid x) \tag{75}$$

The last equation is the kernel of the conventional Metropolis-Hastings algorithm with proposal $q(x' \mid x)$.

Note that the following special cases can be obtained by the same involution $f(x, v) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}$ and different auxiliary distributions:

- the Random-Walk Metropolis (Metropolis et al., 1953) (auxiliary $q(v \mid x) = q(x \mid v)$)
- Metropolis-adjusted Langevin dynamics (Besag, 1994; Roberts & Rosenthal, 1998).
- Any kernel $q(v \mid x)$ that satisfy the detailed balance ($q(v \mid x)p(x) = q(x \mid v)p(v)$)
- Any independent sampler $p(x)$ (auxiliary $p(v)$).

B.2. Mixture Proposal MCMC

Algorithm 3 Mixture Proposal MCMC

input density of target distribution $p(x)$
input mixture proposal distribution $\int q_r(x' \,|\, a) q_f(a \,|\, x) da$
  initialize x
  for $i = 0 \dots n$ **do**
    sample $a \sim q_f(a \,|\, x)$
    sample $x' \sim q_r(x' \,|\, a)$
    $P = \min \left\{ 1, \frac{p(x') q_r(x \,|\, a) q_f(a \,|\, x')}{p(x) q_r(x' \,|\, a) q_f(a \,|\, x)} \right\}$
    $x_i = \begin{cases} x', \text{ with probability } P \\ x, \text{ with probability } (1 - P) \end{cases}$
    $x \leftarrow x_i$
  end for
output $\{x_0, \dots, x_n\}$

We formulate the algorithm from the paper (Habib & Barber, 2018) in Algorithm 3. To demonstrate that the iMCMC formalism includes this algorithm, we take the joint distribution of target variable x and auxiliary variables a, v as $p(x, a, v) = p(x) q_r(v \,|\, a) q_f(a \,|\, x)$. The deterministic involution is $f(x, a, v) = [v, a, x]$. Then the transition kernel in the joint space is

$$t(x', a', v' \,|\, x, a, v) = \delta([x', a', v'] - [v, a, x]) \min \left\{ 1, \frac{p(x') q_r(v' \,|\, a') q_f(a' \,|\, x')}{p(x) q_r(v \,|\, a) q_f(a \,|\, x)} \right\} + \tag{76}$$

$$+ \delta([x', a', v'] - [x, a, v]) \left(1 - \min \left\{ 1, \frac{p(v) q_r(x \,|\, a) q_f(a \,|\, v)}{p(x) q_r(v \,|\, a) q_f(a \,|\, x)} \right\} \right). \tag{77}$$

This transitional kernel is equivalent to the Algorithm 3. Indeed, the probability to accept the proposed state v is the same as the acceptance probability in Algorithm 3 and the state v goes from the same proposal $\int da \; q_r(v \,|\, a) q(a \,|\, x)$.

To make the equivalence more apparent we derive formula (17) from (Habib & Barber, 2018) by integrating the transition kernel $t(x', a', v' \,|\, x, a, v)$ over the corresponding coordinates. That is

$$\widehat{t}(x', a' \,|\, x) = \int da dv' dv \; t(x', a', v' \,|\, x, a, v) p(a, v \,|\, x) = \tag{78}$$

$$= q_r(x' \,|\, a') q_f(a' \,|\, x) \min \left\{ 1, \frac{p(x') q_r(x \,|\, a') q_f(a' \,|\, x')}{p(x) q_r(x' \,|\, a') q_f(a' \,|\, x)} \right\} + \tag{79}$$

$$+ \delta(x' - x) q_f(a' \,|\, x) \left(1 - \int dv \; q_r(v \,|\, a') \min \left\{ 1, \frac{p(v) q_r(x \,|\, a') q_f(a' \,|\, v)}{p(x) q_r(v \,|\, a') q_f(a' \,|\, x)} \right\} \right). \tag{80}$$

Note that if we further marginalize the kernel $\widehat{t}(x', a' \,|\, x)$ over a' we obtain the kernel

$$\widehat{t}(x' \,|\, x) = \int da' \; q_r(x' \,|\, a') q_f(a' \,|\, x) \min \left\{ 1, \frac{p(x') q_r(x \,|\, a') q_f(a' \,|\, x')}{p(x) q_r(x' \,|\, a') q_f(a' \,|\, x)} \right\} + \tag{81}$$

$$+ \delta(x' - x) \left(1 - \int dv da' \; q_r(v \,|\, a') q_f(a' \,|\, x) \min \left\{ 1, \frac{p(v) q_r(x \,|\, a') q_f(a' \,|\, v)}{p(x) q_r(v \,|\, a') q_f(a' \,|\, x)} \right\} \right), \tag{82}$$

which is not equivalent to the Metropolis-Hastings kernel with the proposal

$$\widetilde{q}(v \,|\, x) = \int da \; q_r(v \,|\, a) q(a \,|\, x). \tag{83}$$

B.3. Multiple-Try Metropolis

Algorithm 4 Multiple-Try Metropolis

input target density $p(x)$, proposal $q(y \mid x)$, nonnegative symmetric function $\lambda(x, y) = \lambda(y, x)$
input denote weight function $w(x, y) = p(x)q(y \mid x)\lambda(x, y)$
  initialize x
  for $i = 0 \dots n$ **do**
    sample $y_1, \dots, y_k \sim q(y_j \mid x)$
    evaluate weights $w_j = p(y_j)q(x \mid y_j)\lambda(y_j, x), \quad j = 1, \dots, k$
    set $y = y_j$ with probability $w_j / (\sum_j w_j)$
    sample $x_1^*, \dots, x_{k-1}^* \sim q(x_j \mid y)$
    set $x_k^* = x$
    $P = \min\left\{1, \frac{w(y_1,x)+\ldots+w(y_k,x)}{w(x_1^*,y)+\ldots+w(x_k^*,y)}\right\}$
    $x_i = \begin{cases} y, \text{ with probability } P \\ x, \text{ with probability } (1-P) \end{cases}$
    $x \leftarrow x_i$
  end for
output samples $\{x_0, \dots, x_n\}$

We begin the proof with the recall of the Multiple-Try Metropolis (MTM) algorithm (Algorithm 4). To write MTM as Involutive MCMC, we consider the joint distribution and the family of involutions as follows.

$$p(x, y_1, \dots, y_k, x_1^*, \dots, x_{k-1}^*, j) = p(x) \prod_{i=1}^{k} q(y_i \mid x) p(j \mid y_1, \dots, y_k, x) \prod_{i=1}^{k-1} q(x_i^* \mid y_j), \tag{84}$$

$$p(j \mid y_1, \dots, y_k, x) = \frac{w(y_j, x)}{\sum_j w(y_j, x)}, \quad w(x, y) = p(x)q(y \mid x)\lambda(x, y), \quad j = 1, \dots, k \tag{85}$$

$$f_j(x, y_1, \dots, y_k, x_1^*, \dots, x_{k-1}^*, j) = [y_j, x_1^*, \dots, x_{j-1}^*, x, x_j^*, \dots, x_{k-1}^*, y_1, \dots, y_j, y_{j-1}, \dots, y_k, j] \tag{86}$$

That is, based on the value of the auxiliary variable $j \in \{1, \dots, k\}$, we first swap y_j and x, and then we swap the rest $(k-1)$ y's with all of the x^*. Note that for the fixed j that is an involution. To check that iMCMC provides the equivalent chain, we evaluate the probability to accept y_j as the next sample. That is

$$P = \min\left\{1, \frac{p(y_j)q(x \mid y_j) \prod_{i=1}^{k-1} q(x_i^* \mid y_j) p(j \mid x_1^*, \dots, x_{j-1}^*, x, x_j^*, \dots, x_{k-1}^*, y_j) \prod_{i=1, i\neq j}^{k} q(y_i \mid x)}{p(x) \prod_{i=1}^{k} q(y_i \mid x) p(j \mid y_1, \dots, y_k, x) \prod_{i=1}^{k-1} q(x_i^* \mid y_j)}\right\} = \tag{87}$$

$$= \min\left\{1, \frac{p(y_j)q(x \mid y_j) p(j \mid x_1^*, \dots, x_{j-1}^*, x, x_j^*, \dots, x_{k-1}^*, y_j)}{p(x)q(y_j \mid x) p(j \mid y_1, \dots, y_k, x)}\right\} = \tag{88}$$

$$= \min\left\{1, \frac{p(y_j)q(x \mid y_j) w(x, y_j) (\sum_{i=1}^{k} w(y_i, x))}{p(x)q(y_j \mid x) w(y_j, x) (\sum_{i=1}^{k-1} w(x_i^*, y_j) + w(x, y_j))}\right\} = \tag{89}$$

$$= \min\left\{1, \frac{p(y_j)q(x \mid y_j) p(x) q(y_j \mid x) \lambda(x, y_j) (\sum_{i=1}^{k} w(y_i, x))}{p(x)q(y_j \mid x) p(y_j) q(x \mid y_j) \lambda(y_j, x) (\sum_{i=1}^{k-1} w(x_i^*, y_j) + w(x, y_j))}\right\} = \tag{90}$$

$$= \min\left\{1, \frac{w(y_1, x) + \ldots + w(y_k, x)}{w(x_1^*, y) + \ldots + w(x_{k-1}^*, y) + w(x, y)}\right\}. \tag{91}$$

Note that the distribution of y's and j is the same as in Algorithm 4, hence, the probability to generate proposal y_j is the same, as well as the probability to accept this proposal.

