

Hamiltonian Assisted Metropolis Sampling

Zexi Song¹ & Zhiqiang Tan¹

May 19, 2020

Abstract. Various Markov chain Monte Carlo (MCMC) methods are studied to improve upon random walk Metropolis sampling, for simulation from complex distributions. Examples include Metropolis-adjusted Langevin algorithms, Hamiltonian Monte Carlo, and other recent algorithms related to underdamped Langevin dynamics. We propose a broad class of irreversible sampling algorithms, called Hamiltonian assisted Metropolis sampling (HAMS), and develop two specific algorithms with appropriate tuning and preconditioning strategies. Our HAMS algorithms are designed to achieve two distinctive properties, while using an augmented target density with momentum as an auxiliary variable. One is generalized detailed balance, which induces an irreversible exploration of the target. The other is a rejection-free property, which allows our algorithms to perform satisfactorily with relatively large step sizes. Furthermore, we formulate a framework of generalized Metropolis–Hastings sampling, which not only highlights our construction of HAMS at a more abstract level, but also facilitates possible further development of irreversible MCMC algorithms. We present several numerical experiments, where the proposed algorithms are found to consistently yield superior results among existing ones.

Key words and phrases. Auxiliary variables; Detailed balance; Hamiltonian Monte Carlo; Markov chain Monte Carlo; Metropolis-adjusted Langevin algorithms; Metropolis–Hastings sampling; Underdamped Langevin dynamics.

¹Department of Statistics, Rutgers University. Address: 110 Frelinghuysen Road, Piscataway, NJ 08854.
E-mails: zexisong@stat.rutgers.edu, ztan@stat.rutgers.edu.

1 Introduction

In various statistical applications, it is desired to generate observations from a probability density $\pi(x)$, referred to as the target distribution. The density function $\pi(x)$ is often defined such that an unnormalized density function $\tilde{\pi}(x) \propto \pi(x)$ can be readily evaluated, but the normalizing constant $\int \tilde{\pi}(x) dx$ is intractable due to high-dimensional integration. A prototypical example is posterior sampling for Bayesian analysis, where the product of the likelihood and prior is an unnormalized posterior density. For such sampling tasks, a useful methodology is Markov chain Monte Carlo (MCMC), where a Markov chain is simulated such that the associated stationary distribution coincides with the target $\pi(x)$. Under ergodic conditions, observations from the Markov chain can be considered an approximate sample from $\pi(x)$. See for example Liu (2001) and Brooks et al. (2011).

One of the main workhorses in MCMC is Metropolis–Hastings sampling (Metropolis et al., 1953; Hastings, 1970). Given current variable x_0 , the Metropolis–Hastings algorithm generates x^* from a proposal density $x^* \sim Q(x^*|x_0)$, and then accepts $x_1 = x^*$ as the next variable with probability

$$\rho(x^*|x_0) = \min \left\{ 1, \frac{\pi(x^*)Q(x_0|x^*)}{\pi(x_0)Q(x^*|x_0)} \right\}, \quad (1)$$

or rejects x^* and set $x_1 = x_0$, where $\pi(x^*)/\pi(x_0)$ can be evaluated as $\tilde{\pi}(x^*)/\tilde{\pi}(x_0)$ without requiring the normalizing constant. The update from x_0 to x_1 defines a Markov transition $K(x_1|x_0)$, depending on both the proposal density and the acceptance-rejection step, such that reversibility is satisfied: $\pi(x_0)K(x_1|x_0) = \pi(x_1)K(x_0|x_1)$. This condition is also called detailed balance, originally in physics. As a result, the Markov chain defined by the transition kernel K is reversible and admits $\pi(x)$ as a stationary distribution.

The Metropolis–Hastings algorithm is flexible in allowing various choices of the proposal density Q . A simple choice, known as random walk Metropolis (RWM), is to add a Gaussian noise to x_0 for generating x^* . However, RWM may perform poorly for sampling from complex distributions. To tackle this issue, various MCMC methods are developed by exploiting gradient information in the target density $\pi(x)$. A common approach is to use discretizations of physics-based continuous dynamics as proposal schemes, while staying

within the framework of Metropolis–Hastings sampling. One group of algorithms include preconditioned Metropolis-adjusted Langevin algorithm (pMALA) (Besag, 1994; Roberts and Tweedie, 1996) and preconditioned Crank–Nicolson Langevin (pCNL) (Cotter et al., 2013), related to (overdamped) Langevin diffusion. Another popular algorithm is Hamiltonian Monte Carlo (HMC), which introduces a momentum variable and uses a leapfrog discretization of the deterministic Hamiltonian dynamics as the proposal scheme combined with momentum resampling (Duane et al., 1987; Neal, 2011). A subtle point is that the momentum can be artificially negated at the end of leapfrog to ensure reversibility.

There are also various MCMC methods, designed by simulating irreversible Markov chains which converge to the target distribution (sometimes with auxiliary variables). One group of algorithms include guided Monte Carlo (GMC) (Horowitz, 1991; Ottobre et al., 2016) and the underdamped Langevin sampler (UDL) (Bussi and Parrinello, 2007), related to the underdamped Langevin dynamics. Another group of algorithms includes irreversible MALA (Ma et al., 2018) and non-reversible parallel tempering (Syed et al., 2019), related to lifting with a binary auxiliary variable (Gustafson, 1998; Vucelja, 2016). A third group of algorithms include the bouncy particle (Bouchard-Cote et al., 2018) and Zig-Zag samplers (Bierkens et al., 2019), using Poisson jump processes.

The contribution of this article can be summarized as follows. First, we propose a broad class of irreversible sampling algorithms, called Hamiltonian assisted Metropolis sampling (HAMS), and develop two specific algorithms, HAMS-A/B, with appropriate tuning and preconditioning strategies. Our HAMS algorithms use an augmented target density (corresponding to a Hamiltonian) with momentum as an auxiliary variable. Each iteration of HAMS consists of a proposal step depending on the gradient of the Hamiltonian, and an acceptance-rejection step using an acceptance probability different from the usual formula (1). The two steps are designed to achieve generalized detailed balance and a rejection-free property discussed below. Second, we formulate a framework of generalized Metropolis–Hastings sampling, which not only highlights our construction of HAMS as a special case, but also facilitates possible further development of irreversible MCMC algorithms. Third, we present several numerical experiments, where the proposed algorithms are found to

consistently yield superior results among existing ones.

Compared with existing algorithms, there are two important properties which are *simultaneously* satisfied by our HAMS algorithms. The first is generalized detailed balance (or generalized reversibility), where the backward transition is related to the forward transition after negating the momentum. This condition is known in the study of continuous dynamics in physics (Gardiner, 1997), but seems to receive insufficient treatment in the MCMC literature, where the acceptance-rejection step is also crucial for proper sampling from a target distribution. By generalized detailed balance, the momentum can be accepted without sign negation, which induces an irreversible exploration of the target. Second, our algorithms satisfy a rejection-free property, that is, the proposal is always accepted at each iteration, in the case where the target distribution is standard normal. By preconditioning, the rejection-free property can also be achieved when the target distribution is normal with a pre-specified variance Σ . A similar motivation can be found in the construction of pCNL algorithm (Cotter et al., 2013). From our experiments, this property allows our algorithms to perform satisfactorily with relatively large step sizes.

Notation. Assume that a target density $\pi(x)$ is defined on \mathbb{R}^k . The potential energy function $U(x)$ is defined such that $\pi(x) \propto \exp\{-U(x)\}$ as in physics. Denote the gradient of $U(x)$ as $\nabla U(x)$. The (multivariate) normal distribution with mean μ and variance V is denoted as $\mathcal{N}(\mu, V)$, and the density function as $\mathcal{N}(\cdot | \mu, V)$. Whenever possible, we treat a probability distribution and its density function interchangeably. Write $\mathbf{0}$ for a vector or matrix with all 0 entries, and I for an identity matrix of appropriate dimensions.

2 Related methods

We describe several MCMC algorithms, related to our work, for sampling from a target distribution $\pi(x)$. Throughout, we write the current variable as x_0 , a proposal as x^* , and the next variable as x_1 after the acceptance-rejection step. Denote as Σ a constant variance matrix used as an approximation to the variance of the target $\pi(x)$.

Random walk Metropolis sampling generates a proposal x^* by directly adding a Gaus-

sian noise to x_0 and then performs acceptance or rejection.

Random walk Metropolis sampling (RWM).

- Generate $x^* = x_0 + \epsilon Z$, where $Z \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $\epsilon > 0$ is a tunable step size.
- Set $x_1 = x^*$ with acceptance probability $\rho(x^*|x_0) = \min(1, \pi(x^*)/\pi(x_0))$ by (1), or set $x_1 = x_0$ with the remaining probability.

RWM does not exploit gradient information, and may be slow in exploring the target $\pi(x)$. On the other hand, RWM is operationally low-cost, without gradient evaluation.

The preconditioned Metropolis-adjusted Langevin algorithm (pMALA) generates a proposal x^* by moving along the gradient from current x_0 (Roberts and Tweedie, 1996). Hence pMALA is more directed and encourages exploration to high density regions.

Preconditioned Metropolis-adjusted Langevin algorithm (pMALA).

- Generate $x^* = x_0 - \frac{\epsilon^2}{2} \Sigma \nabla U(x_0) + \epsilon Z$, where $Z \sim \mathcal{N}(\mathbf{0}, \Sigma)$ and $\epsilon > 0$ is a step size.
- Set $x_1 = x^*$ with probability (1), where $Q(x^*|x_0) = \mathcal{N}(x^*|x_0 - \frac{\epsilon^2}{2} \Sigma \nabla U(x_0), \epsilon^2 \Sigma)$, or set $x_1 = x_0$ with the remaining probability.

The preconditioned Crank-Nicolson Langevin (pCNL) algorithm is originally designed for posterior sampling with a latent Gaussian field model (Cotter et al., 2013). The target density is $\pi(x) \propto \exp\{-U(x)\} \propto \exp\{\ell(x)\}\mathcal{N}(x|\mathbf{0}, C)$, a product of a likelihood function and a normal prior with variance C . For easy comparison, we use a parameterization in terms of the step size ϵ and the potential gradient, $\nabla U(x) = -\nabla \ell(x) + C^{-1}x$.

Preconditioned Crank-Nicolson Langevin (pCNL).

- Sample $Z \sim \mathcal{N}(\mathbf{0}, C)$ and compute

$$\begin{aligned} x^* &= \sqrt{1 - \epsilon^2}x_0 + \frac{\epsilon^2}{1 + \sqrt{1 - \epsilon^2}}C\nabla\ell(x_0) + \epsilon Z \\ &= x_0 - \frac{\epsilon^2}{1 + \sqrt{1 - \epsilon^2}}C\nabla U(x_0) + \epsilon Z. \end{aligned} \tag{2}$$

- Set $x_1 = x^*$ with probability (1), where $Q(x^*|x_0) = \mathcal{N}(x^*|x_0 - \frac{\epsilon^2}{1 + \sqrt{1 - \epsilon^2}}C\nabla U(x_0), \epsilon^2 C)$, or set $x_1 = x_0$ with the remaining probability.

It is interesting to compare pMALA and pCNL. On one hand, pCNL is close to pMALA with the preconditioning matrix Σ chosen to be C , as the step size $\epsilon \rightarrow 0$ and hence $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} \rightarrow \frac{\epsilon^2}{2}$ in (2). On the other hand, as ϵ stays away from 0, the coefficient $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}$ associated with the potential gradient in pCNL can differ considerably from $\frac{\epsilon^2}{2}$ in pMALA. As discussed in Cotter et al. (2013), a simple advantage of this difference is that when the likelihood gradient $\nabla\ell$ is dropped, the resulting proposal from (2) becomes $x^* = \sqrt{1-\epsilon^2}x_0 + \epsilon Z$, which is invariant and reversible with respect to the prior $\mathcal{N}(\mathbf{0}, C)$. In this case, the proposal x^* is accepted with probability 1 in pCNL, but not in pMALA. To achieve such a rejection-free property also plays an important role in our work.

From the preceding discussion, it seems straightforward to define a modified pMALA algorithm, by replacing the update coefficient $\frac{\epsilon^2}{2}$ with $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}$ in pMALA. Equivalently, this algorithm can also be obtained from pCNL, by replacing the prior variance C by a general preconditioning matrix Σ , which can be specified as an approximation to the variance of the target distribution $\pi(x)$, instead of being fixed as the prior variance C . As a result, the modified pMALA algorithm is rejection-free (i.e., the proposal x^* is always accepted) when the target density is $\mathcal{N}(\mathbf{0}, \Sigma)$. To our knowledge, such an extension of pMALA and pCNL appears not explicitly studied before. In Section 3.5, we obtain the modified pMALA algorithm as a boundary case of the proposed HAMS algorithms.

Modified preconditioned Metropolis-adjusted Langevin algorithm (pMALA^{}).*

- Generate $x^* = x_0 - \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} \Sigma \nabla U(x_0) + \epsilon Z$, where $Z \sim \mathcal{N}(\mathbf{0}, \Sigma)$.
- Set $x_1 = x^*$ with probability (1), where $Q(x^*|x_0) = \mathcal{N}(x^*|x_0 - \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}} \Sigma \nabla U(x_0), \epsilon^2 \Sigma)$, or set $x_1 = x_0$ with the remaining probability.

We also point out that modified pMALA is distinct from a related gradient-based algorithm in Titsias and Papaspiliopoulos (2018), which is proposed in the context of posterior sampling with the target density $\pi(x) \propto \exp\{-U(x)\} \propto \exp\{\ell(x)\}\mathcal{N}(x|\mathbf{0}, C)$. The associated proposal scheme (without preconditioning) can be written as

$$x^* = \frac{2}{\delta} \tilde{C} x_0 + \tilde{C} \nabla \ell(x_0) + Z = x_0 - \tilde{C} \nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, \frac{2}{\delta} \tilde{C}^2 + \tilde{C}), \quad (3)$$

where $\tilde{C} = (\frac{2}{\delta} I + C^{-1})^{-1}$ and $\nabla U(x_0) = -\nabla \ell(x_0) + C^{-1}x_0$. When the prior variance C is

an identity matrix (i.e., $C = I$), the proposal scheme (3) reduces to

$$x^* = x_0 - \frac{\delta}{2 + \delta} \nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, \frac{\delta(\delta+4)}{(\delta+2)^2} I),$$

which is equivalent to the proposal scheme in pCNL and in modified pMALA with $\Sigma = I$, after matching $\epsilon^2 = \frac{\delta(\delta+4)}{(\delta+2)^2}$. However, except for this coincidence, the algorithm of Titsias and Papaspiliopoulos (2018) based on (3) as well as its preconditioned version in general differ from modified pMALA above. In fact, modified pMALA can also be derived using auxiliary variables, but invoking a different Taylor expansion to approximate the target density from Titsias and Papaspiliopoulos (2018). See the Supplement Section I for further discussion on auxiliary variables and second-order schemes.