B.4. Sample-Adaptive MCMC

Algorithm 5 Sample-Adaptive MCMC

input target density $p(x)$, integer N, aggregation function $g(x_1, \dots, x_N)$, proposal $q\left(x_{N+1} \middle| g(x_1, \dots, x_N)\right)$

samples $= \emptyset$
initialize set $S = \{x_1, \dots, x_N\}$
for $i = 0 \dots n$ **do**
  sample $x_{N+1} \sim q\left(x_{N+1} \middle| g(S)\right)$
  define $S_{-i} = (S$ with x_i replaced with $x_{N+1})$, $S_{-(N+1)} = S$
  evaluate $\lambda_i = q\left(x_i \middle| g(S_{-i})\right)/p(x_i), \quad i = 1, \dots, N+1$
  set $j = i$ with probability $\lambda_i / (\sum_{i=1}^{N+1} \lambda_i)$
  $S \leftarrow S_{-j}$
  samples $=$ samples $\cup\, S$
end for
output samples

We begin the proof with the recall of the Sample-Adaptive MCMC (SA-MCMC) algorithm (Algorithm 5). In Algorithm 5, the output of function g does not depend on the order of arguments, i.e. $g(x) = g(\pi(x))$, where π is an arbitrary permutation of arguments.

To write SA-MCMC as Involutive MCMC, we consider the joint distribution and the family of involutions as follows.

$$p(x_1, \dots, x_{N+1}, j) = \prod_{i=1}^{N} p(x_i) q(x_{N+1} \,|\, g(x_1, \dots, x_N)) p(j \,|\, x_1, \dots, x_{N+1}), \tag{92}$$

$$p(j \,|\, x_1, \dots, x_{N+1}) = \frac{\lambda_j}{(\sum_{j=1}^{N+1} \lambda_j)}, \quad \lambda_j = q(x_j \,|\, g(S_{-j}))/p(x_j), \quad j = 1, \dots, N+1 \tag{93}$$

$$f_j(x_1, \dots, x_{N+1}, j) = f(x_1, \dots, x_{j-1}, x_{N+1}, x_{j+1}, \dots, x_N, x_j, j) \tag{94}$$

Here S_{-j} is the current set of samples $S = \{x_1, \dots, x_N\}$, where x_j is replaced with x_{N+1}, and $S_{-(N+1)} = S$. The involution family operates as follows. Based on the value of the auxiliary variable $j \in \{1, \dots, N+1\}$, we swap x_j and x_{N+1} and leave the rest of arguments untouched. For the fixed j, such function is an involution. One more important thing to note is that now our target distribution is the product $\prod_{i=1}^{N} p(x_i)$. To demonstrate that SA-MCMC is equivalent to Involutive MCMC with aforementioned distribution and involutions, we evaluate the probability to accept the point proposed by f_j.

$$P = \min\left\{1, \frac{p(x_{N+1}) \prod_{i=1, i\neq j}^{N} p(x_i) q(x_j \,|\, g(S_{-j})) p(j \,|\, S_{-j}, x_j)}{\prod_{i=1}^{N} p(x_i) q(x_{N+1} \,|\, g(S)) p(j \,|\, S, x_{N+1})}\right\} = \min\left\{1, \frac{p(x_{N+1}) q(x_j \,|\, g(S_{-j})) p(j \,|\, S_{-j}, x_j)}{p(x_j) q(x_{N+1} \,|\, g(S)) p(j \,|\, S, x_{N+1})}\right\} \tag{95}$$

Now we define $S' = S_{-j}$ and $S'_{-i} \leftarrow (S'$ with i-th element replaced by $x_j)$. If we neglect the order of elements, then $S'_{-i} = S_{-i}$ for $i \neq j$, $S'_{-j} = S$ and $S'_{-(N+1)} = S_{-j}$. Using the fact that the order of arguments in the aggregation function $g(\cdot)$ does not matter, we obtain

$$p(j \,|\, S_{-j}, x_j) = \frac{q(x_{N+1} \,|\, g(S))}{p(x_{N+1})\left(q(x_{N+1} \,|\, g(S))/p(x_{N+1}) + \sum_{i=1, i\neq j}^{N} q(x_i \,|\, g(S_{-i}))/p(x_i) + q(x_j \,|\, g(S_{-j}))/p(x_j)\right)} \tag{96}$$

$$= \frac{q(x_{N+1} \,|\, g(S))}{p(x_{N+1})\left(\sum_{i=1}^{N+1} q(x_i \,|\, g(S_{-i}))/p(x_i)\right)} \tag{97}$$

Putting this equation into (95), we obtain

$$P = \min\left\{1, \frac{q(x_j \mid g(S_{-j}))}{p(x_j)p(j \mid S, x_{N+1})\left(\sum_{i=1}^{N+1} q(x_i \mid g(S_{-i}))/p(x_i)\right)}\right\} = 1. \tag{98}$$

Thus, generating the auxiliary variable j we accept the point $f_j(x_1, \ldots, x_{N+1}, j)$ with probability 1. Since the distribution of j and the corresponding point $f_j(x_1, \ldots, x_{N+1}, j)$ are the same as in Algorithm 5, we have obtained the equivalent scheme in terms of Involutive MCMC.

B.4.1. Generalization of Sample-Adaptive MCMC

From the equations above it is easy to discard the permutation-invariance property of $g(\ldots)$. Then we just denote S to be an ordered array $S = [x_1, \ldots, x_N]$ instead of a set, and accept the proposed swap with probability

$$P_j = \min\left\{1, \frac{p(x_{N+1})q(x_j \mid S_{-j})p(j \mid S_{-j}, x_j)}{p(x_j)q(x_{N+1} \mid S)p(j \mid S, x_{N+1})}\right\}. \tag{99}$$

Then the pseudo-code of the algorithm slightly changes (see Algorithm 6).

Algorithm 6 Generalized Sample-Adaptive MCMC

input target density $p(x)$, integer N, proposal $q\left(x_{N+1}\middle| x_1, \ldots, x_N\right)$

samples $= \emptyset$
initialize array $S = [x_1, \ldots, x_N]$
for $i = 0 \ldots n$ **do**
  sample $x_{N+1} \sim q\left(x_{N+1}\middle| S\right)$
  define $S_{-i} = S($ with x_i replaced by $x_{N+1})$, $\quad S_{-(N+1)} = S$
  evaluate $\lambda_i = q\left(x_i\middle| S_{-i}\right)/p(x_i), \quad i = 1, \ldots, N+1$
  set $j = i$ with probability $\lambda_i/(\sum_{i=1}^{N+1} \lambda_i)$
  evaluate acceptance probability $P = \min\left\{1, \frac{p(x_{N+1})q(x_j \mid S_{-j})p(j \mid S_{-j}, x_j)}{p(x_j)q(x_{N+1} \mid S)p(j \mid S, x_{N+1})}\right\}$
  $S \leftarrow \begin{cases} S_{-j}, \text{ with probability } P \\ S, \text{ with probability } (1-P) \end{cases}$
  samples $=$ samples $\cup\, S$
end for
output samples

B.5. Reversible-Jump MCMC

B.5.1. Reversible-Jump MCMC from (Green & Hastie, 2009)

Algorithm 7 Reversible-Jump MCMC from (Green & Hastie, 2009)

input target density $p(x^{(k)}, k)$, auxiliary distributions $q(u \mid m)$ and $q'(u \mid m)$, move functions $h_m(x, u)$
  initialize state $= [x^{(k)}, k]$
  for $i = 0 \dots n$ **do**
    unpack $[x^{(k)}, k] \leftarrow$ state
    sample move type $m \sim p(m \mid x^{(k)}, k)$
    sample auxiliary $u \sim q(u \mid m)$
    move type m defines k'
    evaluate $[x^{(k')}, u'] = h_m(x^{(k)}, u)$
    evaluate $P = \min\left\{1, \frac{p(x^{(k')},k')p(m \mid x^{(k')},k')q'(u' \mid m)}{p(x^{(k)},k)p(m \mid x^{(k)},k)q(u \mid m)} \left| \frac{\partial h_m}{\partial [x^{(k)}, u]} \right| \right\}$
    accept state $\leftarrow \begin{cases} [x^{(k')}, k'], & \text{with probability } P \\ [x^{(k)}, k], & \text{with probability } (1 - P) \end{cases}$
    state$_i \leftarrow$ state
  end for
output samples $\{\text{state}_0, \dots, \text{state}_n\}$

Reversible-Jump MCMC (Green, 1995) has multiple formulations, which vary significantly both in notation used and in the sampling procedure. Here we choose to stay close to (Green & Hastie, 2009) for illustrative purposes (see pseudo-code in Algorithm 7). Note that the move type m index both models k and k', as well as the smooth map h_m. Indeed, for a proper scheme, auxiliary distributions $q'(u \mid m)$ and $q(u \mid m)$ are defined such that the dimension of $[x^{(k)}, u]$ matches the dimension of $[x^{(k')}, u']$ and the dimension for the input of h_m.

To describe Algorithm 7 in terms of iMCMC, we consider the joint distribution:

$$p(x, k, m, u) = p(x^{(k)}, k)p(m \mid x^{(k)}, k)p(u \mid m, k), \tag{100}$$

where we define $p(u \mid m, k)$ such that for the move type m that goes from k to k' we have $p(u \mid m, k) = q(u \mid m)$ and $p(u \mid m, k') = q'(u \mid m)$. We can do it because m defines both models k and k'. The family of involutions is then defined as follows.

$$f_m(x^{(k)}, u, k) = [h_m(x^{(k)}, u), k'] = [x^{(k')}, u', k'], \quad f_m(x^{(k')}, u', k') = [h_m^{-1}(x^{(k')}, u'), k] = [x^{(k)}, u, k] \tag{101}$$

Here index m choose such involution that map model index k to k' and vice versa. As well as in (Green & Hastie, 2009), mapping from k' to k we apply the inverse h_m^{-1}. For a concrete example of move types and functions h_m, we refer the reader to Section 3 of (Green & Hastie, 2009). The acceptance probability then is in total agreement with Algorithm 7:

$$P = \min\left\{1, \frac{p(x^{(k')}, k')p(m \mid x^{(k')}, k')q'(u' \mid m)}{p(x^{(k)}, k)p(m \mid x^{(k)}, k)q(u \mid m)} \left| \frac{\partial h_m}{\partial [x^{(k)}, u]} \right| \right\}. \tag{102}$$

B.5.2. Another formulation

In the previous section, we encapsulate the knowledge about the next proposed model in the index m. However, the formulation becomes more transparent if we sample the index of the next proposed model explicitly. The following algorithm can be seen as a more general version of the formulation of Reversible-Jump MCMC from (Gagnon & Doucet, 2019). That is, consider the joint distribution

$$p(x, k, j, u) = p(x^{(k)}, k)p(j \mid x^{(k)}, k)p(u^{(k)} \mid x^{(k)}, k, j), \tag{103}$$

where j is the index of the next model. Here we add superscripts for u to highlight that the choice of auxiliary variables relies on the current model k. Usually this is done such that all vectors lie in the same vector space, i.e. $[x^{(k)}, u^{(k)}] \in \mathbb{R}^d \ \ \forall k$. The

involution f then is

$$f(x^{(k)}, u^{(k)}, k, j) = [h_{kj}(x^{(k)}, u^{(k)}), j, k] = [x^{(j)}, u^{(j)}, j, k], \quad h_{jk}(x^{(j)}, u^{(j)}) = h_{kj}^{-1}(x^{(j)}, u^{(j)}) = [x^{(k)}, u^{(k)}]. \quad (104)$$

Here the involution f maps $[x, u]$ based on the indeces of the current model k and the next model j. Note that mapping from k to j via h_{kj} we are obliged to perform the inverse map h_{jk} using the inverse function h_{kj}^{-1}. The acceptance probability is then

$$P = \min\left\{1, \frac{p(x^{(j)}, j)p(k \mid x^{(j)}, j)p(u^{(j)} \mid x^{(j)}, j, k)}{p(x^{(k)}, k)p(j \mid x^{(k)}, k)p(u^{(k)} \mid x^{(k)}, k, j)} \left|\frac{\partial h_{kj}}{\partial [x^{(k)}, u^{(k)}]}\right|\right\}. \quad (105)$$

See the pseudo-code in Algorithm 8. Note that unlike Algorithm 7, here we have a single smooth map from model k to model j. This limitation can be easily removed via Trick 2 by considering the family of involutions

$$f_m(x^{(k)}, u^{(k)}, k, j) = [h_{mkj}(x^{(k)}, u^{(k)}), j, k] = [x^{(j)}, u^{(j)}, j, k], \quad h_{mjk}(x^{(j)}, u^{(j)}) = h_{mkj}^{-1}(x^{(j)}, u^{(j)}) = [x^{(k)}, u^{(k)}], \quad (106)$$

where we can sample index m conditioned on the current state $[x^{(k)}, u^{(k)}, k, j]$.

Finally, we discuss the usage of Tricks from Section 3 here. Trick 2 is explicitly used here when we define a family of involutions and stochastically choose one from the family. The auxiliary direction from Trick 3 here is in the form of indices k and j, which define the smooth map h_{kj} and its inverse $h_{jk} = h_{kj}^{-1}$. Trick 1 can be found here if we define the target distribution as

$$p(x^{(k)}, u^{(k)}, k) = p(x^{(k)}, k)p(u^{(k)} \mid x^{(k)}, k), \quad (107)$$

in order to match the dimensions of all models $[x^{(k)}, u^{(k)}] \in \mathbb{R}^d \ \ \forall k$. As well as in Trick 1, we sample from extended distribution $p(x^{(k)}, u^{(k)}, k)$, and then discard all $u^{(k)}$.

Algorithm 8 Reversible-Jump MCMC

input target density $p(x^{(k)}, k)$, distribution of next models $p(j \mid x^{(k)}, k)$, auxiliary distributions $p(u^{(k)} \mid x^{(k)}, k, j)$
  initialize state $= [x^{(k)}, k]$
  for $i = 0 \dots n$ **do**
    unpack $[x^{(k)}, k] \leftarrow$ state
    sample next model $j \sim p(j \mid x^{(k)}, k)$
    sample auxiliary $u^{(k)} \sim p(u^{(k)} \mid x^{(k)}, k, j)$
    propose $[x^{(j)}, u^{(j)}] = h_{kj}(x^{(k)}, u^{(k)})$
    evaluate $P = \min\left\{1, \frac{p(x^{(j)},j)p(k \mid x^{(j)},j)p(u^{(j)} \mid x^{(j)},j,k)}{p(x^{(k)},k)p(j \mid x^{(k)},k)p(u^{(k)} \mid x^{(k)},k,j)} \left|\frac{\partial h_{kj}}{\partial [x^{(k)},u^{(k)}]}\right|\right\}$
    accept state $\leftarrow \begin{cases} [x^{(j)}, j], \text{ with probability } P \\ [x^{(k)}, k], \text{ with probability } (1 - P) \end{cases}$
    state$_i \leftarrow$ state
  end for
output samples $\{\text{state}_0, \dots, \text{state}_n\}$

B.6. Hybrid Monte Carlo

Algorithm 9 Hybrid Monte Carlo

input joint density $p(x, v) = p(x)p(v)$, auxiliary distribution $p(v) = \mathcal{N}(v \mid 0, 1)$, number of Leap-Frog steps k, step size ε
  initialize x
  for $i = 0 \dots n$ **do**
    sample $v \sim \mathcal{N}(v \mid 0, 1)$
    propose $[x', v'] = FL^k(x, v)$
    evaluate $P = \min\{1, \frac{p(x', v')}{p(x, v)}\}$
    accept $x \leftarrow \begin{cases} x', \text{ with probability } P \\ x, \text{ with probability } (1 - P) \end{cases}$
    $x_i \leftarrow x$
  end for
output $\{x_0, \dots, x_n\}$

Hybrid Monte Carlo (Duane et al., 1987) relies on the numerical integration of Hamiltonian dynamics via the Leap-Frog operator L. For target density $p(x)$, the Hamiltonian is defined as $H(x, v) = -\log p(x, v)$, where $p(x, v) = p(x)p(v)$ is the joint distribution, and $p(v) = \mathcal{N}(v \mid 0, 1)$ is the auxiliary distribution. In the case of independent v (i.e., $p(x, v) = p(x)p(v)$), the Leap-Frog operator $L : [x(t), v(t)] \to [x(t + \varepsilon), v(t + \varepsilon)]$ is defined as follows.

$$v(t + \varepsilon/2) = v(t) - \frac{\varepsilon}{2}\nabla_x(-\log p(x(t))) \tag{108}$$

$$x(t + \varepsilon) = x(t) + \varepsilon\nabla_v(-\log p(v(t + \varepsilon/2))) \tag{109}$$

$$v(t + \varepsilon) = v(t + \varepsilon/2) - \frac{\varepsilon}{2}\nabla_x(-\log p(x(t + \varepsilon))) \tag{110}$$

Flip operator F denotes the negation of the auxiliary variable (momentum) v: $F : [x, v] \to [x, -v]$. These operators together yields the involutive map FL, which is used in Algorithm 9. To demonstrate this, we demonstrate that $FLFL = 1$, i.e. double application of the operator FL results in identity function.

$$v(t + \varepsilon/2) = v(t) - \frac{\varepsilon}{2}\nabla_x(-\log p(x(t))) \tag{111}$$

$$x(t + \varepsilon) = x(t) + \varepsilon\nabla_v(-\log p(v(t + \varepsilon/2))) \tag{112}$$

$$v(t + \varepsilon) = v(t + \varepsilon/2) - \frac{\varepsilon}{2}\nabla_x(-\log p(x(t + \varepsilon))) \tag{113}$$

$$v(t + 3/2\varepsilon) = -v(t + \varepsilon) - \frac{\varepsilon}{2}\nabla_x(-\log p(x(t + \varepsilon))) = -v(t + \varepsilon/2) \tag{114}$$

$$x(t + 2\varepsilon) = x(t + \varepsilon) + \varepsilon\nabla_v(-\log p(v(t + 3/2\varepsilon))) = x(t) \tag{115}$$

$$v(t + 2\varepsilon) = v(t + 3/2\varepsilon) - \frac{\varepsilon}{2}\nabla_x(-\log p(x(t + 2\varepsilon))) = -v(t) \tag{116}$$

Note that here we greatly rely on the symmetry $p(v) = p(-v)$. After the last equation we negate the momentum variable once again yielding $FLFL : [x(t), v(t)] \to [x(t), v(t)]$. Note that having $FLFL = 1$ we can easily obtain the inverse of the Leap-Frog operator $L^{-1} = FLF$. Using the formula for the inverse Leap-Frog we have

$$FL^kFL^k = FL^kFLFFL^{k-1} = FL^{k-1}FL^{k-1} = \dots = FLFL = 1. \tag{117}$$

Thus, an arbitrary number of L can be composed in the involution FL^k.

Using the involution FL^k, the formulation of HMC in terms of iMCMC is now straightforward. Consider the joint distribution $p(x, v) = p(x)\mathcal{N}(v \mid 0, 1)$ and the involutive function FL^k, the acceptance probability according to iMCMC (Algorithm 1) is then

$$P = \min\left\{1, \frac{p(FL^k(x, v))}{p(x, v)}\left|\frac{\partial FL^k}{\partial[x, v]}\right|\right\}. \tag{118}$$

Finally, it is easy to see that FL^k is volume-preserving since the transformations on the each step of L are volume-preserving, e.g. (108) maps $[x(t), v(t)] \to [x(t), v(t+\varepsilon/2)]$ since it is an identity map w.r.t. $x(t)$, and $|\partial v(t+\varepsilon/2)/\partial v(t)| = 1$ it is volume-preserving.