The following methods require augmenting the sample space to include a momentum variable $u \in \mathbb{R}^k$, which is assumed to be normally distributed, $u \sim \mathcal{N}(\mathbf{0}, M)$. The variance M is also called a mass matrix, and the quantity $u^T M^{-1} u / 2$ represents the kinetic energy in physics. The joint target density of (x, u) becomes

$$\pi(x, u) \propto \exp\{-H(x, u)\} = \exp\{-U(x) - \frac{1}{2} u^T M^{-1} u\}, \quad (4)$$

where $H(x, u) = U(x) + \frac{1}{2} u^T M^{-1} u$, called a total energy or Hamiltonian. For sampling from an augmented target distribution $\pi(x, u)$, Hamiltonian Monte Carlo generates a proposal by first redrawing a momentum variable and then performing a series of deterministic updates, based on molecular dynamics (MD) simulations such that the Hamiltonian $H(x, u)$ is approximately preserved (Duane et al., 1987; Neal, 2011).

Hamiltonian Monte Carlo (HMC).

- Sample $u^* \sim \mathcal{N}(\mathbf{0}, M)$, reset $u_0 = u^*$, and set $x^* = x_0$.
- For i from 1 to $nleap$, repeat:

$$u^* \leftarrow u^* - \frac{\epsilon}{2} \nabla U(x^*), \quad x^* \leftarrow x^* + \epsilon M^{-1} u^*, \quad u^* \leftarrow u^* - \frac{\epsilon}{2} \nabla U(x^*).$$
- Set $(x_1, u_1) = (x^*, u^*)$ with probability $\min(1, \exp(H(x_0, u_0) - H(x^*, u^*)))$
or set $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability.

The steps within the for loop are called leapfrog updates, which provide an accurate discretization of the Hamiltonian dynamics, defined as a system of differential equations

by Newton's laws of motion such that the Hamiltonian $H(x, u)$ is preserved over time. Although the update of u can be ignored, the acceptance-rejection step above is stated such that the update of (x, u) matches UDL and GMC later with $c = 1$, if momentum were not resampled. For HMC, both the step size ϵ and the number of leapfrog steps n_{leap} need to be tuned. For automated tuning, it seems popular to use the No-U-Turn Sampler (NUTS) proposed by Hoffman and Gelman (2014). Nevertheless, HMC often requires a large number of leapfrog steps which is computationally costly.

An important extension of the Hamiltonian dynamics is Langevin dynamics, which can be defined as a system of stochastic differential equations,

$$dx_t = u_t dt, \quad du_t = -\eta dx_t - \nabla U(x_t) dt + \sqrt{2\eta} dW_t, \quad (5)$$

where $\eta > 0$ is a friction coefficient and W_t is the standard Brownian process. In the case of $\eta \rightarrow 0$, the Langevin dynamics reduces to the deterministic Hamiltonian dynamics, $dx_t = u_t dt$ and $du_t = -\nabla U(x_t) dt$. In the high-friction limit (i.e., large η), the overdamped Langevin diffusion process is obtained: $dx_t = -\eta^{-1} \nabla U(x_t) dt + \sqrt{2\eta^{-1}} dW_t$. Hence (5) is also called underdamped Langevin dynamics. Although Langevin dynamics has long been used in molecular simulations (e.g., van Gunsteren and Berendsen, 1982), there is extensive and growing research related to Langevin dynamics in physics and chemistry (e.g., Horowitz, 1991; Scemama et al., 2006; Bussi and Parrinello, 2007; Goga et al., 2012; Grønbech-Jensen and Farago, 2013, 2020) and machine learning and statistics (e.g., Ottobre et al., 2016; Cheng et al., 2018; Dalalyan and Riou-Durand, 2018). In particular, the Metropolized version of the algorithm in Bussi and Parrinello (2007) can be described as follows, to accommodate an acceptance-rejection step.

Underdamped Langevin sampling (UDL).

- Sample $Z_1, Z_2 \sim \mathcal{N}(\mathbf{0}, M)$ independently, and compute

$$u^+ = \sqrt{c}u_0 + \sqrt{1-c}Z_1,$$

$$\tilde{u} = u^+ - \frac{\epsilon}{2} \nabla U(x_0), \quad x^* = x_0 + \epsilon M^{-1} \tilde{u}, \quad u^- = \tilde{u} - \frac{\epsilon}{2} \nabla U(x^*),$$

$$u^* = \sqrt{c}u^- + \sqrt{1-c}Z_2,$$

where $0 \leq c \leq 1$ is a tuning parameter and can be interpreted as $c = e^{-\eta\epsilon/2}$.

- Set $(x_1, u_1) = (x^*, u^*)$ with probability $\min(1, \exp(H(x_0, u^+) - H(x^*, u^-)))$
or set $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability.

There are several interesting features in UDL. First, the proposal scheme in UDL contains a (deterministic) leapfrog update, which is sandwiched by two random updates of the momentum. Notably, the current momentum u_0 is partially refreshed at the beginning, where the amount of ‘‘carryover’’ is controlled by the parameter c . At the two extremes, $c = 0$ or 1 , UDL recovers pMALA or Metropolized leapfrog respectively. When $c = 0$, the first updated momentum $u^+ = Z_1$ is independent of u_0 and the final updated momentum $u^* = Z_2$ can be ignored. In this case, UDL reduces to HMC with one leapfrog step (after redrawing the momentum) and hence is equivalent to pMALA as discussed in Neal (2011). When $c = 1$, UDL generates a proposal by one leapfrog update and then accept or reject (with u_0 flipped) based on the change in the Hamiltonian.

Second, the proposal scheme in UDL is derived in Bussi and Parrinello (2007) by a particular choice of operator splitting in discretizing the Langevin dynamics (5). Compared with other possible choices, the UDL proposal scheme is shown to satisfy a generalized formulation of detailed balance. However, as discussed later in Section 4, whether a sampling algorithm leaves a target distribution invariant depends also on how acceptance or rejection is executed. While Bussi and Parrinello (2007) only mentioned that acceptance-rejection can be performed similarly as in Scemama et al. (2006), the acceptance-rejection step above is explicitly added by our understanding. In the Appendix, we verify the validity of the UDL algorithm in leaving the target augmented density $\pi(x, u)$ invariant, using our proposed framework of generalized Metropolis–Hastings sampling.

Third, both the two momentum updates are in the form of an order-1 autoregressive process, which leaves the momentum distribution invariant: if $u_0 \sim \mathcal{N}(\mathbf{0}, M)$ then $u^+ \sim \mathcal{N}(\mathbf{0}, M)$ and, similarly, if $u^- \sim \mathcal{N}(\mathbf{0}, M)$ then $u^* \sim \mathcal{N}(\mathbf{0}, M)$. As discussed in Bussi and Parrinello (2007), such updates using two independent noise vectors are exploited to achieve generalized detailed balance. In fact, it is instructive to compare UDL with a related algorithm in Horowitz (1991), which uses only one noise vector per iteration as described below. To our knowledge, it seems difficult to show that generalized detailed balance is

satisfied by this algorithm, although invariance with respect to $\pi(x, u)$ is valid because each iteration is a composition of two steps, first $(x_0, u_0) \rightarrow (x_0, u^+)$ and then $(x_0, u^+) \rightarrow (x_1, u_1)$ by Metropolized leapfrog, and each step leaves the target $\pi(x, u)$ invariant.

Guided Monte Carlo (GMC).

- Sample $Z_1 \sim \mathcal{N}(\mathbf{0}, M)$, and compute

$$u^+ = \sqrt{c}u_0 + \sqrt{1-c}Z_1,$$

$$\tilde{u} = u^+ - \frac{\epsilon}{2}\nabla U(x_0), \quad x^* = x_0 + \epsilon M^{-1}\tilde{u}, \quad u^- = \tilde{u} - \frac{\epsilon}{2}\nabla U(x^*).$$

- Set $(x_1, u_1) = (x^*, u^-)$ with probability $\min(1, \exp(H(x_0, u^+) - H(x^*, u^-)))$
or set $(x_1, u_1) = (x_0, -u^+)$ with the remaining probability.

Another interesting method is the irreversible MALA algorithm in Ma et al. (2018). Compared with our method using an augmented density with momentum as an auxiliary variable, this method relies on a binary auxiliary variable to facilitate irreversible sampling, while using discretizations of continuous dynamics in the original variable x as proposal schemes. See Section 4 and Supplement Section III for further discussion.

3 Proposed Methods

We develop our methods in several steps. We first construct proposal schemes using gradient information, then introduce modifications to derive a class of generalized reversible algorithms HAMS, and finally study two specific algorithms, HAMS-A/B, and propose tuning and preconditioning strategies. To focus on main ideas, consider the augmented target density (4) with momentum variance $M = I$, that is,

$$\pi(x, u) \propto \exp(-H(x, u)) = \exp(-U(x) - u^T u / 2), \quad (6)$$

until Section 3.6 to discuss preconditioning. The proposed algorithms are then placed in a more abstract framework of generalized Metropolis–Hastings sampling in Section 4.

3.1 Construction of Hamiltonian proposals

We provide a simple, broad class of proposal distributions, which are suitable for use in standard Metropolis–Hastings sampling from an augmented density $\pi(x, u)$. These proposal schemes will be modified later for developing irreversible algorithms.

Given current variables (x_0, u_0) , a proposal (x^*, u^*) can be generated as

$$\begin{pmatrix} x^* \\ u^* \end{pmatrix} = \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - A \begin{pmatrix} \nabla U(x_0) \\ u_0 \end{pmatrix} + \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}, \quad \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, 2A - A^2), \quad (7)$$

where A is a $(2k) \times (2k)$ symmetric positive semi-definite (PSD) matrix and $Z_1, Z_2 \in \mathbb{R}^k$ are Gaussian noises independent of (x_0, u_0) , with k the dimension of x and that of u . We require $\mathbf{0} \leq A \leq 2I$, where inequalities between matrices are in the PSD sense. This ensures that $2A - A^2$ is also symmetric positive semi-definite, although allowed to be singular. The update in (7) takes a gradient step from the current variables (x_0, u_0) and then injects Gaussian noises (Z_1, Z_2) . Hence the proposal scheme (7) is similar to that in pMALA. However, (7) is applied to (x, u) jointly, instead of x alone.

The proposal scheme (7) can be derived through an auxiliary variable argument related to Titsias and Papaspiliopoulos (2018), while incorporating an over-relaxation technique as in Adler (1981) and Neal (1998). See Supplement Section I for details.

Another important motivation for the proposal scheme (7) is that Metropolis–Hastings sampling using (7) becomes rejection-free, while generating correlated draws, in the canonical case where the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, that is, $U(x) = -x^T x / 2$ with the gradient $\nabla U(x) = x$. In fact, the proposal scheme (7) in this case gives

$$\begin{pmatrix} x^* \\ u^* \end{pmatrix} = (I - A) \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} + \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix}, \quad \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} \sim \mathcal{N}(\mathbf{0}, 2A - A^2). \quad (8)$$

The update from (x_0, u_0) to (x^*, u^*) in (8) can be seen to define an order-1 vector autoregressive process, VAR(1), which is reversible and admits $\mathcal{N}(\mathbf{0}, I)$ as a stationary distribution due to symmetry of A (Osawa, 1988). The stationary distribution can be easily verified: if $(x_0, u_0) \sim \mathcal{N}(\mathbf{0}, I)$, then (x^*, u^*) is normal and the mean and variance are

$$\mathbb{E}[(x^{*\top}, u^{*\top})^\top] = \mathbf{0}, \quad \text{Var}[(x^{*\top}, u^{*\top})^\top] = (I - A)(I - A)^\top + 2A - A^2 = I. \quad (9)$$

The reversibility of (8) with stationary distribution $\mathcal{N}(\mathbf{0}, I)$ implies that when the target density $\pi(x, u)$ is $\mathcal{N}(\mathbf{0}, I)$, Metropolis–Hastings sampling using the proposal scheme (8) is rejection-free: the draws (x^*, u^*) are always accepted. This can also be shown by using the proposal density, $Q(x^*, u^* | x_0, u_0) = \mathcal{N}(x^*, u^* | (I - A)(x^T, u^T)^T, 2A - A^2)$, and directly verifying that the acceptance probability (1) with x replaced by (x, u) reduces to 1.

Our discussion focuses on the proposal scheme (7) for a Hamiltonian with momentum $u \sim \mathcal{N}(\mathbf{0}, I)$ and the VAR(1) representation (8) in the canonical case $x \sim \mathcal{N}(\mathbf{0}, I)$, related to the normal approximation (S2) with identity variance I in the auxiliary variable derivation. The development can be readily extended to handle general variance matrices, for a momentum distribution $u \sim \mathcal{N}(\mathbf{0}, M)$ and a normal approximation to $\pi(x)$ with variance matrix Σ . Nevertheless, as discussed in Section 3.6, it is convenient to set $M = I$ and if an approximation of $\text{Var}(x)$ is available, apply linear transformation to x such that the target density $\pi(x)$ can be roughly aligned with an identity variance $\Sigma = I$.

3.2 HAMS: a class of generalized reversible algorithms

In this and subsequent sections, we exploit the class of proposals (7) with general choices of A matrix, to first derive a broad class of generalized reversible algorithms HAMS and then study two specific algorithms HAMS-A/B more elaborately.

For simplicity, consider the following form of A matrix in (7),

$$A = \begin{pmatrix} a_1 I & a_2 I \\ a_2 I & a_3 I \end{pmatrix} \quad (10)$$

with each I a $k \times k$ identity matrix and a_1, a_2, a_3 scalar coefficients. We require $a_1, a_3 \geq 0, a_1 + a_3 \leq 2$ and $a_1 a_3 \geq a_2^2$, which is sufficient for the constraint $\mathbf{0} \leq A \leq 2I$ (in the PSD sense). Substituting this choice of A into (7) yields

$$x^* = x_0 - a_1 \nabla U(x_0) - a_2 u_0 + Z_1, \quad (11)$$

$$u^* = u_0 - a_2 \nabla U(x_0) - a_3 u_0 + Z_2, \quad (12)$$

where $(Z_1^T, Z_2^T)^T \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$ as before. As discussed in Section 3.1, standard Metropolis-Hastings sampling using this proposal scheme is rejection-free, that is, (x^*, u^*)

is always accepted, when the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$.

Modification for generalized reversibility. We first make a modification to (11)–(12) by replacing the momentum u_0 with $-u_0$. Although a formal justification is to achieve generalized reversibility as shown in Proposition 1, we give a heuristic motivation by noticing that $a_2 u_0$ in (11) and $a_2 \nabla U(x_0)$ in (12) are of the same sign. In contrast, for the discretization of Hamiltonian dynamics using Euler’s method:

$$x^* = x_0 + \epsilon u_0, \quad u^* = u_0 - \epsilon \nabla U(x_0),$$

the momentum u_0 and gradient $\nabla U(x_0)$ are of the opposite signs. This discrepancy can be resolved by setting $u_0 \mapsto -u_0$, for which (11)–(12) become

$$x^* = x_0 - a_1 \nabla U(x_0) + a_2 u_0 + Z_1, \quad (13)$$

$$u^* = -u_0 - a_2 \nabla U(x_0) + a_3 u_0 + Z_2, \quad (14)$$

where $(Z_1^T, Z_2^T)^T \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$ as before.