Another possible way to represent HMC in terms of iMCMC is to use Trick 3 and introduce the directional variable $p(d) = \text{Uniform}\{-1, +1\}$. Then the involutive map is defined as

$$f(x, v, d) = [T_d(x, v), -d], \quad T_{d=+1} = L, \quad T_{d=-1} = L^{-1}. \tag{119}$$

This formulation allow for a more general formulation that does not rely on the symmetry $p(v) = p(-v)$ as HMC. Indeed, the inverse Leap-Frog operator L^{-1} can be obtained just by the inversion of the time:

$$v(t-\varepsilon/2) = v(t) + \frac{\varepsilon}{2}\nabla_x(-\log p(x(t))) \tag{120}$$

$$x(t-\varepsilon) = x(t) - \varepsilon\nabla_v(-\log p(v(t-\varepsilon/2))) \tag{121}$$

$$v(t-\varepsilon) = v(t-\varepsilon/2) + \frac{\varepsilon}{2}\nabla_x(-\log p(x(t-\varepsilon))) \tag{122}$$

B.7. RMHMC

Algorithm 10 Riemann Manifold HMC

input joint density $p(x, v)$, auxiliary distribution $p(v) = \mathcal{N}(v \,|\, 0, G(x))$, number of Leap-Frog steps k, step size ε
  initialize x
  for $i = 0 \dots n$ **do**
    sample $v \sim \mathcal{N}(v \,|\, 0, G(x))$
    propose $[x', v'] = FL^k(x, v)$
    evaluate $P = \min\{1, \frac{p(x',v')}{p(x,v)}\}$
    accept $x \leftarrow \begin{cases} x', \text{ with probability } P \\ x, \text{ with probability } (1-P) \end{cases}$
    $x_i \leftarrow x$
  end for
output $\{x_0, \dots, x_n\}$

In Riemann Manifold HMC (Girolami & Calderhead, 2011), the authors propose to take into account the "curvature" of the space during sampling by considering the following Hamiltonian

$$H(x, v) = -\log p(x) + \tfrac{1}{2}\log|G(x)| + \tfrac{1}{2}v^T G(x)^{-1} v. \tag{123}$$

As you can see from Algorithm 10, the pseudo-code for RMHMC is almost the same as for HMC (see B.6). The key difference between them is the integration operator L. Since the Hamiltonian $H(x, v)$ is not separable, we need to use the implicit numerical scheme to guarantee volume-preserving and involutive properties. The integration operator L is defined as follows.

$$v(t+\varepsilon/2) = v(t) - \frac{\varepsilon}{2}\nabla_x H(x(t), v(t+\varepsilon/2)) \tag{124}$$

$$x(t+\varepsilon/2) = x(t) + \frac{\varepsilon}{2}\nabla_v H(x(t), v(t+\varepsilon/2)) \tag{125}$$

$$x(t+\varepsilon) = x(t+\varepsilon/2) + \frac{\varepsilon}{2}\nabla_v H(x(t+\varepsilon), v(t+\varepsilon/2)) \tag{126}$$

$$v(t+\varepsilon) = v(t+\varepsilon/2) - \frac{\varepsilon}{2}\nabla_x H(x(t+\varepsilon), v(t+\varepsilon/2)) \tag{127}$$

The involution can be constructed as FL, where F is the negation of v: $F : [x, v] \to [x, -v]$. To demonstrate this, we integrate further in time from $[x(t+\varepsilon), -v(t+\varepsilon)]$ obtaining $FLFL = 1$ (double application yields identity function). That

is, applying step (124), we get

$$v(t+3/2\varepsilon) = -v(t+\varepsilon) - \frac{\varepsilon}{2}\nabla_x H(x(t+\varepsilon), v(t+3/2\varepsilon)) \tag{128}$$

$$v(t+\varepsilon) = -v(t+3/2\varepsilon) - \frac{\varepsilon}{2}\nabla_x H(x(t+\varepsilon), -v(t+3/2\varepsilon)) \implies -v(t+3/2\varepsilon) = v(t+\varepsilon/2) \tag{129}$$

$$\tag{130}$$

Here we use $\nabla_x H(x,v) = \nabla_x H(x,-v)$. Further, applying step (125), we get

$$x(t+3/2\varepsilon) = x(t+\varepsilon) + \frac{\varepsilon}{2}\nabla_v H(x(t+\varepsilon), v(t+3/2\varepsilon)) = x(t+\varepsilon/2), \tag{131}$$

where we use $\nabla_v H(x,-v) = -\nabla_v H(x,v)$. The last two steps (126) and (127) follow the same logic.

$$x(t+2\varepsilon) = x(t+3/2\varepsilon) + \frac{\varepsilon}{2}\nabla_v H(x(t+2\varepsilon), v(t+3/2\varepsilon)) = x(t) \tag{132}$$

$$v(t+2\varepsilon) = v(t+3/2\varepsilon) - \frac{\varepsilon}{2}\nabla_x H(x(t+2\varepsilon), v(t+3/2\varepsilon)) = -v(t) \tag{133}$$

Further negation of $-v(t)$ results in the initial point $[x(t), v(t)]$. Thus, FL is an involution ($FLFL = 1$) and FL^k is also an involution:

$$FL^kFL^k = FL^{k-1}F(FLFL)L^{k-1} = FL^{k-1}FL^{k-1} = \ldots = 1. \tag{134}$$

Using the involution FL^k, the formulation of RMHMC in terms of iMCMC is now straightforward. Consider the joint distribution $p(x,v) = p(x)\mathcal{N}(v \mid 0, G(x))$ and the involutive function FL^k, the acceptance probability according to iMCMC (Algorithm 1) is then

$$P = \min\left\{1, \frac{p(FL^k(x,v))}{p(x,v)}\left|\frac{\partial FL^k}{\partial[x,v]}\right|\right\}. \tag{135}$$

Finally, it is easy to see that FL^k is volume-preserving. For illustrative purposes, we evaluate the Jacobian of the first two steps (124) and (125).

$$\frac{\partial x(t+\varepsilon/2)}{\partial x(t)} = 1 + \frac{\varepsilon}{2}\nabla_{vx}H(x(t), v(t+\varepsilon/2)) + \frac{\varepsilon}{2}\nabla_{vv}H(x(t), v(t+\varepsilon/2))\frac{\partial v(t+\varepsilon/2)}{\partial x(t)} \tag{136}$$

$$\frac{\partial x(t+\varepsilon/2)}{\partial v(t)} = \frac{\varepsilon}{2}\nabla_{vv}H(x(t), v(t+\varepsilon/2))\frac{\partial v(t+\varepsilon/2)}{\partial v(t)} \tag{137}$$

$$\frac{\partial v(t+\varepsilon/2)}{\partial x(t)} = -\frac{\varepsilon}{2}\nabla_{xx}H(x(t), v(t+\varepsilon/2)) \tag{138}$$

$$\frac{\partial v(t+\varepsilon/2)}{\partial v(t)} = 1 - \frac{\varepsilon}{2}\nabla_{xv}H(x(t), v(t+\varepsilon/2))\frac{\partial v(t+\varepsilon/2)}{\partial v(t)} \tag{139}$$

$$\left|\frac{\partial FL^k}{\partial[x,v]}\right| = \left(1 + \frac{\varepsilon}{2}\nabla_{vx}H(x(t), v(t+\varepsilon/2))\right)\frac{\partial v(t+\varepsilon/2)}{\partial v(t)} = 1 \tag{140}$$

B.8. NeuTra

Algorithm 11 NeuTra

input target density $p_x(x)$, auxiliary density $p(v) = \mathcal{N}(v \,|\, 0, 1)$, flow $T(x)$
  initialize z
  for $i = 0 \dots n$ **do**
    sample $v \sim p(v) = \mathcal{N}(v \,|\, 0, 1)$
    propose $[z', v'] = FL^k(z, v)$, where the target density for Leap-Frog is $p_z(z, v) = p_x(T(z))|\frac{\partial T}{\partial z}|p(v)$
    evaluate $P = \min\left\{1, \frac{p_z(z', v')}{p_z(z, v)}\right\}$
    accept $x \leftarrow \begin{cases} z', \text{ with probability } P \\ z, \text{ with probability } (1 - P) \end{cases}$
    $x_i \leftarrow T(z)$
  end for
output samples $\{x_0, \dots, x_n\}$

In the recent paper (Hoffman et al., 2019), the authors learn an invertible transformation $T^{-1} : X \to Z$ to map the target random variable $x \in X$ with the density $p_x(x)$ into another random variable $z \in Z$, which has more simple geometry of density levels. Further, they run HMC in Z with the target density $p_z(z) = p_x(T(z))|\partial T/\partial z|$. Finally, one can obtain samples in the original space X by mapping the collected samples using $T : Z \to X$. We provide the pseudo-code in Algorithm 11.