The proposal (x^*, u^*) in (13)–(14) can be accepted or rejected, similarly as in standard Metropolis–Hastings sampling but using a different acceptance probability, which we derive through generalized detailed balance. Rewrite the proposal scheme (13)–(14) as

$$\tilde{Z}_1 = Z_1 - a_1 \nabla U(x_0) + a_2 u_0, \quad \tilde{Z}_2 = Z_2 - a_2 \nabla U(x_0) + a_3 u_0, \quad (15)$$

$$x^* = x_0 + \tilde{Z}_1, \quad u^* = -u_0 + \tilde{Z}_2. \quad (16)$$

Equations (16)–(15) determine a forward transition from (x_0, u_0) to (x^*, u^*) , depending on noises (Z_1, Z_2) . To construct a backward transition, define new noises

$$Z_1^* = \tilde{Z}_1 - a_1 \nabla U(x^*) - a_2 u^*, \quad Z_2^* = \tilde{Z}_2 - a_2 \nabla U(x^*) - a_3 u^*. \quad (17)$$

Then (17) and (16) can be equivalently rearranged to

$$-\tilde{Z}_1 = -Z_1^* - a_1 \nabla U(x^*) + a_2(-u^*), \quad -\tilde{Z}_2 = -Z_2^* - a_2 \nabla U(x^*) + a_3(-u^*), \quad (18)$$

$$x_0 = x^* + (-\tilde{Z}_1), \quad -u_0 = u^* + (-\tilde{Z}_2). \quad (19)$$

Importantly, equations (18)–(19) can be seen to correspond to the *same* mapping as (15)–(16), but applied from $(x^*, -u^*)$ to $(x_0, -u_0)$ using the new noises $(-Z_1^*, -Z_2^*)$. In other

words, (18)–(19) are obtained from (15)–(16) by replacing (x_0, u_0) , (x^*, u^*) , and (Z_1, Z_2) with $(x^*, -u^*)$, $(x_0, -u_0)$, and $(-Z_1^*, -Z_2^*)$ respectively.

From the preceding discussion, the forward and backward transitions of the proposals in (15)–(16) and (18)–(19) can be illustrated as

$$\begin{pmatrix} x_0 \\ u_0 \end{pmatrix} \xrightarrow{(Z_1, Z_2)} \begin{pmatrix} x^* \\ u^* \end{pmatrix}, \quad \begin{pmatrix} x^* \\ -u^* \end{pmatrix} \xrightarrow{-(Z_1^*, Z_2^*)} \begin{pmatrix} x_0 \\ -u_0 \end{pmatrix}, \quad (20)$$

where the two arrows denote the *same* mapping, depending on (Z_1, Z_2) or $-(Z_1^*, Z_2^*)$. For $(Z_1^T, Z_2^T)^T \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$, the proposal density from (x_0, u_0) to (x^*, u^*) is

$$Q(x^*, u^* | x_0, u_0) = \mathcal{N}(Z_1, Z_2 | \mathbf{0}, 2A - A^2).$$

Moreover, evaluation of the *same* proposal density from $(x^*, -u^*)$ to $(x_0, -u_0)$ gives

$$Q(x_0, -u_0 | x^*, -u^*) = \mathcal{N}(-(Z_1^*, Z_2^*) | \mathbf{0}, 2A - A^2),$$

because the transition from $(x^*, -u^*)$ to $(x_0, -u_0)$ is determined by the same mapping as (x_0, u_0) to (x^*, u^*) , only with the noises $(-Z_1^*, -Z_2^*)$ used instead of (Z_1, Z_2) .

By mimicking (and extending) the standard Metropolis–Hastings probability, we set $(x_1, u_1) = (x^*, u^*)$ with the acceptance probability

$$\rho(x^*, u^* | x_0, u_0) = \min \left(1, \frac{\pi(x^*, -u^*) Q(x_0, -u_0 | x^*, -u^*)}{\pi(x_0, u_0) Q(x^*, u^* | x_0, u_0)} \right), \quad (21)$$

or set $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability. Due to the evenness of mean-zero normal distributions, the probability (21) can be calculated as

$$\rho(x^*, u^* | x_0, u_0) = \min \left(1, \frac{\exp \left\{ -H(x^*, u^*) - \frac{1}{2} \mathbf{Z}^{*\top} (2A - A^2)^{-1} \mathbf{Z}^* \right\}}{\exp \left\{ -H(x_0, u_0) - \frac{1}{2} \mathbf{Z}^{\top} (2A - A^2)^{-1} \mathbf{Z} \right\}} \right), \quad (22)$$

where $\mathbf{Z} = (Z_1^T, Z_2^T)^T$ and $\mathbf{Z}^* = (Z_1^{*\top}, Z_2^{*\top})^T$. Note that $u_1 = u^*$ upon acceptance, but $u_1 = -u_0$ in the case of rejection. The resulting transition from (x_0, u_0) to (x_1, u_1) can be shown to satisfy generalized detailed balance.

Proposition 1 *For an augmented density $\pi(x, u)$ in (6), let $K_0(x_1, u_1 | x_0, u_0)$ be the transition kernel from (x_0, u_0) to (x_1, u_1) , defined by the proposal scheme (15)–(16) and the acceptance probability (21). Then generalized detailed balance holds for $x_1 \neq x_0$:*

$$\pi(x_0, u_0) K_0(x_1, u_1 | x_0, u_0) = \pi(x_1, -u_1) K_0(x_0, -u_0 | x_1, -u_1). \quad (23)$$

Furthermore, the augmented density $\pi(x, u)$ is a stationary distribution of the Markov chain defined by transition kernel K_0 .

Condition (23), called generalized detailed balance (or generalized reversibility), differs from detailed balance (or reversibility) in standard Metropolis–Hastings sampling because the momentum variable is negated in defining the backward transition. Accordingly, the acceptance probability (21) is called a generalized Metropolis–Hastings probability. A similar concept of detailed balance is known in connection with Fokker–Planck equations in physics (Gardiner, 1997, Section 5.3.4). The momentum is called an odd variable, for which the time-reversed variable is defined with sign negation to achieve generalized detailed balance. Such a general formulation of detailed balance is used in the derivation of underdamped Langevin sampling (Bussi and Parrinello, 2007), but overall seems to be under-appreciated in the MCMC literature. See Section 4 for a further extension.

Modification for updating momentum. To further broaden our method, we introduce another modification to the proposal scheme (15)–(16). In fact, a potential limitation of (15)–(16), compared with the popular leapfrog scheme, is that the updated momentum u^* ignores the new gradient information $\nabla U(x^*)$. To incorporate $\nabla U(x^*)$ in updating the momentum, we revise (16) with an additional term in u^* as

$$x^* = x_0 + \tilde{Z}_1, \quad u^* = -u_0 + \tilde{Z}_2 + \phi(\tilde{Z}_1 + \nabla U(x_0) - \nabla U(x^*)), \quad (24)$$

where ϕ is a (constant) tuning parameter, and $(\tilde{Z}_1, \tilde{Z}_2)$ remain the same as in (15). Moreover, the update (24) can be rearranged to

$$x_0 = x^* + (-\tilde{Z}_1), \quad -u_0 = u^* + (-\tilde{Z}_2) + \phi(-\tilde{Z}_1 + \nabla U(x^*) - \nabla U(x_0)). \quad (25)$$

With (Z_1^*, Z_2^*) still defined as (17), equations (15) and (24) and equations (18) and (25) can be seen to be determined by the *same* mapping, similarly as illustrated in (20). The forward transition is from (x_0, u_0) to (x^*, u^*) depending on (Z_1, Z_2) , whereas the backward transition is from $(x^*, -u^*)$ to $(x_0, -u_0)$ depending on $-(Z_1^*, Z_2^*)$. With the modified proposal (x^*, u^*) , the acceptance-rejection is the same as before: set $(x_1, u_1) = (x^*, u^*)$ with probability (21) or $(x_1, u_1) = (x_0, -u_0)$ with the remaining probability. Then generalized detailed balance remains valid for the transition from (x_0, u_0) and (x_1, u_1) .

Proposition 2 For an augmented density $\pi(x, u)$ in (6), let $K_\phi(x_1, u_1|x_0, u_0)$ be the transition kernel from (x_0, u_0) to (x_1, u_1) , defined by the proposal scheme (15) and (24) and the acceptance probability (21). Then generalized detailed balance holds for $x_1 \neq x_0$:

$$\pi(x_0, u_0)K_\phi(x_1, u_1|x_0, u_0) = \pi(x_1, -u_1)K_\phi(x_0, -u_0|x_1, -u_1). \quad (26)$$

Furthermore, the augmented density $\pi(x, u)$ is a stationary distribution of the Markov chain defined by transition kernel K_ϕ .

General HAMS. Using the proposal scheme and acceptance probability as in Proposition 2 leads to a class of generalized reversible MCMC algorithms, which is called Hamiltonian assisted Metropolis sampling (HAMS) and shown in Algorithm 1.

Although the modifications of the proposal scheme from (11)–(12) to (13)–(14) and then to (15) and (24) are constructed for different purposes, the resulting HAMS algorithm preserves the rejection-free property with a standard normal target density $\pi(x)$, which is satisfied by standard Metropolis–Hastings sampling with proposal scheme (11)–(12). In fact, the second modification from (16) to (24) has no effect when $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, because in this case $\tilde{Z}_1 + \nabla U(x_0) - \nabla U(x^*) = \tilde{Z}_1 + x_0 - x^* = \mathbf{0}$. The justification for the first modification is subtler. Whether rejection-free is achieved by a sampling algorithm depends on both a proposal scheme and an associated acceptance-rejection mechanism. When $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, our HAMS algorithm is rejection-free, due to the fact the proposal scheme (13)–(14) is used in conjunction with the generalized acceptance probability (21), not the standard Metropolis–Hastings probability. We provide further discussion in Section 4, where it can be seen that consideration of the rejection-free property is instrumental to a general approach for constructing generalized reversible algorithms.

Corollary 1 Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$. Then the generalized acceptance probability (21) or equivalently (22) reduces to 1, and hence (x^*, u^*) from the proposal scheme (13)–(14) is always accepted under the HAMS algorithm.

The general HAMS involves four tuning parameters ϕ, a_1, a_2, a_3 , which need to be specified for practical implementation. In the following sections, we develop more concrete

Algorithm 1: General HAMS

 Initialize x_0, u_0
for $t = 0, 1, 2, \dots, N_{iter}$ **do**

 Sample $w \sim \text{Uniform}[0, 1]$ and $(Z_1, Z_2)^\top \sim N(\mathbf{0}, 2A - A^2)$ with $A = \begin{pmatrix} a_1 I & a_2 I \\ a_2 I & a_3 I \end{pmatrix}$

$$\tilde{Z}_1 = Z_1 - a_1 \nabla U(x_t) + a_2 u_t$$

$$\tilde{Z}_2 = Z_2 - a_2 \nabla U(x_t) + a_3 u_t$$

 Propose $x^* = x_t + \tilde{Z}_1$ and $u^* = -u_t + \tilde{Z}_2 + \phi(\tilde{Z}_1 + \nabla U(x_t) - \nabla U(x^*))$

$$Z_1^* = \tilde{Z}_1 - a_1 \nabla U(x^*) - a_2 u^*$$

$$Z_2^* = \tilde{Z}_2 - a_2 \nabla U(x^*) - a_3 u^*$$

$$\rho = \exp \left\{ H(x_t, u_t) - H(x^*, u^*) + \frac{1}{2} \mathbf{Z}^\top (2A - A^2)^{-1} \mathbf{Z} - \frac{1}{2} \mathbf{Z}^{*\top} (2A - A^2)^{-1} \mathbf{Z}^* \right\}$$

if $w < \min(1, \rho)$ **then**

$$| (x_{t+1}, u_{t+1}) = (x^*, u^*) \quad \# \text{Accept}$$

else

$$| (x_{t+1}, u_{t+1}) = (x_t, -u_t) \quad \# \text{Reject}$$

end
end

versions of HAMS with a reduced number of tuning parameters. As the augmented target density is $2k$ dimensional, HAMS in general allows the noise term (Z_1, Z_2) to be drawn directly from a $2k$ dimensional Gaussian distribution. Nevertheless, there are related methods developed for simulating Langevin dynamics, using k dimensional noises at each time step (Grønbech-Jensen and Farago, 2013, 2020). We investigate HAMS which also uses only k dimensional Gaussian noises in each iteration. This requires the variance matrix $2A - A^2$ to be singular. There are two possible choices: either A itself is singular or $2I - A$ is singular, corresponding to HAMS-A and HAMS-B in Section 3.3.

3.3 HAMS-A and HAMS-B

We develop two concrete versions of HAMS with the noise variance $2A - A^2$ singular, hence using only k dimensional Gaussian noises in each iteration.

HAMS-A. First, we set A singular by taking $a_1 = a, a_3 = b$ and $a_2 = \sqrt{ab}$ in (10).

The constraints on A require that $a \geq 0, b \geq 0$ and $a + b \leq 2$. To avoid trivial cases, we also assume that $a > 0$. The noise variance becomes

$$\text{Var} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = 2A - A^2 = \begin{pmatrix} a(2-a-b)I & \sqrt{ab}(2-a-b)I \\ \sqrt{ab}(2-a-b)I & b(2-a-b)I \end{pmatrix}. \quad (27)$$

As expected, this implies that Z_1 and Z_2 are proportional: $Z_2 = \sqrt{b/a}Z_1$. By definitions (15), (24), and (17), it can be easily verified that $\tilde{Z}_2 = \sqrt{b/a}\tilde{Z}_1$ and $Z_2^* = \sqrt{b/a}Z_1^*$ as well. The proportionality between Z_1^* and Z_2^* is important, because it ensures that both forward and backward transitions, illustrated in (20), can be determined using a single noise vector, Z_1 or $-Z_1^*$. Hence the proposal density from (x_0, u_0) to (x^*, u^*) is $\mathcal{N}(Z_1|\mathbf{0}, a(2-a-b)I)$ and that from $(x^*, -u^*)$ to $(x_0, -u_0)$ is $\mathcal{N}(-Z_1^*|\mathbf{0}, a(2-a-b)I)$. The acceptance probability (21) can be evaluated as (28) below, while (22) is not well defined.

From the preceding discussion, the HAMS algorithm can be simplified as follows, given current variables (x_0, u_0) :

$$\begin{aligned} \tilde{Z} &= Z - a\nabla U(x_0) + \sqrt{ab}u_0, \quad Z \sim \mathcal{N}(\mathbf{0}, a(2-a-b)I), \\ x^* &= x_0 + \tilde{Z}, \quad u^* = -u_0 + \sqrt{\frac{b}{a}}\tilde{Z} + \phi(\tilde{Z} + \nabla U(x_0) - \nabla U(x^*)), \\ Z^* &= \tilde{Z} - a\nabla U(x^*) - \sqrt{ab}u^*. \end{aligned}$$

The proposal (x^*, u^*) is accepted with probability

$$\min \left(1, \exp \left\{ H(x_0, u_0) - H(x^*, u^*) + \frac{Z^T Z - (Z^*)^T Z^*}{2a(2-a-b)} \right\} \right). \quad (28)$$

Except for the choice of ϕ derived below, this algorithm is shown as HAMS-A in Algorithm 2, after a transformation $Z = \sqrt{a(2-a-b)}\zeta$ with $\zeta \sim \mathcal{N}(\mathbf{0}, I)$.

To derive a specific choice for ϕ , we examine the situation where the target density $\pi(x)$ deviates from standard normal. As discussed in Section 3.2, the HAMS algorithm is rejection-free, that is, the acceptance probability (28) is always 1, when the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$. We seek a choice of ϕ such that the acceptance probability can be minimally affected by the deviation of γ from 1, when $\pi(x)$ is $\mathcal{N}(\mathbf{0}, \gamma^{-1}I)$. For simplicity, we study the behavior of the quantity inside $\exp()$ in (28) as γ varies.

Lemma 1 Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, \gamma^{-1}I)$. Then the quantity inside $\exp()$ in (28) can be expressed as a quadratic form,

$$H(x_0, u_0) - H(x^*, u^*) + \frac{Z^T Z - (Z^*)^T Z^*}{2a(2-a-b)} = (x_0^T, u_0^T, Z^T) G(\gamma) (x_0^T, u_0^T, Z^T)^T,$$

where $G(\gamma)$ is a 3×3 block matrix. For $i, j = 1, 2, 3$, the (i, j) th block of $G(\gamma)$ is of the form $g_{ij}(\gamma)I$, where $g_{ij}(\gamma)$ is a scalar, polynomial of γ , with coefficients depending on (a, b, ϕ) . For any $a > 0, b \geq 0$ and $a + b \leq 2$, the coefficients of the leading terms of $g_{11}(\gamma), g_{22}(\gamma), g_{33}(\gamma)$ are simultaneously minimized in absolute values by the choice $\phi = \sqrt{ab}/(2-a)$.

It seems remarkable that a single choice of ϕ leads to simultaneous minimization of the absolute coefficients of the leading terms of $g_{11}(\gamma), g_{22}(\gamma), g_{33}(\gamma)$. Moreover, the particular choice $\phi = \sqrt{ab}/(2-a)$ also ensures that HAMS-A reduces to leapfrog or modified pMALA in the special cases where $a + b = 2$ or $b = 0$, as discussed in Section 3.5.

HAMS-B. For a singular $2A - A^2$, another possibility is to set $2I - A$ singular. We take $a_1 = 2 - a, a_3 = 2 - b$ and $a_2 = \sqrt{ab}$ in (10), with the constraints that $a > 0, b \geq 0$ and $a + b \leq 2$. The noise variance is then

$$\text{Var} \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = 2A - A^2 = \begin{pmatrix} a(2-a-b)I & \sqrt{ab}(a+b-2)I \\ \sqrt{ab}(a+b-2)I & b(2-a-b)I \end{pmatrix}, \quad (29)$$

which implies that Z_1 and Z_2 are proportional: $Z_2 = -\sqrt{b/a}Z_1$. However, it does not in general hold that $Z_2^* = -\sqrt{b/a}Z_1^*$, except for the choice $\phi = \sqrt{b/a}$. Moreover, this choice of ϕ is the only one such that any proportionality between (Z_1^*, Z_2^*) holds. This situation is in contrast with HAMS-A, where $Z_2^* = \sqrt{b/a}Z_1^*$ automatically holds for any choice of ϕ and additional consideration is needed to derive a specific choice of ϕ .

Lemma 2 For the preceding choice of A in (10), it holds that $Z_2^* = rZ_1^*$ for a constant coefficient $r \in \mathbb{R}$ and arbitrary values (x_0, u_0, Z_1) by definitions (15), (24), and (17) if and only if $r = -\sqrt{b/a}$ and $\phi = \sqrt{b/a}$.

To maintain the forward and backward transitions, illustrated in (20), using a single noise vector, we take the only feasible choice $\phi = \sqrt{b/a}$. Then the HAMS algorithm can be simplified as follows, given current variables (x_0, u_0) :

Algorithm 2: HAMS-A/HAMS-B

 Initialize x_0, u_0
for $t = 0, 1, 2, \dots, N_{iter}$ **do**

 Sample $w \sim \text{Uniform}[0, 1]$ and $\zeta \sim \mathcal{N}(\mathbf{0}, I)$

$$\text{Propose } x^* = x_t - a\nabla U(x_t) + \sqrt{ab}u_t + \sqrt{a(2-a-b)}\zeta$$
if HAMS-A **then**

$$\begin{aligned} \text{Propose } u^* &= \left(\frac{2b}{2-a} - 1\right)u_t - \frac{\sqrt{ab}}{2-a}(\nabla U(x_t) + \nabla U(x^*)) + \frac{2\sqrt{b(2-a-b)}}{2-a}\zeta \\ \zeta^* &= \left(1 - \frac{2b}{2-a}\right)\zeta - \frac{\sqrt{a(2-a-b)}}{2-a}(\nabla U(x_t) + \nabla U(x^*)) + \frac{2\sqrt{b(2-a-b)}}{2-a}u_t \end{aligned}$$

end
if HAMS-B **then**

$$\begin{aligned} \text{Propose } u^* &= u_t - \frac{\sqrt{ab}}{2-a}(\nabla U(x_t) + \nabla U(x^*)) \\ \zeta^* &= \zeta - \frac{\sqrt{a(2-a-b)}}{2-a}(\nabla U(x_t) + \nabla U(x^*)) \end{aligned}$$

end

$$\rho = \exp \left\{ H(x_t, u_t) - H(x^*, u^*) + \frac{1}{2}\zeta^T\zeta - \frac{1}{2}(\zeta^*)^T\zeta^* \right\}$$

if $w < \min(1, \rho)$ **then**

$$(x_{t+1}, u_{t+1}) = (x^*, u^*) \quad \# \text{ Accept}$$

else

$$(x_{t+1}, u_{t+1}) = (x_t, -u_t) \quad \# \text{ Reject}$$

end
end

$$\tilde{Z} = Z - (2-a)\nabla U(x_0) + \sqrt{ab}u_0, \quad Z \sim \mathcal{N}(0, a(2-a-b)I),$$

$$x^* = x_0 + \tilde{Z}, \quad u^* = u_0 + \sqrt{\frac{b}{a}}(\nabla U(x_0) + \nabla U(x^*)),$$

$$Z^* = \tilde{Z} - (2-a)\nabla U(x^*) - \sqrt{ab}u^*.$$

Similarly as discussed for HAMS-A, the acceptance probability (21) can be evaluated as (28). To facilitate comparison with HAMS-A, we use a reparametrization, $\tilde{a} = 2 - a$ and $\tilde{b} = ab/(2 - a)$, such that $ab = \tilde{a}\tilde{b}$ and $a(2 - a - b) = \tilde{a}(2 - \tilde{a} - \tilde{b})$. The transformation is one-to-one between $\{(a, b) : a > 0, b > 0, a + b \leq 2\}$ and $\{(\tilde{a}, \tilde{b}) : \tilde{a} > 0, \tilde{b} > 0, \tilde{a} + \tilde{b} \leq 2\}$. The resulting algorithm, with (\tilde{a}, \tilde{b}) relabeled as (a, b) , is shown as HAMS-B in Algorithm 2. Then the two algorithms, HAMS-A and HAMS-B, agree in the expressions for x^* .

3.4 Default choices of carryover

While the (a, b) parameterization arises naturally in our development above, the (ϵ, c) parameterization used in existing algorithms (see Section 2) has a desirable interpretation, with ϵ corresponding to a step size and c the amount of carryover momentum. By matching leapfrog and modified pMALA in special cases (see Section 3.5), our HAMS algorithms can be translated into an (ϵ, c) parameterization with the following formulae:

$$a = \frac{\epsilon^2}{1 + \sqrt{1 - \epsilon^2}} = 1 - \sqrt{1 - \epsilon^2}, \quad b = c(2 - a), \quad 0 \leq \epsilon, c \leq 1. \quad (30)$$

Because a is expressed as a function of ϵ only, and b given a is a function of c only, we also refer to a as a step size and b as a carryover.

So far, the number of tuning parameters is reduced from four in general HAMS (Algorithm 1) to two in HAMS-A/B (Algorithm 2). To facilitate applications, we seek to further reduce tuning by studying the lag-1 auto-covariance matrix for a HAMS chain in stationary when the target density $\pi(x)$ is standard normal.

Lemma 3 *Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, and $(x_0, u_0) \sim \mathcal{N}(\mathbf{0}, I)$. Given step size a , the maximum modulus of the eigenvalues of the lag-1 auto-covariance matrix $\text{Cov}((x_0, u_0), (x_1, u_1))$ is minimized by the following choice of b :*

$$\text{HAMS-A: } b = (\sqrt{2} - \sqrt{a})^2, \quad \text{HAMS-B: } b = \frac{a(2 - a)}{(\sqrt{2} + \sqrt{2 - a})^2}. \quad (31)$$

For convenience, the formulae (31) can be used as the default choices of carryover b , given step size a . On the other hand, such choices are derived under an idealized setting, where the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$. For the default tuning to be effective, we often need to first apply transformations to bring $\pi(x)$ closer to $\mathcal{N}(\mathbf{0}, I)$, which will be discussed in Section 3.6. If such a transformation is not available for various reasons, then it is preferable to tune both a and b instead of using the default values in (31).

3.5 Special Cases of HAMS-A/B

Recall that the constraints on the step size and carryover are $a \geq 0, b \geq 0, a + b \leq 2$. In the following, we examine three boundary cases.

The first case is when $a + b = 2$ (or equivalently $c = 1$). For both HAMS-A and HAMS-B, the updates become deterministic from (x_0, u_0) to (x^*, u^*) . To help understanding, we introduce an intermediate variable \tilde{u} . Then the updates can be written as

$$\begin{aligned}\zeta &\sim \mathcal{N}(\mathbf{0}, I), \quad \zeta^* = -\zeta \text{ (HAMS-A)}, \quad \zeta^* = \zeta \text{ (HAMS-B)}, \\ \tilde{u} &= u_0 - \sqrt{\frac{a}{2-a}} \nabla U(x_0) = u_0 - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}} \nabla U(x_0), \\ x^* &= x_0 + \sqrt{a(2-a)} \tilde{u}_0 = x_0 + \epsilon \tilde{u}, \\ u^* &= \tilde{u} - \sqrt{\frac{a}{2-a}} \nabla U(x^*) = \tilde{u} - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}} U(x^*),\end{aligned}$$

where the Metropolis ratio is $\rho = \exp(H(x_0, u_0) - H(x^*, u^*))$. The above is similar to the leapfrog discretization of the Hamiltonian dynamics but with step size $\epsilon/(1+\sqrt{1-\epsilon^2})$ instead of $\epsilon/2$ for momentum updates. The proposal (x^*, u^*) can be accepted or rejected (with u_0 flipped) based on the change in the Hamiltonian from the update.

The second case is when $b = 0$ (or equivalently $c = 0$). We introduce another intermediate variable $\tilde{\zeta}$ to the updates. Then HAMS-A and HAMS-B reduce to

$$\begin{aligned}\zeta &\sim \mathcal{N}(\mathbf{0}, I), \quad u^* = -u_0 \text{ (HAMS-A)}, \quad u^* = u_0 \text{ (HAMS-B)}, \\ \tilde{\zeta} &= \zeta - \sqrt{\frac{a}{2-a}} \nabla U(x_0) = \zeta - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}} \nabla U(x_0), \\ x^* &= x_0 + \sqrt{a(2-a)} \tilde{\zeta} = x_0 + \epsilon \tilde{\zeta}, \\ \zeta^* &= \tilde{\zeta} - \sqrt{\frac{a}{2-a}} \nabla U(x^*) = \tilde{\zeta} - \frac{\epsilon}{1+\sqrt{1-\epsilon^2}} \nabla U(x^*),\end{aligned}$$

where the Metropolis ratio is $\rho = \exp(U(x_0) - U(x^*) + \frac{1}{2}\zeta^T \zeta - \frac{1}{2}(\zeta^*)^T \zeta^*)$. Hence u_0 remains unchanged in HAMS-A, and is negated in HAMS-B, although the update of u_0 is irrelevant in this case. The update of x_0 to x^* and acceptance-rejection coincide with modified pMALA in Section 2, which differs from ordinary pMALA because the step size $\epsilon^2/(1+\sqrt{1-\epsilon^2})$ is associated with $\nabla U(x_0)$ for updating x_0 , instead of $\epsilon^2/2$.

The third case is when $a = 0$ (or equivalently $\epsilon = 0$). This case is not interesting because x remains constant. Our discussion is for completeness. When $a = 0$, HAMS-B sets all variables constant: $x^* = x_0$, $u^* = u_0$, and $\zeta^* = \zeta$. HAMS-A gives the updates

$$\begin{aligned}\zeta &\sim \mathcal{N}(\mathbf{0}, I), & x^* &= x_0, \\ u^* &= (b-1)u_0 + \sqrt{b(2-b)}\zeta, & \zeta^* &= (1-b)\zeta + \sqrt{b(2-b)}u_0.\end{aligned}$$

In this case, the Metropolis ratio is always 1. Hence HAMS-A can be viewed as an autoregressive process on u while x remains constant.

Finally, we note that our HAMS-A/B algorithms differ from UDL (Bussi and Parrinello, 2007), which uses two noise vectors per iteration, although UDL also recovers leapfrog and pMALA in the extreme cases of $c = 1$ and $c = 0$ respectively.

3.6 Preconditioning

As commonly recognized in MCMC literatures, if there is information about the variance structure of the target density, then the performance of MCMC samplers can be improved by applying a linear transformation, i.e., preconditioning. Suppose that Σ is an approximation to $\text{Var}(x)$, or M is an approximation to $(\text{Var}(x))^{-1}$. Then RWM and pMALA involve preconditioning using the approximate variance Σ on x , whereas HMC and UDL involve preconditioning using M as the momentum variance. These two approaches are conceptually equivalent, as discussed in the context of HMC by Neal (2011), although one can be more preferable than the other in computational implementations.

We use the first approach of preconditioning: applying a linear transformation to the original variable x while keeping the momentum $u \sim \mathcal{N}(\mathbf{0}, I)$. Let L be the lower triangular matrix obtained from the Cholesky decomposition $M = LL^\top$. The transformed variable is $\tilde{x} = L^\top x$. If x is approximately $\mathcal{N}(0, M^{-1})$, then \tilde{x} is approximately $\mathcal{N}(\mathbf{0}, I)$. Application of HAMS-A/B in Algorithm 2 to the transformed variable \tilde{x} leads to HAMS-A/B algorithms with preconditioning, which are shown in Algorithm 3. The gradient of the potential after the transformation, denoted as $\nabla U(\tilde{x})$, is $L^{-1}\nabla U(x)$.