A straightforward application of Trick 4 allows for iMCMC formulation of NeuTra. That is, the joint distribution is just the same as in HMC

$$p(x, v) = p_x(x)\mathcal{N}(v \,|\, 0, 1). \tag{141}$$

For the involutive map, we take

$$f(x, v) = \begin{bmatrix} T \\ 1 \end{bmatrix} \circ F \circ L^k \circ \begin{bmatrix} T^{-1} \\ 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}, \tag{142}$$

where F is the velocity flip operator, L is the Leap-Frog, and the notation $\begin{bmatrix} T^{-1} \\ 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix}$ means element-wise application $(x, v) \to (T^{-1}(x), v)$. Note that the only necessary condition for the operators L and F is the $(F \circ L^k)^{-1} = F \circ L^k$. Then, by the straightforward evaluation $f(f(x, v))$ we can see that f is an involution. To obtain an equivalent sampler to NeuTra we choose the joint density for L as $p(z, v) = p_x(T(z))|\partial T/\partial z|p(v)$. Thus, we obtain the same dynamics in Z. However, note that iMCMC assumes the acceptance test in the original space X, while NeuTra performs the acceptance test in Z. Nevertheless, for an initial point x and the velocity $v \sim p(v)$, Algorithm 1 gives us the following acceptance test

$$P = \min\left\{1, \frac{p(f(x, v))}{p(x, v)}\left|\frac{\partial f(x, v)}{\partial [x, v]}\right|\right\}, \quad f(x, v) = \begin{bmatrix} T \\ 1 \end{bmatrix} \circ F \circ L^k \circ \begin{bmatrix} T^{-1} \\ 1 \end{bmatrix} \begin{bmatrix} x \\ v \end{bmatrix} \tag{143}$$

Using the chain rule, we have

$$\left|\frac{\partial f(x, v)}{\partial [x, v]}\right| = \left|\frac{\partial T}{\partial y}\right|_{y = FL^kT^{-1}(x)}\left|\frac{\partial T^{-1}}{\partial x}\right| = \left|\frac{\partial T}{\partial y}\right|_{y = FL^kT^{-1}(x)}\left|\frac{\partial T}{\partial y}\right|^{-1}_{y = T^{-1}(x)} \tag{144}$$

Denoting $z = T^{-1}(x)$, and $[z', v'] = FL^k(z, v)$, we have

$$P = \min\left\{1, \frac{p_x(T(z'))p(v')}{p_x(T(z))p(v)}\left|\frac{\partial T}{\partial y}\right|_{y = z'}\left|\frac{\partial T}{\partial y}\right|^{-1}_{y = z}\right\} = \min\left\{1, \frac{p_z(z', v')}{p_z(z, v)}\right\}. \tag{145}$$

Thus, we obtain the same acceptance probability, and, hence, equivalent kernel to Algorithm 11.

B.9. A-NICE-MC

Algorithm 12 A-NICE-MC

input target density $p(x, v) = p(x)\mathcal{N}(v \mid 0, 1)$, NICE-proposal $T(x, v)$ and $T^{-1}(x, v)$
  initialize x
  for $i = 0 \dots n$ **do**
    sample $v \sim \mathcal{N}(v \mid 0, 1)$
    sample $d \sim \text{Uniform}\{-1, +1\}$
    propose $[x', v'] = T_d(x, v)$, where $T_{d=+1} = T$ and $T_{d=-1} = T^{-1}$
    evaluate $P = \min\left\{1, \frac{p(x',v')}{p(x,v)}\right\}$
    accept $x \leftarrow \begin{cases} x', \text{ with probability } P \\ x, \text{ with probability } (1 - P) \end{cases}$
    $x_i \leftarrow x$
  end for
output samples $\{x_0, \dots, x_n\}$

We recall A-NICE-MC (Song et al., 2017) in Algorithm 12. The core part of the algorithm is the volume-preserving NICE proposal $T(x, v)$, which is learned before the sampling. Trick 3 with directional variable d allows for a straightforward formulation of A-NICE-MC in terms of iMCMC. Consider the joint distribution

$$p(x, v, d) = p(x)\mathcal{N}(v \mid 0, 1)p(d), \quad p(d) = \text{Uniform}\{-1, +1\}, \tag{146}$$

and the involution

$$f(x, v, d) = [T_d(x, v), -d], \quad T_{d=+1} = T, \;\; T_{d=-1} = T^{-1}. \tag{147}$$

Then it is easy to see that the acceptance probability of iMCMC (Algorithm 1) is the same as the probability P in Algorithm 12.

B.10. L2HMC

Algorithm 13 L2HMC

input target density $p(x, v) = p(x)\mathcal{N}(v \mid 0, 1)$, proposal $T(x, v)$ and $T^{-1}(x, v)$
  initialize x
  for $i = 0 \dots n$ **do**
    sample $v \sim \mathcal{N}(v \mid 0, 1)$
    sample $d \sim \text{Uniform}\{-1, +1\}$
    propose $[x', v'] = T_d(x, v)$, where $T_{d=+1} = T$ and $T_{d=-1} = T^{-1}$
    evaluate $P = \min\left\{1, \frac{p(x',v')}{p(x,v)} \left| \frac{\partial T_d(x,v)}{\partial [x,v]} \right| \right\}$
    accept $x \leftarrow \begin{cases} x', \text{ with probability } P \\ x, \text{ with probability } (1 - P) \end{cases}$
    $x_i \leftarrow x$
  end for
output samples $\{x_0, \dots, x_n\}$

We recall L2HMC (Levy et al., 2017) in Algorithm 13. The core part of the algorithm is the proposal $T(x, v)$, which is learned before the sampling. The only two differences with A-NICE-MC (see B.9) is the form of proposal T (in L2HMC it is not volume-preserving) and the way the proposals are learned. Since here we do not consider the training stage, we can say that the only difference between A-NICE-MC and L2HMC is the Jacobian of deterministic transformation in the test.

Trick 3 with directional variable d allows for a straightforward formulation of L2HMC in terms of iMCMC. Consider the joint distribution

$$p(x, v, d) = p(x)\mathcal{N}(v \mid 0, 1)p(d), \quad p(d) = \text{Uniform}\{-1, +1\}, \tag{148}$$

and the involution

$$f(x, v, d) = [T_d(x, v), -d], \quad T_{d=+1} = T, \;\; T_{d=-1} = T^{-1}. \tag{149}$$

Then it is easy to see that the acceptance probability of iMCMC (Algorithm 1) is the same as the probability P in Algorithm 13.

B.11. HMC with persistent momentum

Algorithm 14 HMC with persistent momentum

input target density $p(x)$, auxiliary distribution $p(v) = \mathcal{N}(v \mid 0, 1)$, number of Leap-Frog steps k, hyperparameter α
  initialize x, v
  for $i = 0 \dots n$ **do**
    update $v \leftarrow v\sqrt{1-\alpha^2} + \alpha\varepsilon, \;\; \varepsilon \sim \mathcal{N}(\varepsilon \mid 0, 1)$
    propose $[x', v'] = FL^k(x, v)$
    evaluate $P = \min\{1, \frac{p(x', v')}{p(x, v)}\}$
    accept $[x, v] \leftarrow \begin{cases} [x', v'], \text{ with probability } P \\ [x, v], \text{ with probability } (1 - P) \end{cases}$
    $x_i \leftarrow x$
    $v \leftarrow -v$
  end for
output $\{x_0, \dots, x_n\}$

The HMC algorithm with persistent momentum (Horowitz, 1991) is usually formulated as in Algorithm 14. The iMCMC formulation of this algorithm can be derived in two ways. One of the ways is to apply Trick 5, we return to it further during the discussion of the generalization of Algorithm 14. For illustrative purposes, we firstly describe a straightforward way where we use involution FL^k as a proposal, and compose it with another two iMCMC kernels. The first kernel $t_1(x', v', a' \mid x, v, a)$ preserves the joint distribution $p(x, v, a) = p(x)p(v)p(a \mid v)$, where $p(v) = \mathcal{N}(v \mid 0, 1)$, and $p(a \mid v) = \mathcal{N}(a \mid v\sqrt{1-\alpha^2}, \alpha^2)$. Note that using the involution $f_1(x, v, a) = [x, a, v]$ that just swaps v and a we accepting the new state $[x, a, v]$ with probability 1. Indeed,

$$P_1 = \left\{1, \frac{p(x)\mathcal{N}(a \mid 0, 1)\mathcal{N}(v \mid a\sqrt{1-\alpha^2}, \alpha^2)}{p(x)\mathcal{N}(v \mid 0, 1)\mathcal{N}(a \mid v\sqrt{1-\alpha^2}, \alpha^2)}\right\} = 1. \tag{150}$$

The second kernel $t_2(x', v' \mid x, v)$ is equivalent to vanilla HMC algorithm with the joint distribution $p(x, v) = p(x)\mathcal{N}(v \mid 0, 1)$ and the involution $f_2(x, v) = FL^k(x, v)$. The third kernel $t_3(x', v' \mid x, v)$ is equivalent to the flip kernel from Trick 5, i.e. iMCMC with the joint distribution $p(x, v) = p(x)\mathcal{N}(v \mid 0, 1)$ and the involution $f_2(x, v) = [x, -v]$. Note that the last kernel preserves the distribution without any test since $p(x, v) = p(x, -v)$.

The obtained composition of iMCMC kernels greatly relies on the fact that $p(x, -v) = p(x, v)$, as well as the original proof (Horowitz, 1991). However, using the Trick 5 we can straightforwardly obtain a generalization of this algorithm as depicted in Algorithm 15. The key idea here is to use an additional directional variable $d \sim \text{Uniform}\{-1, +1\}$ and involution $f(x, v, d) = [T_d(x, v), -d]$, where $T_{d=+1}(x, v) = L^k(x, v)$, and $T_{d=-1}(x, v) = L^{-k}(x, v)$, where L^{-1} is the Leap-Frog inverted in time. Then we can flip the direction d as in Trick 5 since $p(d) = p(-d)$. In the case $p(v) = p(-v)$, and the choice of $t_1(v' \mid v)$ as in Algorithm 14, we obtain the algorithm equivalent to Algorithm 14. Note that in Algorithm 15 we consider the case $p(x, v) = p(x)p(v)$ only to be able to apply the explicit version of the Leap-Frog integrator, the same logic applies for implicit integrators as used in RMHMC (Appendix B.7).