Our Algorithm 3 is carefully formulated, such that transforming x and keeping $u \sim \mathcal{N}(\mathbf{0}, I)$ improves computational efficiency, compared with using the original variable x and $u \sim \mathcal{N}(\mathbf{0}, M)$. See the Appendix Section IV.8 for details of simplification. Excluding the evaluation of $U(x)$ and $\nabla U(x)$, Algorithm 3 involves 2 matrix-by-vector multiplications per iteration, $(L^\top)^{-1}\tilde{x}^*$ and $L^{-1}\nabla U(x^*)$. Moreover, computation of the Metropolis ratio ρ

Algorithm 3: HAMS-A/HAMS-B (with preconditioning)

Initialize $x_0, u_0, \tilde{x}_0 = L^T x_0$ and $\nabla U(\tilde{x}_0) = L^{-1} \nabla U(x_0)$.

```

for  $t = 0, 1, 2, \dots, N_{iter}$  do
    Sample  $w \sim \text{Uniform}[0, 1]$  and  $\zeta \sim \mathcal{N}(\mathbf{0}, I)$ 
     $\xi = \sqrt{ab}u_t + \sqrt{a(2-a-b)}\zeta, \quad \tilde{x}^* = \tilde{x}_t - a\nabla U(\tilde{x}_t) + \xi$ 
    Propose  $x^* = (L^T)^{-1}\tilde{x}^*$ 
     $\nabla U(\tilde{x}^*) = L^{-1}\nabla U(x^*), \quad \tilde{\xi} = \nabla U(\tilde{x}^*) + \nabla U(\tilde{x}_t)$ 
     $\rho = \exp \left\{ U(x_t) - U(x^*) + \frac{1}{2-a}(\tilde{\xi})^T(\xi - \frac{a}{2}\tilde{\xi}) \right\}$ 
    if  $w < \min(1, \rho)$  then
         $x_{t+1} = x^*, \quad \tilde{x}_{t+1} = \tilde{x}^*, \quad \nabla U(\tilde{x}_{t+1}) = \nabla U(\tilde{x}^*) \quad \# \text{Accept}$ 
        if HAMS-A then
             $u_{t+1} = \left( \frac{2b}{2-a} - 1 \right) u_t + \frac{2\sqrt{b(2-a-b)}}{2-a} \zeta - \frac{\sqrt{ab}}{2-a} \tilde{\xi}$ 
        end
        if HAMS-B then
             $u_{t+1} = u_t - \frac{\sqrt{ab}}{2-a} \tilde{\xi}$ 
        end
    else
         $x_{t+1} = x_t, u_{t+1} = -u_t, \tilde{x}_{t+1} = \tilde{x}_t, \nabla U(\tilde{x}_{t+1}) = \nabla U(\tilde{x}_t) \quad \# \text{Reject}$ 
    end
end

```

is also optimized, requiring only 1 inner product instead of 4 as in Algorithm 2. In contrast, UDL as described in Section 2 needs 5 matrix-by-vector multiplications per iteration: 2 for sampling from $\mathcal{N}(\mathbf{0}, M)$, 1 for computing x^* , and 2 in the Metropolis ratio. In the simulation studies, we implement UDL with reduced runtime in a similar way as Algorithm 3, in order to make fair comparisons with HAMS-A/B.

4 Generalized Metropolis–Hastings sampling

Our development in Section 3 presents a concrete class of generalized reversible algorithms, HAMS, using an augmented target density originated from a Hamiltonian in physics. In

this section, we discuss a flexible framework of generalized Metropolis–Hastings sampling for a target distribution satisfying an invariance property. This framework not only accommodates and sheds light on our construction of HAMS at a more abstract level, but also facilitates possible further development of irreversible MCMC algorithms.

Importance of rejection. Before describing our generalization, it is instructive to discuss a fictitious generalization of Metropolis–Hastings sampling, which satisfies a reversibility-like condition upon acceptance of a proposal, but in general fails to leave a target density invariant due to impropriety incurred when a proposal is rejected.

Let $\pi(y)$ be a pre-specified probability density function on a space \mathcal{Y} . By abuse of notation, we allow that $\pi(y)$ be directly a target density $\pi(x)$ in the context of Section 1 or an augmented target density $\pi(x, u)$ with auxiliary variables u . Consider an MCMC algorithm with the following transition kernel given a current value y_0 .

A fictitious generalization of Metropolis–Hastings sampling.

- Sample y^* from a (forward) proposal density $Q(\cdot|y_0)$;
- Set $y_1 = y^*$ with the acceptance probability

$$\tilde{\rho}(y^*|y_0) = \min\left(1, \frac{\pi(y^*)Q_b(y_0|y^*)}{\pi(y_0)Q(y^*|y_0)}\right),$$

or set $y_1 = y_0$ with the remaining probability, where $Q_b(\cdot|y^*)$ is a backward proposal density.

Let $\tilde{K}(y_1|y_0)$ be the (forward) transition kernel from y_0 to y_1 for the sampling scheme above. Then for any $y_1 \neq y_0$ (i.e., a proposal is accepted, $y_1 = y^*$), it can be easily shown that $\tilde{K}(y_1|y_0) = Q(y_1|y_0)\tilde{\rho}(y_1|y_0)$ and, by a symmetry argument,

$$\pi(y_0)\tilde{K}(y_1|y_0) = \pi(y_1)\tilde{K}_b(y_0|y_1), \quad (32)$$

where $\tilde{K}_b(y_0|y_1) = Q_b(y_0|y_1)\tilde{\rho}(y_0|y_1)$. If (32) were satisfied for $y_1 = y_0$ as well (i.e., a proposal is rejected), then integrating (32) over y_0 would indicate $\int \pi(y_0)\tilde{K}(y_1|y_0) dy_0 = \pi(y_1)$, that is, the transition kernel \tilde{K} leaves $\pi(\cdot)$ invariant. Standard Metropolis–Hastings sampling corresponds to choosing $Q_b = Q$, in which case (32) holds trivially for $y_1 = y_0$ as well as for $y_1 \neq y_0$. Such a condition (32) with $\tilde{K}_b = \tilde{K}$ is known as detailed balance or

reversibility. For $Q_b \neq Q$, however, (32) may not hold for $y_1 = y_0$, in spite of the fact that (32) is satisfied for $y_1 \neq y_0$. Therefore, the preceding sampling scheme in general fails to leave $\pi(\cdot)$ invariant, for the complication caused by rejection of a proposal.

Our discussion above uses an heuristic interpretation of the transition kernel \tilde{K} in the case of rejection of a proposal. The issue is also reflected in the difficulty to obtain a more rigorous justification similar as in Tierney (1994). See Ma et al. (2018), Section 3.3, for a related discussion on a naive approach for constructing irreversible samplers.

Generalized Metropolis–Hastings sampling. As motivated by our construction of HAMS algorithms, we propose generalized Metropolis–Hastings sampling provided that a target density $\pi(y)$ is invariant under an orthogonal transformation. Let J be an orthogonal matrix J such that $\pi(J^{-1}y) = \pi(y)$ for $y \in \mathcal{Y}$. By the change of variables with $|\det(J)| = 1$, this is equivalent to requiring that for any set $C \subset \mathcal{Y}$,

$$\int_{J(C)} \pi(y) dy = \int_C \pi(y) dy. \quad (33)$$

where $J(C) = \{Jy : y \in C\} \subset \mathcal{Y}$. Consider a sampling algorithm defined by the following transition kernel given a current value y_0 .

Generalized Metropolis–Hastings sampling (GMH).

- Sample y^* from a (forward) proposal density $Q(\cdot|y_0)$.
- Set $y_1 = y^*$ with the acceptance probability

$$\rho(y^*|y_0) = \min \left(1, \frac{\pi(J^{-1}y^*)Q(Jy_0|J^{-1}y^*)}{\pi(y_0)Q(y^*|y_0)} \right), \quad (34)$$

or set $y_1 = Jy_0$ with the remaining probability.

Condition (33) is trivially satisfied for $J = I$ (the identity matrix), in which case the preceding algorithm reduces to standard Metropolis–Hastings sampling.

There are two notable differences compared with the fictitious generalization earlier. First, the backward proposal density is explicitly defined as $Q(Jy_0|J^{-1}y^*)$. It is helpful to think of the proposal density $Q(y^*|y_0)$ as being induced by a stochastic mapping, $y^* = \mathcal{M}(y_0; Z)$ for a noise Z . Then $Q(Jy_0|J^{-1}y^*)$ corresponds to the density of Jy_0 given $J^{-1}y^*$ under the same mapping, $Jy_0 = \mathcal{M}(J^{-1}y^*; Z^*)$, but with a new noise Z^* considered to

be identically distributed as Z . See for example (36)–(37) below. Hence the forward and backward transitions of the proposals can be illustrated, similarly to (20), as

$$y_0 \xrightarrow{Z} y^*, \quad J^{-1}y^* \xrightarrow{Z^*} Jy_0,$$

where the two arrows denote the same mapping, depending on Z or Z^* . Second, the next variable y_1 is defined as Jy_0 instead of y_0 , in the case of rejection. The generalization can be shown to be valid in leaving the target distribution $\pi(y)$ invariant.

Proposition 3 *Suppose that invariance (33) is satisfied. Let $K(y_1|y_0)$ be the (forward) transition kernel from y_0 to y_1 for generalized Metropolis–Hastings sampling. Then generalized detailed balance holds for any $y_1 \neq Jy_0$:*

$$\pi(y_0)K(y_1|y_0) = \pi(J^{-1}y_1)K(Jy_0|J^{-1}y_1), \quad (35)$$

Moreover, the target density $\pi(y)$ is a stationary density of the Markov chain defined by the transition kernel $K(y_1|y_0)$.

To connect with HAMS, generalized Metropolis–Hastings sampling is discussed above in terms of continuous variables. However, our framework can be broadened to accommodate both continuous and discrete variables, by allowing Jy to be an orthogonal-like mapping, for example, flipping a binary variable from one value to the other. In the Supplement, we show that the irreversible jump sampler (I-Jump) in Ma et al. (2018) can be obtained as a special case of generalized Metropolis–Hastings sampling with a symmetric, binary auxiliary variable. Hence our HAMS algorithm differs from I-Jump in using momentum as an auxiliary variable, and exploiting symmetry of mean-zero normal distributions.

Generalized gradient-guided Metropolis sampling. The framework of generalized Metropolis–Hastings sampling allows a flexible specification of the proposal density Q . Our HAMS algorithms use a proposal scheme which takes a gradient step and then adds Gaussian noises. Using a similar update scheme, (36) below, in generalized Metropolis–Hastings sampling leads to a class of gradient-guided sampling algorithms. Similarly as in Section 3.1, let $\mathbf{0} \leq A \leq 2I$ be a symmetric matrix in the order on positive semi-definite matrices. For a target $\pi(y)$, a potential function $U(y)$ is defined such that $\pi(y) \propto \exp\{-U(y)\}$. This potential $U(y)$ can be the augmented potential $U(x) + u^T u / 2$ in Section 3.

Generalized gradient-guided Metropolis sampling (G2MS).

- Generate y^* as

$$y^* = y_0 - B\nabla U(y_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, 2A - A^2), \quad (36)$$

where $B = I - (I - A)J$ and $2A - A^2 = B + B^T - BB^T$. Compute Z^* by

$$Jy_0 = J^{-1}y^* - B\nabla U(J^{-1}y^*) + Z^*, \quad (37)$$

obtained by replacing (y_0, y^*) with $(J^{-1}y^*, Jy_0)$ and Z with Z^* in (36).

- Set $y_1 = y^*$ with the acceptance probability (34), simplified as

$$\rho(y^*|y_0) = \min\left(1, \frac{\pi(y^*)\mathcal{N}(Z^*|\mathbf{0}, 2A - A^2)}{\pi(y_0)\mathcal{N}(Z|\mathbf{0}, 2A - A^2)}\right), \quad (38)$$

or set $y_1 = Jy_0$ with the remaining probability.

Corollary 2 Suppose that invariance (33) is satisfied. The conclusions of Proposition 3 hold with transition kernel K defined by generalized gradient-guided Metropolis sampling.

In addition to exploiting gradient information, the G2MS algorithm is carefully designed to achieve the rejection-free property when the target density $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$, which satisfies invariance (33) for any orthogonal matrix J . In this case, $U(y) = y^T y / 2$ with gradient $\nabla U(y) = y$, and hence the proposal scheme (36) becomes

$$y^* = (I - A)Jy_0 + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, 2A - A^2). \quad (39)$$

The update from y_0 to y^* defines a VAR(1) process, which admits $\mathcal{N}(\mathbf{0}, I)$ as a stationary distribution, that is, if $y_0 \sim \mathcal{N}(\mathbf{0}, I)$ then $y^* \sim \mathcal{N}(\mathbf{0}, I)$, by similar calculation as in (9). However, stationarity of (39) with respect to $\mathcal{N}(\mathbf{0}, I)$ does not automatically imply rejection-free. In fact, because $(I - A)J$ may be asymmetric, the VAR(1) process in (39) is in general irreversible. Standard Metropolis–Hastings sampling using the proposal scheme (39) is not rejection-free when $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$. Otherwise, the resulting Markov chain is irreversible, which contradicts reversibility of standard Metropolis–Hastings sampling. Nevertheless, the G2MS algorithm achieves rejection-free when $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$, due to the combination of the proposal scheme (39) with the generalized acceptance probability (38). In other words, the backward proposal density induced from (37) agrees with the conditional density of y_0 given y^* if $y_0 \sim \mathcal{N}(\mathbf{0}, I)$ and y^* is generated by (39). See the proof for details.

Corollary 3 Suppose that the target density $\pi(y)$ is $\mathcal{N}(\mathbf{0}, I)$. Then the generalized acceptance probability (38) reduces to 1, and hence y^* from the proposal scheme (36) is always accepted under the G2MS algorithm.

From the preceding discussion, the G2MS algorithm can be seen as being extended from a VAR(1) process in the form (39). For completeness, we remark that the form of (39) depending on A and J is universal. In fact, consider a general VAR(1) process

$$y^* = (I - \tilde{B})y_0 + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, \tilde{B} + \tilde{B}^T - \tilde{B}\tilde{B}^T), \quad (40)$$

where \tilde{B} is a possibly asymmetric matrix such that $\tilde{B} + \tilde{B}^T - \tilde{B}\tilde{B}^T$ is (symmetric and) positive semi-definite. Let $I - \tilde{B} = O_1 \Lambda O_2$ be a singular value decomposition, where O_1 and O_2 are orthogonal matrices, Λ is a diagonal matrix containing the singular values of $I - \tilde{B}$. Then $I - \tilde{B}$ can be written as

$$I - \tilde{B} = (O_1 \Lambda O_1^T)(O_1 O_2) = (I - \tilde{A})\tilde{J},$$

where $\tilde{A} = I - O_1 \Lambda O_1^T$ is symmetric and $\tilde{J} = O_1 O_2$ is orthogonal. Moreover, the noise variance becomes $\tilde{B} + \tilde{B}^T - \tilde{B}\tilde{B}^T = I - (I - \tilde{B})(I - \tilde{B})^T = I - (I - \tilde{A})^2 = 2\tilde{A} - \tilde{A}^2$. Therefore, the VAR(1) process (40) can be put in the form (39).

Back to HAMS. The invariance (33) can be satisfied by an augmented target density defined with auxiliary variables. In fact, our HAMS algorithms can be recovered as special cases of generalized Metropolis–Hastings sampling, with $\pi(y) = \pi(x, u)$ in (6) and J a block-diagonal matrix with $(I, -I)$ on the diagonal. The invariance (33) is satisfied due to evenness of mean-zero normal distributions. The HAMS algorithm studied in Proposition 1 is a special case of G2MS with the A matrix in (10). The HAMS algorithm in Proposition 2 is not contained in G2MS due to a modification with $\phi \neq 0$, but can still be treated in the framework of generalized Metropolis–Hastings sampling, with the forward and backward proposal schemes discussed in Section 3.2. The general discussion here broadens our understanding of HAMS algorithms and opens doors for further development.

5 Simulation Studies

We report simulation studies comparing HAMS-A/B with RWM, pMALA, pMALA*, HMC, UDL, and GMC (see Section 2). We include RWM as a performance baseline. The simulations include a multivariate normal distribution, a stochastic volatility model and a log-Gaussian Cox model. For space limitation, the normal experiment and results from pMALA* and GMC in the other two experiments are deferred to the Supplement.

For ease of comparison and tuning, we use the (ϵ, c) parameterization for HAMS-A and HAMS-B, equivalent to the (a, b) parametrization by (30). We fix the number of leapfrog steps for HMC similarly as in Girolami and Calderhead (2011): $nleap = 50$ in sampling latent variables or $nleap = 6$ in sampling parameters. When preconditioning is applied, the c values for HAMS-A/B as well as UDL and GMC are determined in terms of ϵ , by translating the default choices of b given a in (31). Without preconditioning, the c values are specified by the following consideration. Recall that the first momentum update of UDL is $u^+ = \sqrt{cu_0 + \sqrt{1-c}}Z_1$ in the form of an AR(1) process. With a standard normal noise, the lag- h auto-covariance for AR(1) is $\gamma(h) = c^{h/2}$. To match resampling of momentum in HMC, we require $c = \gamma(h)^{2/h}$ with $h = nleap$ and a small value, 0.001, for $\gamma(h)$. Hence we set $c = 0.76$ or 0.1 corresponding to $nleap = 50$ or 6.

For tuning, we adjust step size ϵ during a burn-in period to achieve reasonable acceptance rates: around 30% for RWM and 70% for all other methods. See the Supplement Section V.3 for details. Samples are then collected after the burn-in.

To evaluate MCMC samples, a useful metric is the effective sample size, $ESS = n/\{1 + 2\sum_{k=1}^{\infty} \rho(k)\}$, where n is the total number of draws and $\rho(k)$ is the lag- k correlation. To deal with irreversible Markov chains obtained by HAMS-A/B as well as UDL, we use the Bartlett window estimator of ESS similarly as in Ma et al. (2018):

$$ESS = \frac{n}{1 + 2\sum_{k=1}^K \left(1 - \frac{k}{K}\right) \rho(k)}, \quad (41)$$

where the cutoff value K is a large number (taken to be 3000 in our results). Moreover, ESS can be estimated from each coordinate for a multi-dimensional distribution. As suggested in Girolami and Calderhead (2011), we report the minimum ESS over all coordinates, adjusted by runtime, as a measure of computational efficiency.

5.1 Stochastic volatility model

Consider a stochastic volatility model (Kim et al., 1998), where latent volatilities are generated as

$$x_t = \phi x_{t-1} + \eta_t, \quad \eta_t \sim \mathcal{N}(0, \sigma^2), \quad t = 2, 3, \dots, T, \quad (42)$$

with $x_1 \sim \mathcal{N}(0, \sigma^2/(1 - \phi^2))$, and the observations are generated as

$$y_t = z_t \beta \exp\{x_t/2\}, \quad z_t \sim \mathcal{N}(0, 1), \quad t = 1, \dots, T. \quad (43)$$

The parameters of interest are $\theta = (\beta, \sigma, \phi)^T$. We simulate $T = 1000$ observations from (42)–(43) using parameter values $\beta = 0.65$, $\sigma = 0.15$ and $\phi = 0.98$. Let $\mathbf{x} = (x_1, \dots, x_T)^T$ and $\mathbf{y} = (y_1, \dots, y_T)^T$. Two sets of experiments are conducted. First, we fix parameter values and sample latent variables from $p(\mathbf{x}|\mathbf{y}, \theta)$. Then we perform Bayesian analysis and sample both the parameters and latent variables from $p(\mathbf{y}|\mathbf{x}, \theta)$. See Supplement Section V.1 for expressions of gradients and preconditioning matrices used.

For the first experiment, we fix parameters at their true values and perform sampling for latent variables only. The joint distribution of (x_1, \dots, x_T) is $\mathcal{N}(\mathbf{0}, C)$, with entries of the covariance matrix given by $C[i, j] = \phi^{|i-j|}\sigma^2/(1 - \phi^2)$. Its inverse C^{-1} retains a simple tri-diagonal form. Following Girolami and Calderhead (2011), the inverse variance $[\text{Var}(\mathbf{x})]^{-1}$ can be approximated by $-\mathbb{E}[\nabla^2 \log p(\mathbf{x}|\mathbf{y}, \theta)] = C^{-1} + \frac{1}{2}I$. Hence for preconditioning, we set $M = C^{-1} + \frac{1}{2}I$ for HAMS-A/B, UDL and HMC, and $\Sigma = M^{-1}$ for pMALA and RWM. As mentioned earlier, we use $nleap = 50$ for HMC and choose c given ϵ by (30). All algorithms are run for 5000 burn-in iterations, and then samples are collected from 5000 iterations. The simulation process is repeated for 50 times.

Table 1 shows the runtime and ESS comparison. Clearly, HAMS-A has the best performance in terms of time-adjusted minimum ESS, followed by HAMS-B. An interesting phenomenon about the ESSs from HAMS-A/B as well as HMC is that an ESS value estimated by (41) can exceed the actual number of draws collected, due to negative auto-correlations. Figure 1 shows trace plots of one latent variable and corresponding autocorrelation function (ACF) plots from an individual run. The plots for each method are adjusted for runtime after burn-in: we keep the number of draws inversely proportional to the runtime, with

Method	Time (s)	ESS (min, median, max)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	98.7	(2420, 3660, 6668)	24.51
HAMS-B	99.6	(1915, 3404, 6229)	19.23
UDL	98.4	(657, 1020, 1661)	6.68
HMC	1250.1	(1125, 3698, 11240)	0.90
pMALA	120.5	(374, 610, 990)	3.11
RWM	51.7	(7, 12, 20)	0.14

Table 1: Runtime and ESS comparison for sampling latent variables in the stochastic volatility model. Results are averaged over 50 repetitions.

RWM keeping all 5000 draws as the baseline. All time-adjusted plots are produced similarly in this and next sections. From the trace plots, HAMS-A and HAMS-B appear to mix better than other methods. Moreover, the ACFs of HAMS-A and HAMS-B decay faster to 0 compared with other methods, while exhibiting negative auto-correlations.

Figure 2 shows the time-adjusted boxplots of the sample means of all latent variables for each method over 50 repeated runs. The boxplots are centered at the corresponding averages, and narrower boxplots indicate that a method is more consistent across repeated simulations. Clearly, HAMS-A and HAMS-B are the most consistent, followed by UDL and pMALA. Much more variability is associated with HMC and RWM.

The superior performances of HAMS-A/B can be attributed to the fact that larger step sizes are used by HAMS-A/B than other methods, while similar acceptance rates are obtained. See the Supplement Figure S4. A possible explanation for the step size differences is that HAMS-A/B satisfies the rejection-free property and hence is more capable of achieving reasonable acceptance rates with relatively large step sizes when the target density is not far from a normal density through preconditioning.

In the second experiment, we perform Bayesian analysis and sample both latent variables and parameters from the posterior $p(\mathbf{x}, \theta | \mathbf{y})$. The priors are, independently, $\pi(\beta) \propto \beta^{-1}$, $\sigma^2 \sim \text{Inv-}\chi^2(10, 0.05)$ and $(\phi + 1)/2 \sim \text{Beta}(20, 1.5)$. Moreover, we use the transfor-

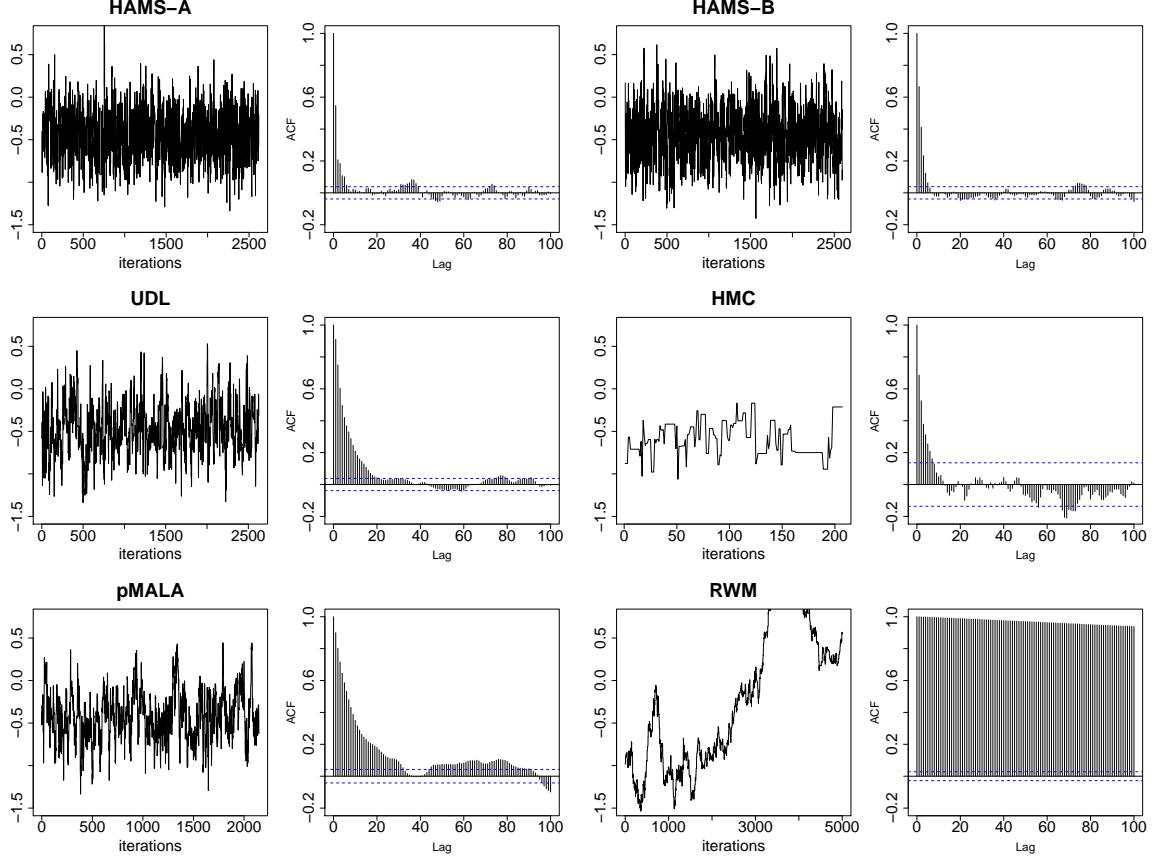


Figure 1: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the stochastic volatility model.

mations $\sigma = \exp(\gamma)$ and $\phi = \tanh(\alpha)$ to ensure that $\sigma > 0$ and $|\phi| < 1$. We employ a Gibbs-sampling scheme, alternating between $p(\mathbf{x}|\mathbf{y}, \theta)$ and $p(\theta|\mathbf{y}, \mathbf{x})$, similarly as in Girolami and Calderhead (2011). In the first experiment, the preconditioning matrix for latent variables needs to be computed only once because the parameters are fixed. In the current experiment, to avoid re-evaluating the preconditioning matrix every Gibbs iteration, we first run each algorithm without any preconditioning to obtain a crude estimate of the parameters, and then fix the preconditioning matrix evaluated at this estimate. For HMC, the numbers of leapfrog steps are 50 for latent variables and 6 for parameters. The initial values of parameters are dispersed over the following intervals $\beta \in [0.5, 2]$, $\sigma \in [0.1, 1]$, and $\phi \in [0, 0.3]$. For all methods, 10000 draws are collected after a burn-in of 10000 iterations, which include two stages without preconditioning and one stage of tuning with

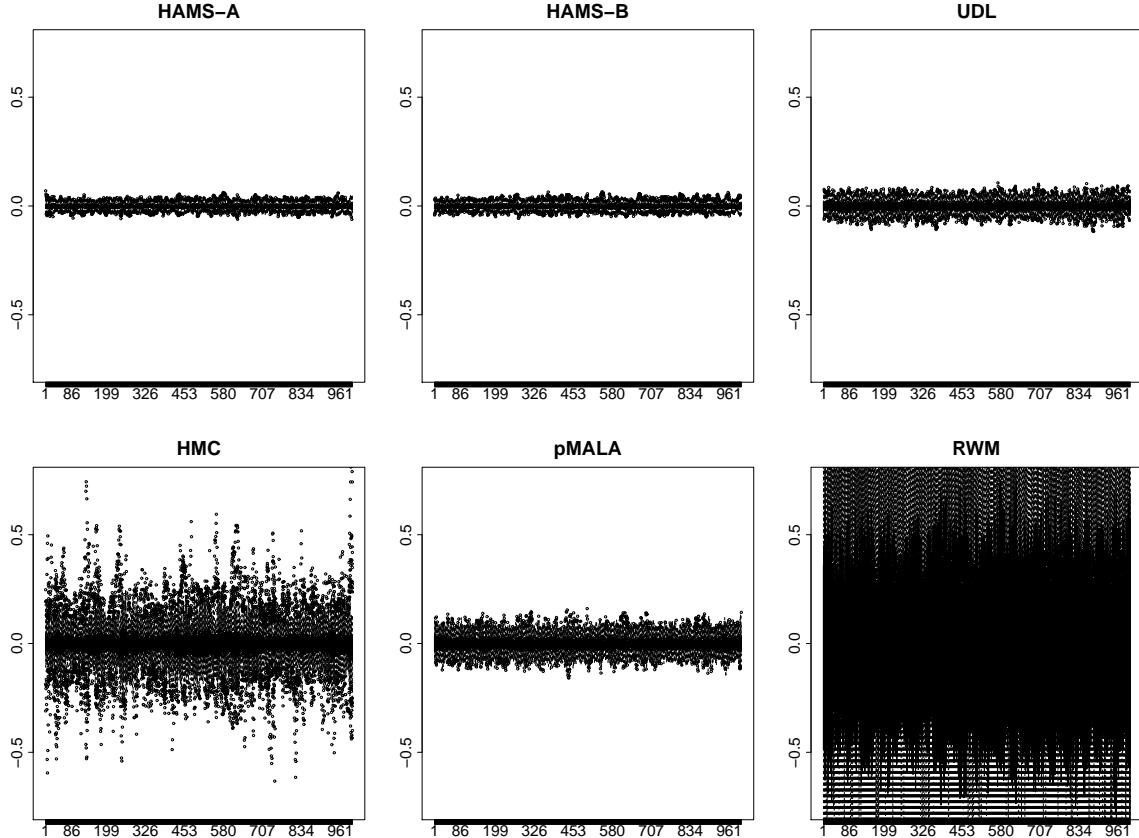


Figure 2: Time-adjusted and centered boxplots of sample means of all latent variables over 50 repetitions for sampling latent variables in the stochastic volatility model.

preconditioning. The simulation process is repeated for 20 times.