Algorithm 15 Generalized HMC with persistent momentum

input target density $p(x)$, auxiliary distribution $p(v)$, number of Leap-Frog steps k
input iMCMC kernel $t_1(v' \mid v)$ for updating v
  initialize x, v, d
  for $i = 0 \dots n$ **do**
    update $v \sim t_1(\cdot \mid v)$
    propose $[x', v', d'] = [T_d(x, v), -d]$, where $T_{d=+1}(x, v) = L^k(x, v)$, and $T_{d=-1}(x, v) = L^{-k}(x, v)$
    evaluate $P = \min\{1, \frac{p(x', v')}{p(x, v)}\}$
    accept $[x, v, d] \leftarrow \begin{cases} [x', v', d'], & \text{with probability } P \\ [x, v, d], & \text{with probability } (1 - P) \end{cases}$
    flip the direction $d \leftarrow -d$
    $x_i \leftarrow x$
  end for
output $\{x_0, \dots, x_n\}$

B.12. Gibbs sampling

Algorithm 16 Gibbs sampling

input conditional densities $p(x_k \mid \dots, x_{k-1}, x_{k+1}, \dots)$ of the target distribution $p(x_1, \dots, x_n)$
  initialize $x = (x_1, \dots, x_n)$
  for $i = 0 \dots N$ **do**
    for $k = 0 \dots d$ **do**
      sample $x'_k \sim p(x'_k \mid \dots, x'_{k-1}, x_{k+1}, \dots)$
    end for
    $x[i] \leftarrow (x'_1, \dots, x'_n)$
    $x \leftarrow x[i]$
  end for
output $\{x[0], \dots, x[N]\}$

Algorithm 16 describes the Gibbs sampling. Further, we formulate it as the composition of iMCMC kernels, where each kernel is a single step of the inner loop of Algorithm 16. That is, for the transition kernel $t_k(x^k \mid x^{k-1})$ we define the joint distribution as

$$p(x_1, \dots, x_n, v_k) = p(x_1, \dots, x_n) p(v_k \mid \dots, x_{k-1}, x_{k+1}, \dots), \tag{151}$$

and the involutive map f as

$$f(x_1, \dots, x_n, v_k) = [x_1, \dots, x_{k-1}, v_k, x_{k+1}, \dots, x_n, x_k]. \tag{152}$$

It swaps x_k with v_k and leaves the rest of the variables untouched. The acceptance probability of such a proposal is

$$P = \min\left\{1, \frac{p(x_1, \dots x_{k-1}, v_k, x_{k+1}, \dots, x_n) p(x_k \mid \dots, x_{k-1}, x_{k+1}, \dots)}{p(x_1, \dots, x_n) p(v_k \mid \dots, x_{k-1}, x_{k+1}, \dots)}\right\} = 1. \tag{153}$$

Thus, every proposed point will be accepted and we update variables one by one as in the Gibbs sampling. The resulted kernel is

$$t(x^n \mid x^0) = \int \prod_{k=1}^{n-1} dx^k \prod_{k=1}^{n} t_k(x^k \mid x^{k-1}). \tag{154}$$

Another way to describe the Gibbs sampling is to use Trick 5. Consider the augmented distribution $p(x_1 \dots x_n)p(k)p(d)$, where $p(k) = \text{Uniform}\{1, \dots, n\}$, and $p(d) = \text{Uniform}\{-1, +1\}$. Taking the auxiliry distribution as $p(v \mid x_1 \dots x_n, k) =$

$p(v \mid \ldots, x_{k-1}, x_{k+1}, \ldots)$, we set the involution as

$$f(x_1, \ldots, x_n, v, k, d = +1) = [x_1, \ldots, x_{k-1}, v, x_{k+1}, \ldots, x_n, x_k, k+1, -1], \tag{155}$$
$$f(x_1, \ldots, x_n, v, k, d = -1) = [x_1, \ldots, x_{k-2}, v, x_k, \ldots, x_n, x_{k-1}, k-1, +1], \tag{156}$$

That is, moving in the positive direction we swap x_k and v, increment $k \to k+1$ mod n and flip the directional variable $d \to -d$, whereas moving in the negative direction we x_{k-1} and v, decrease $k \to k-1$ mod n and also flip the directional variable $d \to -d$. The acceptance probability of such iMCMC kernel is 1. Composing this kernel with the flip of the direction as in Trick 5, we obtain a composition of kernels, which every n-th sample equals to the samples from Algorithm 16.

B.13. Look Ahead HMC

Algorithm 17 Look Ahead HMC

input target density $p(x)$, auxiliary distribution $p(v) = \mathcal{N}(v \mid 0, 1)$, hyperparameter α
  initialize x, v
  for $i = 0 \ldots n$ **do**
    update $v \leftarrow v\sqrt{1-\alpha^2} + \alpha\varepsilon, \quad \varepsilon \sim \mathcal{N}(\varepsilon \mid 0, 1)$
    evaluate $\pi_k = \min\left\{1 - \sum_{j<k} \pi_j(x, v), \frac{p(FL^k(x,v))}{p(x,v)}\left(1 - \sum_{j<k} \pi_j(FL^k(x, v))\right)\right\}$
    accept $[x, v] \leftarrow \begin{cases} L^k(x, v), & \text{with probability } \pi_k(x, v) \\ [x, -v], & \text{with probability } (1 - \sum_k \pi_k(x, v)) \end{cases}$
    $x_i \leftarrow x$
  end for
output $\{x_0, \ldots, x_n\}$

The Look Ahead HMC algorithm (Sohl-Dickstein et al., 2014) operates by proposing several points for acceptance, which are evaluated with different number of steps in the Leap-Frog integrator (see Algorithm 17). The iMCMC formulation of Look Ahead HMC is similar to the formulation of Horowitz's algorithm (see Appendix B.11). The key feature of Look Ahead HMC is that it use a mixture of involutions in the intermediate kernel.

To describe Look Ahead HMC, we use the following composition of iMCMC kernels. The first kernel $t_1(x', v', a' \mid x, v, a)$ preserves the joint distribution $p(x, v, a) = p(x)p(v)p(a \mid v)$, where $p(v) = \mathcal{N}(v \mid 0, 1)$, and $p(a \mid v) = \mathcal{N}(a \mid v\sqrt{1-\alpha^2}, \alpha^2)$. Note that using the involution $f_1(x, v, a) = [x, a, v]$ that just swaps v and a we accepting the new state $[x, a, v]$ with probability 1. Indeed,

$$P_1 = \left\{1, \frac{p(x)\mathcal{N}(a \mid 0, 1)\mathcal{N}(v \mid a\sqrt{1-\alpha^2}, \alpha^2)}{p(x)\mathcal{N}(v \mid 0, 1)\mathcal{N}(a \mid v\sqrt{1-\alpha^2}, \alpha^2)}\right\} = 1. \tag{157}$$

The second kernel $t_2(x', v', k' \mid x, v, k)$ preserves the joint distribution

$$p(x, v, k) = p(x, v)p(k \mid x, v), \quad p(k \mid x, v) = 1 - \sum_{j<k} \pi_j(x, v), \quad k = 1, \ldots, K, \quad p(0 \mid x, v) = 1 - \sum_{k=1}^{K} \pi_k(x, v) \tag{158}$$
$$\pi_k(x, v) = \min\left\{1 - \sum_{j<k} \pi_j(x, v), \frac{p(FL^k(x, v))}{p(x, v)}\left(1 - \sum_{j<k} \pi_j(FL^k(x, v))\right)\right\}, \tag{159}$$

where $p(k \mid x, v)$ defines the index of involution that we apply on the current step. To be more precise, k defines the number of Leap-Frog steps:

$$f_k(x, v) = FL^k(x, v). \tag{160}$$

The probability to accept $FL^k(x,v)$ is then

$$P = \min\left\{1, \frac{p(FL^k(x,v))p(k \mid FL^k(x,v))}{p(x,v)p(k \mid x,v)}\right\} p(k \mid x,v) = \min\left\{p(k \mid x,v), \frac{p(FL^k(x,v))}{p(x,v)} p(k \mid FL^k(x,v))\right\} = \tag{161}$$

$$= \min\left\{1 - \sum_{j<k} \pi_j(x,v), \frac{p(FL^k(x,v))}{p(x,v)}\left(1 - \sum_{j<k} \pi_j(FL^k(x,v))\right)\right\} = \pi_k(x,v) \tag{162}$$

The third kernel $t_3(x', v' \mid x, v)$ simply negates the auxiliary variable v. That is without any resampling, we just apply $f_3(x,v) = [x, -v]$. Composing all the kernels together we obtain the chain that is equivalent to Algorithm 17.

In the formulation above the sign of v plays the role of directional variable d from Trick 5. However, the same can be done explicitly by considering involutions

$$f_k(x,v,d=+1) = [L^k(x,v), -d], \quad f_k(x,v,d=-1) = [FL^kF(x,v), -d] \tag{163}$$

in the kernel $t_2(x', v', k' \mid x, v, k)$, where $p(d) = \text{Uniform}\{-1, +1\}$.

Further, this Look Ahead technique can be generalized to the case of arbitrary functions T by considering the following family of involutions

$$f_k(x,v,d=+1) = [T^k(x,v), -d], \quad f_k(x,v,d=-1) = [T^{-k}(x,v), -d]. \tag{164}$$

B.14. Non-Reversible Jump

Algorithm 18 Non-Reversible Jump

input target density $p(x^{(k)}, k)$, auxiliary distributions $q_{k\to k'}(u^{(k)})$, smooth maps $T_{k\to k'}(x^{(k)}, u^{(k)})$
  initialize state $= [x^{(k)}, k, \nu]$
  for $i = 0 \dots n$ **do**
    $u \sim \text{Uniform}[0,1]$
    if $u \le \tau$ **then**
      update $x^{(k)}$ staying in the same model k and fixing the direction ν
    else
      unpack $[x^{(k)}, k, \nu] \leftarrow$ state
      $k' = k + \nu$
      sample auxiliary $u^{(k)} \sim q_{k\to k'}(u^{(k)})$
      propose $[x^{(k')}, u^{(k')}] = T_{k\to k'}(x^{(k)}, u^{(k)})$
      evaluate $P = \min\left\{1, \frac{p(x^{(k')},k')q_{k'\to k}(u^{(k')})}{p(x^{(k)},k)q_{k\to k'}(u^{(k)})}\left|\frac{\partial T_{k\to k'}}{\partial[x^{(k)},u^{(k)}]}\right|\right\}$
      accept state $\leftarrow \begin{cases}[x^{(k')}, k', \nu], \text{ with probability } P\\ [x^{(k)}, k, -\nu], \text{ with probability } (1-P)\end{cases}$
    end if
    state$_i \leftarrow$ state
  end for
output samples $\{\text{state}_0, \dots, \text{state}_n\}$