Table 2 shows the results of posterior sampling. Except for RWM, the methods yield similar averages of sample means of the parameters. However, HAMS-A and HAMS-B produce smaller standard deviations of sample means than the remaining methods, except that pMALA gives a smaller standard deviation of sample means in σ , although substantially lower ESSs in all the parameters than HAMS-A/B. In fact, HAMS-A and HAMS-B clearly outperform the other methods in terms of ESSs in all three parameters.

Figure 3 shows time-adjusted density plots for the parameters. Each plot shows densities from 20 repeated runs overlaid together. Clearly, HAMS-A yields the most consistent density curves for all three parameters, followed by HAMS-B, UDL, and pMALA which sometimes produce outlying curves, especially in β and σ .

Method	Time (s)	Sample Mean			ESS (β, σ, ϕ)	$\frac{\text{minESS}}{\text{Time}}$
		β (sd)	σ (sd)	ϕ (sd)		
HAMS-A	1951.3	0.68 (0.034)	0.19 (0.006)	0.98 (0.001)	(30, 73, 220)	0.015
HAMS-B	1942.3	0.68 (0.037)	0.19 (0.007)	0.98 (0.001)	(25, 59, 188)	0.013
UDL	1945.8	0.68 (0.039)	0.20 (0.008)	0.98 (0.002)	(29, 37, 87)	0.015
HMC	20920.2	0.69 (0.050)	0.19 (0.014)	0.98 (0.003)	(19, 12, 78)	0.001
pMALA	2013.0	0.68 (0.040)	0.20 (0.005)	0.98 (0.001)	(15, 30, 76)	0.008
RWM	1311.1	0.76 (0.050)	0.47 (0.229)	0.51 (0.149)	(89, 12, 7)	0.006

Table 2: Comparison of posterior sampling in the stochastic volatility model. Standard deviations of sample means are in parentheses. Results are averaged over 20 repetitions.

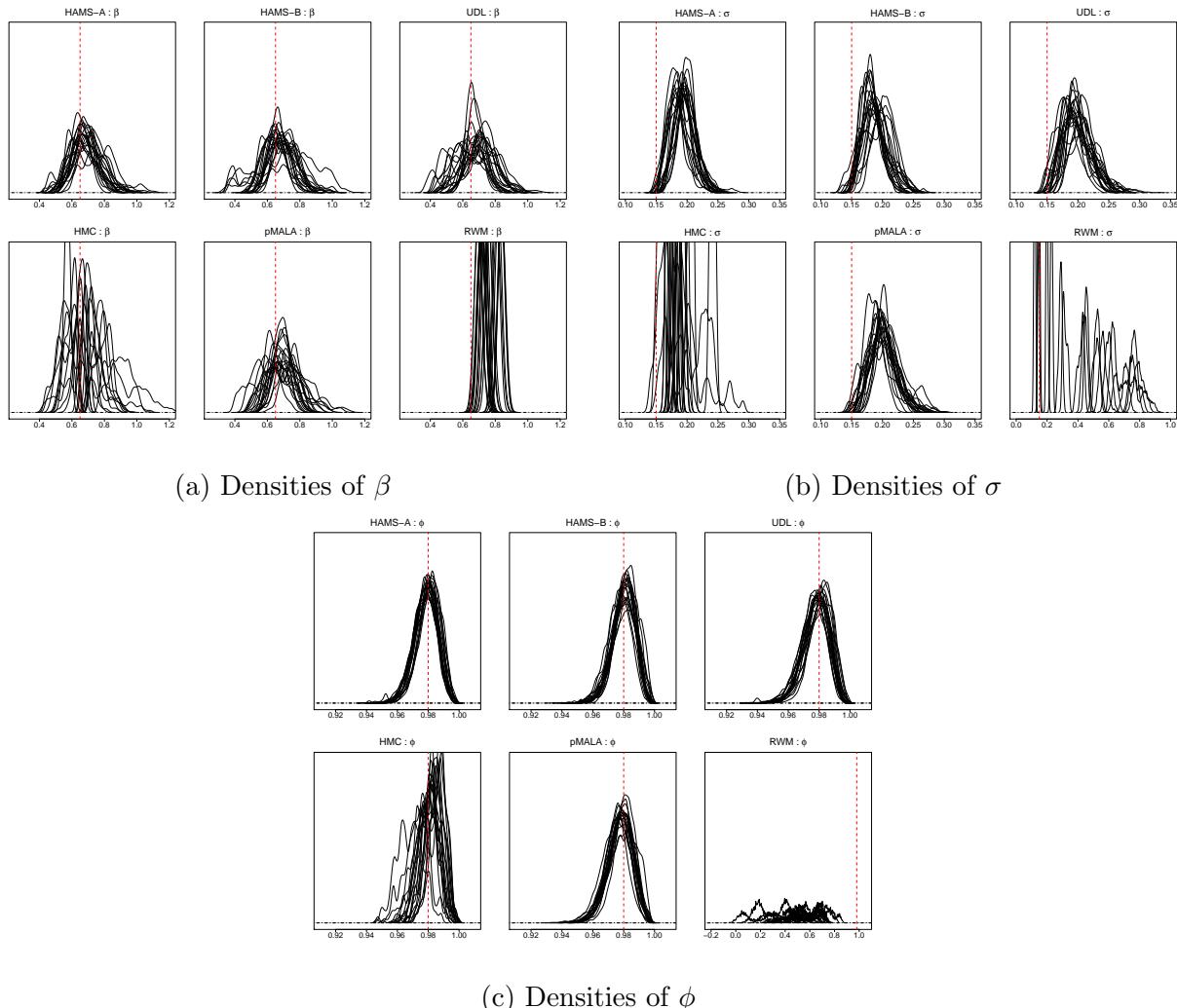


Figure 3: Time-adjusted posterior density plots of parameters (20 repetitions overlaid) in the stochastic volatility model. The true parameter values are marked by vertical lines.

Method	Time (s)	ESS (min, median, max)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	81.0	(803, 1655, 5461)	9.91
HAMS-B	78.8	(619, 1376, 4831)	7.86
UDL	78.8	(322, 622, 1761)	4.08
HMC	1285.9	(935, 1621, 4523)	0.73
pMALA	116.4	(184, 340, 1002)	1.58
RWM	51.1	(8, 13, 22)	0.16

Table 3: Runtime and ESS comparison for sampling latent variables in the log-Gaussian Cox model ($n = 1024$). Results are averaged over 50 repetitions.

5.2 Log-Gaussian Cox model

Consider a log-Gaussian Cox model, where the latent variables $\mathbf{x} = (x_{ij})_{i,j=1,\dots,m}$ are associated with an $m \times m$ grid (Christensen et al., 2005; Girolami and Calderhead, 2011). Assume that x_{ij} 's are normal with means 0 and a covariance function $C[(i, j), (i', j')] = \sigma^2 \exp(-\sqrt{(i - i')^2 + (j - j')^2}/(m\beta))$. By abuse of notation, we denote $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, C)$, of dimension $n = m^2$. The observations $(y_{ij})_{i,j=1,\dots,m}$ are independently Poisson, where the mean of y_{ij} is $\lambda_{ij} = n^{-1} \exp(x_{ij} + \mu)$, with μ treated as known. Hence the unknown parameters are $\theta = (\sigma^2, \beta)^T$. Given a prior $\pi(\theta)$, the posterior density is

$$p(\mathbf{x}, \theta | \mathbf{y}) \propto \pi(\theta) |\det(C)|^{-1/2} \exp \left\{ -\frac{1}{2} \mathbf{x}^T C^{-1} \mathbf{x} \right\} \exp \left\{ \sum_{i,j} (y_{ij}(x_{ij} + \mu) - \lambda_{ij}) \right\}. \quad (44)$$

As in Section 5.1, we conduct two sets of experiments: one is sampling latent variables with fixed parameters, and the other is sampling both parameters and latent variables.

For latent variables sampling, we take $m = 32$ and generate $n = 32^2 = 1024$ observations using the parameter values $\sigma^2 = 1.91$, $\beta = 0.3$ and $\mu = \log(126) - 0.5(1.91)$. The example in Christensen et al. (2005) and Girolami and Calderhead (2011) used $\beta = 1/33$. Here we increase β to introduce more correlations in \mathbf{x} which makes the problem more challenging and leads to clearer comparison between different methods. From (44), the gradient of the negative log-likelihood is $\nabla U(\mathbf{x}) = n^{-1} \exp(\mathbf{x} + \mu) + C^{-1}\mathbf{x} - \mathbf{y}$. The expected Hessian is

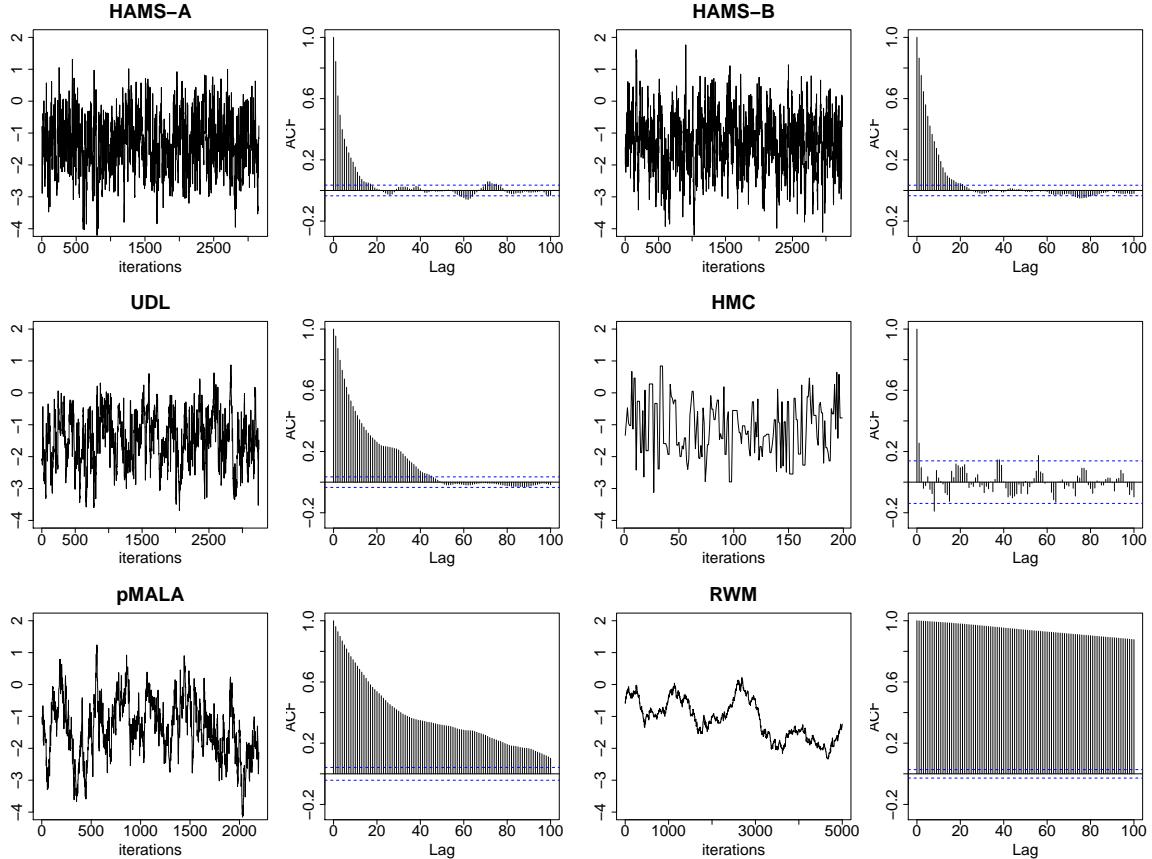


Figure 4: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

$\mathbb{E}[\nabla^2 U(\mathbf{x})] = D + C^{-1}$, taken with respect to the prior of \mathbf{x} , where D is a diagonal matrix with diagonal elements $n^{-1} \exp(\mu + \frac{1}{2}\sigma^2)$. Hence for preconditioning, we set $\Sigma^{-1} = M = D + C^{-1}$. The number of leapfrog steps is 50 for HMC. For all methods, 5000 draws are collected after a burn-in of 5000. The simulation process is repeated for 50 times.

Table 3 summarizes runtime and ESSs. Similarly as in Section 5.1, HAMS-A has the best performance in terms of time-adjusted minimum ESS, followed by HAMS-B. Notice that HMC has large raw ESSs than UDL and pMALA, but its performance is worse after adjusting for runtime. Figure 4 shows time-adjusted trace plots of one latent variable and corresponding ACF plots taken from an individual run. From both plots, HAMS-A and HAMS-B appear to mix better than the other methods. Figure 5 shows the time-adjusted and centered boxplots of sample means for each method over 50 repetitions. The spread of these boxplots corroborate the ESS results: HAMS-A and HAMS-B are less variable than

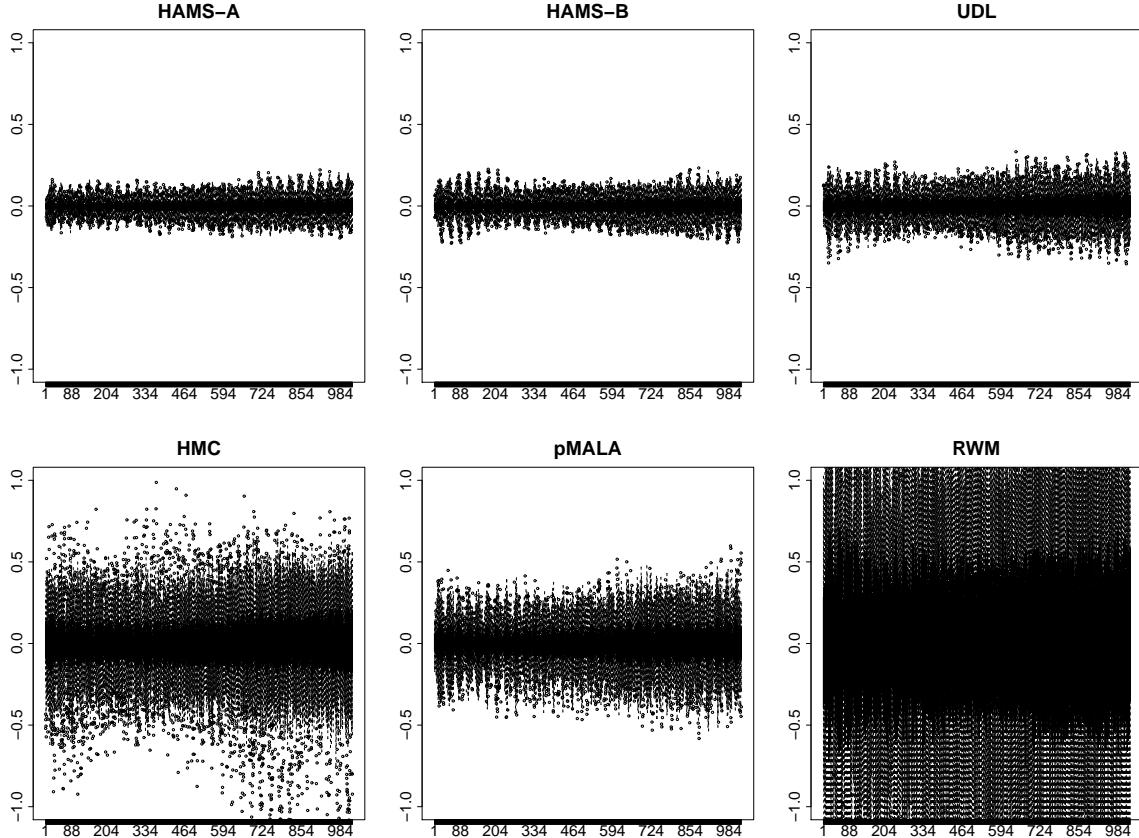


Figure 5: Time-adjusted and centered boxplots of sample means of all latent variables for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

the remaining methods over repeated simulations.