We provide the pseudo-code for Non-Reversible Jump scheme (Gagnon & Doucet, 2019) in Algorithm 18. Further, we describe this algorithm in terms of iMCMC using Trick 5. To build the first kernel $t_1(\cdot \mid \cdot)$, we consider the following joint distribution

$$p(x^{(k)}, u^{(k)}, v^{(k)}, k, \nu, m) = p(x^{(k)}, k)p(\nu)p(u^{(k)} \mid k, \nu)p(m)p(v^{(k)} \mid k), \tag{165}$$

where $p(\nu) = \text{Uniform}\{-1, +1\}$ is analogue of direction d in Trick 3; $p(m) = \text{Bernoulli}(\tau, 1-\tau)$ defines the index of involution applied; $p(v^{(k)} \mid k)$ and $p(u^{(k)} \mid k, \nu)$ define auxiliary variables, which we choose as $p(u^{(k)} \mid k, \nu) = q_{k\to k+\nu}(u^{(k)})$

and $p(v^{(k)}) = q_{k\to k}(v^{(k)})$. With probability $1-\tau$ (when $m=1$), we apply involution

$$f_1(x^{(k)}, u^{(k)}, v^{(k)}, k, \nu) = [T_{k\to(k+\nu)}(x^{(k)}, u^{(k)}), v^{(k)}, k+\nu, -\nu, v^{(k)}] = [x^{(k+\nu)}, u^{(k+\nu)}, v^{(k)}, k+\nu, -\nu], \quad (166)$$

$$T_{k'\to k}(x^{(k')}, u^{(k')}) = T^{-1}_{k\to k'}(x^{(k')}, u^{(k')}) = [x^{(k)}, u^{(k)}]. \quad (167)$$

That is, based on indices k and $k+\nu$ we choose a smooth map that we apply to $x^{(k')}, u^{(k')}$; we also update $k \to k+\nu$ and negate the direction ν. The acceptance probability for such a proposal is

$$P = \min\left\{1, \frac{p(x^{(k+\nu)}, k+\nu)p(u^{(k+\nu)} \mid k+\nu, -\nu)}{p(x^{(k)}, k)p(u^{(k)} \mid k, \nu)} \left| \frac{\partial T_{k\to(k+\nu)}}{\partial[x^{(k)}, u^{(k)}]} \right| \right\}, \quad (168)$$

which is equivalent to the acceptance probability in Algorithm 18, when we denote $k' = k+\nu$ and $p(u^{(k)} \mid k, \nu) = q_{k\to k+\nu}(u^{(k)})$. With probability τ (when $m=0$), we apply involution

$$f_0(x^{(k)}, v^{(k)}, u^{(k)}, k, \nu) = [T_{k\to k}(x^{(k)}, v^{(k)}), u^{(k)}, k, \nu], \quad T_{k\to k}(x^{(k)}, v^{(k)}) = T^{-1}_{k\to k}(x^{(k)}, v^{(k)}), \quad (169)$$

which does not change neither k nor ν. Here we also apply involutive smooth map $T_{k\to k}$ to the vector $[x^{(k)}, v^{(k)}]$ instead of $[x^{(k)}, u^{(k)}]$. Without the loss of generality, we can treat the case of $m=0$ to be equivalent to the corresponding update when $u \le \tau$ in Algorithm 18.

As well as in Trick 5, we combine the obtained kernel t_1 on the joint distribution $p(x^{(k)}, u^{(k)}, v^{(k)}, k, \nu, m)$ with the kernel t_2 on the same distribution. Applying t_2 we do not resample any variables, instead we use the following involution

$$f(\nu, m=0) = [\nu, m], \quad f(\nu, m=1) = [-\nu, m]. \quad (170)$$

The rest of the variables remains the same. Based on the value of m we change only ν to obtain the persistent irreversible movement in the case when ν was negated by the kernel t_1. The combination of kernels t_1 and t_2 yields the sampler that is equivalent to Non-Reversible Jump scheme (Algorithm 18).

B.15. Lifted Metropolis-Hastings

Firstly, we recall a general approach of Lifting in (Turitsyn et al., 2011) following the formulation from (Bierkens et al., 2017). Lifting modifies the reversible kernel T on the state space X by splitting each state $x \in X$ in two replicas: $\{x, +\}$ and $\{x, -\}$. Then, for each replica, the authors introduce its own transition kernel: $T^{(+)}$ for positive replicas and $T^{(-)}$ for negative ones. These transition kernels must satisfy

$$T(x,y)^{(+)}p(x) = T(y,x)^{(-)}p(y), \quad \forall x \neq y, \quad (171)$$

where p is the target distribution. The kernels $T^{(+)}$ and $T^{(-)}$ define in-replica transitions and are obtained from the original kernel T by splitting the support of T using some decision function $\eta : X \to \mathbb{R}$. For non-diagonal elements $x \neq y$ these transitions can be written as

$$T^{(+)}(x,y) = \begin{cases} T(x,y), & \text{if } \eta(y) \ge \eta(x), \\ 0, & \text{if } \eta(y) < \eta(x) \end{cases} \quad \text{and} \quad T^{(-)}(x,y) = \begin{cases} 0, & \text{if } \eta(y) > \eta(x), \\ T(x,y), & \text{if } \eta(y) \le \eta(x) \end{cases}. \quad (172)$$

Inter-replica transitions are defined as

$$T^{(-,+)}(x) = \max\left\{0, \sum_{y:y\neq x} T^{(+)}(x,y) - T^{(-)}(x,y)\right\}, \quad (173)$$

$$T^{(+,-)}(x) = \max\left\{0, \sum_{y:y\neq x} T^{(-)}(x,y) - T^{(+)}(x,y)\right\}. \quad (174)$$

Where $T^{(+,-)}$ define the transition probability from positive replicas to negative ones. Finally, the diagonal elements of $T^{(+)}$ and $T^{(-)}$ are defined as follows.

$$T^{(+)}(x,x) = 1 - T^{(+,-)}(x) - \sum_{y:y\neq x} T^{(+)}(x,y), \quad T^{(-)}(x,x) = 1 - T^{(-,+)}(x) - \sum_{y:y\neq x} T^{(-)}(x,y) \quad (175)$$

Note that

$$T^{(+)}(x,x) = T^{(-)}(x,x) = \min\left\{1 - \sum_{y:y\neq x} T^{(-)}(x,y), 1 - \sum_{y:y\neq x} T^{(+)}(x,y)\right\}. \tag{176}$$

The whole transition kernel on the extended space is defined as

$$\mathcal{T} = \begin{bmatrix} T^{(+)} & T^{(+,-)} \\ T^{(-,+)} & T^{(-)} \end{bmatrix}. \tag{177}$$

To describe Lifting in terms of iMCMC we follow Trick 6 introducing the directional variable $p(d) = \text{Uniform}\{-1, +1\}$, which define the proposal we are currently using to sample new state. Further, we compose this kernel with the flip of d to obtain an irreversible kernel. That is, the first kernel t_1 operates on the following distribution.

$$p(x, v, d) = p(x)p(d)q(v \,|\, x, d), \tag{178}$$

$$q(v \,|\, x, +1) = T^{(+)}(x, v) \;\; \forall v \neq x, \quad q(v \,|\, x, -1) = T^{(-)}(x, v) \;\; \forall v \neq x, \tag{179}$$

$$q(x \,|\, x, +1) = 1 - \sum_{v:v\neq x} T^{(+)}(x, v), \quad q(x \,|\, x, -1) = 1 - \sum_{v:v\neq x} T^{(-)}(x, v) \tag{180}$$

The involutive map is then

$$f_1(x, v, d) = [v, x, -d], \tag{181}$$

which is just the swap of x and v and the negation of d. Kernel t_1 is then obtained by substitution of $p(x, v, d)$ and $f_1(x, v, d)$ into Algorithm 1. Then we compose the first kernel t_1 with the kernel t_2 that just negate the directional variable one more time applying the involution $f_2(x, v, d) = [x, v, -d]$. The composition of kernels t_1 and t_2 we denote as $t(x', v', d' \,|\, x, v, d)$.

To prove that the iMCMC formulation is equivalent to the original chain we consider three following cases. The first case is the transition to the new state $v \neq x$ staying in the same replica (same direction d).

$$\forall x \neq v, \;\; t(v, +1 \,|\, x, +1) = q(v \,|\, x, +1) \min\left\{1, \frac{p(v)q(x \,|\, v, -1)}{p(x)q(v \,|\, x, +1)}\right\} = T^{(+)}(x, v) \min\left\{1, \frac{p(v)T^{(-)}(v, x)}{p(x)T^{(+)}(x, v)}\right\} = T^{(+)}(x, v). \tag{182}$$

Note that the directional variable remains the same because of the double negation: firstly in f_1 and then in f_2. The second case is the staying in the same state x with the same direction.

$$t(x, +1 \,|\, x, +1) = q(x \,|\, x, +1) \min\left\{1, \frac{p(x)q(x \,|\, x, -1)}{p(x)q(x \,|\, x, +1)}\right\} \tag{183}$$

$$= (1 - \sum_{v:v\neq x} T^{(+)}(x, v)) \min\left\{1, \frac{p(x)(1 - \sum_{v:v\neq x} T^{(-)}(x, v))}{p(x)(1 - \sum_{v:v\neq x} T^{(+)}(x, v))}\right\} = T^{(+)}(x, x). \tag{184}$$

The last case is the inter-replica transition of Lifting, which corresponds to the rejection in its iMCMC formulation.

$$t(x, -1 \,|\, x, +1) = 1 - \sum_{v} t(v, +1 \,|\, x, +1) = 1 - \sum_{v:v\neq x} t(v, +1 \,|\, x, +1) - t(x, +1 \,|\, x, +1) = \tag{185}$$

$$= 1 - \sum_{v:v\neq x} T^{(+)}(x, v) - T^{(+)}(x, x) = \tag{186}$$

$$= 1 - \sum_{v:v\neq x} T^{(+)}(x, v) - \min\left\{1 - \sum_{v:v\neq x} T^{(-)}(x, v), 1 - \sum_{v:v\neq x} T^{(+)}(x, v)\right\} = \tag{187}$$

$$= \max\left\{0, \sum_{v:v\neq x} T^{(-)}(x, v) - T^{(+)}(x, v)\right\} = T^{(+,-)}(x). \tag{188}$$

C. Experiments

C.1. Distributions

Here we provide analytical forms of considered target distributions. Target density for MoG2 is:

$$p(x) = \frac{1}{2}\mathcal{N}(x|\mu_1, \sigma_1) + \frac{1}{2}\mathcal{N}(x|\mu_2, \sigma_2) \tag{189}$$

where $\mu_1 = [2, 0]$, $\mu_2 = [-2, 0]$, $\sigma_1^2 = \sigma_2^2 = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$.

For the Bayesian logistic regression, we define likelihood and prior as

$$p(y = 1 \,|\, x, \theta) = \frac{1}{1 + \exp(-x^T\theta_w + \theta_b)}, \quad p(\theta) = \mathcal{N}(\theta \,|\, 0, 0.1). \tag{190}$$

Then the unnormalized density of the posterior distribution for a dataset $D = \{(x_i, y_i)\}_i$ is

$$p(\theta \,|\, D) \propto \prod_i p(y_i \,|\, x_i, \theta) p(\theta). \tag{191}$$

We sample from the posterior distribution on three datasets: German (25 covariates, 1000 data points), Heart (14 covariates, 532 data points) and Australian (15 covariates, 690 data points). We provide all the data with the code in supplementary.

C.2. Effective sample size

The effective sample size (ESS) is defined as the reciprocal of the autocorrelation time. It is designed to represent the number of truly independent samples that would be equivalent to a correlated sample drawn using the chain. There are several approaches to evaluation of autocorrelation time (Thompson, 2010). One of the most common approaches is the initial sequence estimators. That is, the autocorrelation ρ of sequence $\{X_i\}_{i=1}^n$ is estimated as

$$\widehat{\rho} = 1 + 2\sum_{k=1}^{\infty} \rho_k, \quad \widehat{\rho}_k = \frac{1}{ns^2}\sum_{i=1}^{n-k}(X_i - \overline{X}_n)(X_{i+k} - \overline{X}_n), \tag{192}$$

where $\overline{X}_n$ and s^2 are the sample mean and variance of the sequence. Further, assuming the reversibility of the chain, the consecutive pair $\rho_i + \rho_{i+1}$ is always positive (Geyer, 1992). Thus, one can obtain initial positive sequence estimator by truncating the negative values of the sums $\widehat{\rho}_i + \widehat{\rho}_{i+1}$.

However, the initial positive sequence estimator fails to converge to the true autocorrelation in some cases (Thompson, 2010). Moreover, in this paper we cannot rely on the reversibility of the chain since we compare reversible chains with their irreversible analogues. That is why we turn to the batch-means estimator of the autocorrelation time, which operates as follows. It divides the initial sequence $\{X_i\}_{i=1}^n$ into subsequences (batches) of size m and evaluate sample means of each batch. Then we estimate ρ as

$$\widehat{\rho} = m\frac{s_m^2}{s^2}, \tag{193}$$

where s_m^2 is the sample variance of batch means. For the choice of m we follow (Thompson, 2010), and take $n^{1/3}$ batches of the size $m = n^{2/3}$. For multivariate distributions we follow the common practice of evaluating the minimal ESS across all dimensions.

To include computation efforts into the performance evaluation, we calculate ESS per second. We run all the algorithms on a single GPU with batch size 100 sampling 20000 samples, and discarding first 1000 for burn-in. The final formula is

$$\text{ESS/s} = \frac{1}{\rho}\frac{\text{number of samples}}{\text{run time}}. \tag{194}$$

C.3. Irr-MALA

Following Trick 5, we modify the original algorithm by introducing the directional variable $p(d) = \text{Uniform}\{-1, +1\}$. For the first kernel $t_1(x', v', d' \,|\, x, v, d)$, the joint distribution is

$$p(x, v, d) = p(x)\mathcal{N}(v \,|\, x + d\varepsilon\nabla_x \log p(x), 2\varepsilon)p(d),$$

and the involutive map is

$$f_1(x, v, d) = [v, x, -d \cdot \text{sign}(\nabla_x \log p(x)^T \nabla_v \log p(v))].$$

Then the acceptance probability is

$$P = \min\left\{1, \frac{p(v)\mathcal{N}(x \,|\, v + d'\varepsilon\nabla_v \log p(v), 2\varepsilon)}{p(x)\mathcal{N}(v \,|\, x + d\varepsilon\nabla_x \log p(x), 2\varepsilon)}\right\}, \quad d' = -d \cdot \text{sign}\Big(\nabla_x \log p(x)^T \nabla_v \log p(v)\Big). \tag{195}$$

Note that defining the sign of the gradient $\nabla_v \log p(v)$ via d', we ensure that the mean $v + d'\varepsilon\nabla_v \log p(v)$ will be close to the initial point x. The second kernel $t_2(x', v', d' \,|\, x, v, d)$, as well as in Trick 5, is just the flip of the direction d. That is, we do not resample any variables, instead we apply the involution $f_2(x, v, d) = [x, v, -d]$. Combining the kernels t_1 and t_2, we obtain an irreversible chain. We provide the pseudo-code in Algorithm 19.

Algorithm 19 Irr-MALA

input target density $p(x)$, step size ε
  initialize $[x, d]$
  for $i = 0 \dots n$ **do**
    sample $v \sim \mathcal{N}(v \,|\, x + d\varepsilon\nabla_x \log p(x), 2\varepsilon)$
    evaluate $d' = -d \cdot \text{sign}\Big(\nabla_x \log p(x)^T \nabla_v \log p(v)\Big)$
    evaluate $P = \min\left\{1, \frac{p(v)\mathcal{N}(x \,|\, v + d'\varepsilon\nabla_v \log p(v), 2\varepsilon)}{p(x)\mathcal{N}(v \,|\, x + d\varepsilon\nabla_x \log p(x), 2\varepsilon)}\right\}$
    accept $[x, d] \leftarrow \begin{cases} [v, d'], \text{ with probability } P \\ [x, d], \text{ with probability } (1 - P) \end{cases}$
    $d \leftarrow -d$
    $x_i \leftarrow x$
  end for
output samples $\{x_0, \dots, x_n\}$

C.4. Irr-NICE-MC

The irreversible analog of A-NICE-MC (Song et al., 2017) is easily obtained from the original algorithm (see B.9) by composing it with two additional kernels. The first kernel $t_1(x', v', d', a' \,|\, x, v, d, a)$ operates by changing only the auxiliary variable v. That is, consider the joint distribution

$$p(x, v, d, a) = p(x)p(v)p(d)p(a \,|\, v), \quad p(v) = \mathcal{N}(v \,|\, 0, 1), \quad p(a \,|\, v) = \mathcal{N}(a \,|\, v\sqrt{1 - \alpha^2}, \alpha^2), \quad p(d) = \text{Uniform}\{-1, +1\}. \tag{196}$$

And the involution $f_1(x, v, d, a) = [x, a, d, v]$ that just swap a and v. Note that the acceptance probability

$$P_1 = \left\{1, \frac{p(x)\mathcal{N}(a \,|\, 0, 1)\mathcal{N}(v \,|\, a\sqrt{1 - \alpha^2}, \alpha^2)}{p(x)\mathcal{N}(v \,|\, 0, 1)\mathcal{N}(a \,|\, v\sqrt{1 - \alpha^2}, \alpha^2)}\right\} = 1. \tag{197}$$

The second kernel $t_2(x', v', d' \,|\, x, v, d)$ is equivalent to the A-NICE-MC kernel with only difference that we do not resample d at each step. The joint distribution of this kernel is

$$p(x, v, d) = p(x)p(v)p(d), \quad p(v) = \mathcal{N}(v \,|\, 0, 1), \quad p(d) = \text{Uniform}\{-1, +1\}. \tag{198}$$

And the involutive map is

$$f_2(x, v, d) = [T_d(x, v), -d], \quad T_{d=+1} = T, \;\; T_{d=-1} = T^{-1}. \tag{199}$$

The last kernel $t_3(x', v', d' \,|\, x, v, d)$ operates on the same joint distribution $p(x, v, d)$, and just negate the directional variable d with involution $f_3(x, v, d) = [x, v, -d]$. Combining all three kernels, we obtain irreversible modification of A-NICE-MC. See pseudo-code in Algorithm 20.

Algorithm 20 Irr-NICE-MC

input target density $p(x, v) = p(x)\mathcal{N}(v \,|\, 0, 1)$, NICE-proposal $T(x, v)$ and $T^{-1}(x, v)$
  initialize $[x, v, d]$
  for $i = 0 \dots n$ **do**
    sample $\widehat{v} \sim \mathcal{N}(\widehat{v} \,|\, v\sqrt{1-\alpha^2}, \alpha^2)$
    propose $[x', v'] = T_d(x, \widehat{v})$, where $T_{d=+1} = T$ and $T_{d=-1} = T^{-1}$
    evaluate $P = \min\left\{1, \frac{p(x', v')}{p(x, v)}\right\}$
    accept $[x, v, d] \leftarrow \begin{cases} [x', v', -d], \text{ with probability } P \\ [x, \widehat{v}, d], \text{ with probability } (1 - P) \end{cases}$
    $d \leftarrow -d$
    $x_i \leftarrow x$
  end for
output samples $\{x_0, \dots, x_n\}$