Similarly as in Section 5.1, the superior performances of HAMS-A/B are related to the rejection-free property of HAMS-A/B, which facilitates use of relatively large step sizes while reasonable acceptance rates are obtained. See Supplement Figure S13.

Our final experiment is sampling both latent variables and parameters for Bayesian analysis of the log-Gaussian Cox model. Unlike the stochastic volatility model where the inverse of covariance matrix of latent variables admits a closed-form expression, the matrix C here needs be inverted numerically whenever we evaluate the density (44) or its gradient. For large n , sampling both parameters and latent variables is computationally demanding. Hence we consider a reduced size $m = 16$ and $n = 256$. We still simulate observations \mathbf{y} using the ground truth $\sigma^2 = 1.91$, $\beta = 0.3$ and $\mu = \log(126) - 0.5(1.91)$. The priors

Method	Time (s)	Sample Mean σ^2 (sd)	β (sd)	ESS (σ^2, β)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	2766.8	3.90 (0.155)	0.68 (0.073)	(978, 207)	0.075
HAMS-B	2762.8	3.93 (0.190)	0.69 (0.106)	(838, 263)	0.095
UDL	2759.1	3.79 (0.171)	0.59 (0.105)	(755, 246)	0.089
HMC	25386.0	3.88 (0.084)	0.75 (0.113)	(2253, 139)	0.005
pMALA	2755.3	3.76 (0.189)	0.57 (0.101)	(528, 178)	0.065
RWM	1752.2	3.70 (0.662)	1.26 (1.434)	(226, 87)	0.050

Table 4: Comparison of posterior sampling in log-Gaussian Cox model ($n = 256$). Standard deviations of sample means are in parentheses. Results are averaged over 20 repetitions.

are $\sigma^2, \beta \sim \text{Gamma}(2, 0.5)$, independently. Then we perform Gibbs sampling, alternating between $p(\mathbf{x}|\mathbf{y}, \theta)$ and $p(\theta|\mathbf{y}, \mathbf{x})$, after the transformation $\sigma^2 = \exp(\varphi_1)$ and $\beta = \exp(\varphi_2)$. See Supplement Section V.2 for details of associated calculations. HMC takes 50 leapfrog steps for latent variables and 6 for parameters. For each method, 5000 draws are collected after a burn-in period of 9000, which include two stages without preconditioning and one stage of tuning with preconditioning. The simulation process is repeated for 20 times using dispersed starting values for the parameters $\sigma^2 \in [0.25, 4]$ and $\beta \in [0.05, 1]$.

Table 4 summarizes the results of posterior sampling. Figure 6 shows time-adjusted overlaid density plots for the parameters. As shown in these plots, the posterior distributions of both σ^2 and β are highly right-skewed. Accounting for this skewness, we consider the sample means roughly aligned between different methods excluding RMW. From Table 4, while HMC has the smallest standard deviation and largest ESS in σ^2 , it shows poor performance in β with the largest standard deviation and smallest ESS. Among the remaining four methods, HAMS-A has the smallest standard deviation in both σ^2 and β , the largest ESS in σ^2 while HAMS-B has the largest ESS in β .

6 Conclusion

We propose a broad class of HAMS algorithms and develop two specific algorithms, HAMS-A/B, with convenient tuning and preconditioning strategies. These algorithms achieve

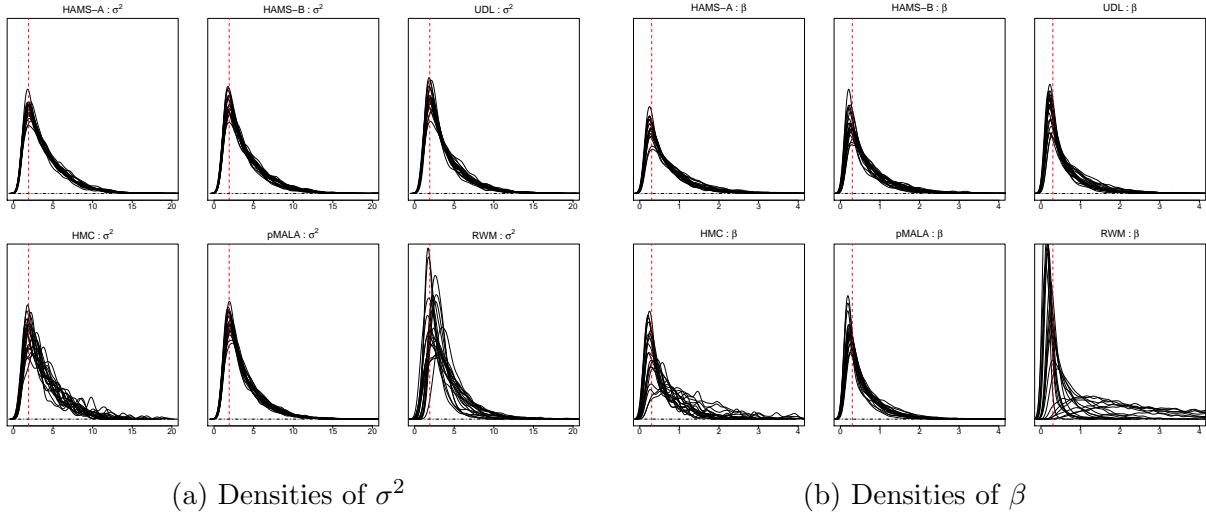


Figure 6: Time-adjusted posterior density plots (20 repetitions overlaid) in log-Gaussian Cox model ($n = 256$). The true parameter values are marked by vertical lines.

two distinctive properties: generalized reversibility and, for a normal target with a pre-specified variance, rejection-free. Our numerical experiments demonstrate advantages of the proposed algorithms compared with existing ones. Nevertheless, there are various topics of interest for further research. In addition to HAMS-A/B, alternative algorithms can be derived by choosing a nonsingular noise variance $2A - A^2$, which corresponds to two noise vectors per iteration. These algorithms can be studied, together with HAMS-A/B and other algorithms related to underdamped Langevin dynamics. In addition, it is desired to provide quantitative analysis of performances of sampling algorithms with or without the rejection-free property. Finally, our framework of generalized Metropolis–Hastings can be exploited to develop other possible irreversible sampling algorithms.

References

- Adler, S. L. (1981). Over-relaxation method for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D*, 23:2901–2904.
- Besag, J. E. (1994). Comments on “Representations of knowledge in complex systems” by U. Grenander and M.I. Miller. *Journal of the Royal Statistical Society, Ser. B*, 56:591–592.
- Bierkens, J., Fearnhead, P., and Roberts, G. (2019). The Zig-Zag process and super-efficient sampling for Bayesian analysis of big data. *Annals of Statistics.*, 47:1288–1320.

- Bouchard-Cote, A., Vollmer, S. J., and Doucet, A. (2018). The bouncy particle sampler: A nonreversible rejection-free Markov chain Monte Carlo method. *Journal of the American Statistical Association*, 113:855–867.
- Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. CRC press.
- Bussi, G. and Parrinello, M. (2007). Accurate sampling using Langevin dynamics. *Physical Review E*, 75:056707.
- Cheng, X., Chatterji, N. S., Bartlett, P. L., and Jordan, M. I. (2018). Underdamped Langevin MCMC: A non-asymptotic analysis. In *Proceedings of the 31st Conference On Learning Theory*, volume 75, pages 300–323.
- Christensen, O. F., Roberts, G. O., and Rosenthal, J. S. (2005). Scaling limits for the transient phase of local Metropolis–Hastings algorithms. *Journal of the Royal Statistical Society, Ser. B*, 67:253–268.
- Cotter, S. L., Roberts, G. O., Stuart, A. M., and White, D. (2013). MCMC methods for functions: Modifying old algorithms to make them faster. *Statistical Science*, 28:424–446.
- Dalalyan, A. S. and Riou-Durand, L. (2018). On sampling from a log-concave density using kinetic Langevin diffusions. *Bernoulli*. to appear.
- Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195:216–222.
- Gardiner, C. (1997). *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*. Springer.
- Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society, Ser. B*, 73:123–214.
- Goga, N., Rzepiela, A. J., de Vries, A. H., Marrink, S. J., and Berendsen, H. J. C. (2012). Efficient algorithms for Langevin and DPD dynamics. *Journal of Chemical Theory and Computation*, 8:3637–3649.
- Grønbech-Jensen, N. and Farago, O. (2013). A simple and effective Verlet-type algorithm for simulating Langevin dynamics. *Molecular Physics*, 111:983–991.
- Grønbech-Jensen, N. and Farago, O. (2020). Defining velocities for accurate kinetic statistics in the Grønbech-Jensen Farago thermostat. *Physical Review E*, 101:022123.
- Gustafson, P. (1998). A guided walk Metropolis algorithm. *Statistics and Computing*, 8:357–364.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109.
- Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn Sampler: Adaptively setting path

- lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623.
- Horowitz, A. M. (1991). A generalized guided Monte Carlo algorithm. *Physics Letters B*, 268:247–252.
- Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *Review of Economic Studies*, 65:361–393.
- Liu, J. (2001). *Monte Carlo Strategies in Scientific Computing*. Springer.
- Ma, Y.-A., Fox, E., Chen, T., and Wu, L. (2018). Irreversible samplers from jump and continuous Markov processes. *Statistics and Computing*, 29:177–202.
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092.
- Neal, R. M. (1998). Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In *Learning in Graphical Models*, pages 205–228. Springer.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo*, chapter 5. CRC Press.
- Osawa, H. (1988). Reversibility of first-order autoregressive processes. *Stochastic Processes and their Applications*, 28:61–69.
- Ottobre, M., Pillai, N. S., Pinski, F. J., and Stuart, A. M. (2016). A function space HMC algorithm with second order Langevin diffusion limit. *Bernoulli*, 22:60–106.
- Roberts, G. O. and Tweedie, R. L. (1996). Exponential convergence of Langevin distributions and their discrete approximations. *Bernoulli*, 2:341–363.
- Scemama, A., Lelivre, T., Stoltz, G., Cancs, E., and Caffarel, M. (2006). An efficient sampling algorithm for variational Monte Carlo. *Journal of Chemical Physics*, 125:114105.
- Syed, S., Bouchard-Cote, A., Deligiannidis, G., and Doucet, A. (2019). Non-reversible parallel tempering: A scalable highly parallel MCMC scheme. arXiv preprint:1905.02939.
- Tierney, L. (1994). Markov chains for exploring posterior distributions. *Annals of Statistics*, 22:1701–1728.
- Titsias, M. K. and Papaspiliopoulos, O. (2018). Auxiliary gradient-based sampling algorithms. *Journal of the Royal Statistical Society, Ser. B*, 80:749–767.
- van Gunsteren, W. and Berendsen, H. (1982). Algorithms for brownian dynamics. *Molecular Physics*, 45:637–647.
- Vucelja, M. (2016). Lifting — A nonreversible Markov chain Monte Carlo algorithm. *American Journal of Physics*, 84:958–968.

Supplementary Material for
“Hamiltonian Assisted Metropolis Sampling”

Zexi Song and Zhiqiang Tan

I Auxiliary variable derivation of proposal schemes

We show that the proposal scheme (7) can also be derived through an auxiliary variable argument related to Titsias and Papaspiliopoulos (2018), combined with an over-relaxation technique as in Adler (1981) and Neal (1998). Compared with Titsias and Papaspiliopoulos (2018), our derivation deals with the augmented density of (x, u) , instead of x alone. More importantly, our derivation incorporates an over-relaxation technique to accommodate all possible proposal schemes (7). Finally, our derivation invokes a different normal approximation to the target distribution and, when applied without the momentum variable, would lead to the modified pMALA algorithm as discussed in Section 2.

The starting point of our derivation is to introduce auxiliary variables (y, v) and further augment the target density as $\pi(x, u, y, v) = \pi(x, u)\pi(y, v|x, u)$. The conditional density $\pi(y, v|x, u)$ can be defined from a random walk update,

$$(y, v)|(x, u) \sim \mathcal{N}((x, u), S), \quad (\text{S1})$$

where S is a $(2k) \times (2k)$ variance matrix independent from (x, u) . Given (x_0, u_0) , consider the following steps to sample from the new target:

- sample $(y, v)|(x_0, u_0) \sim \pi(y, v|x_0, u_0)$ directly according to (S1),
- sample $(x_1, u_1)|(y, v) \sim \pi(x_1, u_1|y, v)$ by drawing (x^*, u^*) from a conditional proposal density $q(x^*, u^*|y, v, x_0, u_0)$ and accepting $(x_1, u_1) = (x^*, u^*)$ with the usual Metropolis–Hastings probability or otherwise setting $(x_1, u_1) = (x_0, u_0)$.

The two steps can be identified as Gibbs sampling and Metropolis–Hastings within Gibbs sampling respectively. Next, the proposal density $q(x^*, u^*|y, v, x_0, u_0)$ can be defined as an approximation to $\pi(x^*, u^*|y, v)$, based on an approximation to $\pi(x)$ by a normal density

with an identity variance anchored at x_0 :

$$\begin{aligned}\tilde{\pi}(x; x_0) &\propto \exp \left\{ -U(x_0) - (x - x_0)^T \nabla U(x_0) - \frac{1}{2} (x - x_0)^T (x - x_0) \right\} \\ &\propto \mathcal{N}(x | x_0 - \nabla U(x_0), I).\end{aligned}\tag{S2}$$

Specifically, $\tilde{\pi}(x; x_0)$ is determined such that the gradient of $-\log \tilde{\pi}(x; x_0)$ at x_0 coincides with $\nabla U(x_0)$, the gradient of $U(x) = -\log \pi(x)$ at x_0 . We take $q(x^*, u^* | y, v, x_0, u_0) = \tilde{\pi}(x^*, u^* | y, v; x_0)$, the induced conditional density by (S3) in Lemma S1. This result can be shown by similar calculation as in Gelman et al. (2014, Section 3.5).

Lemma S1 Define $\tilde{\pi}(x, u; x_0) \propto \tilde{\pi}(x; x_0) \exp(-u^T u / 2)$. Then the joint density defined by $\tilde{\pi}(x, u; x_0) \pi(y, v | x, u)$ induces the conditional density

$$\tilde{\pi}(x, u | y, v; x_0) = \mathcal{N}(x, u | \mu_{x_0}, A),\tag{S3}$$

where $\pi(y, v | x, u)$ is as in (S1), and

$$A = (I + S^{-1})^{-1}, \quad \mu_{x_0} = A \left[\begin{pmatrix} x_0 - \nabla U(x_0) \\ \mathbf{0} \end{pmatrix} + S^{-1} \begin{pmatrix} y \\ v \end{pmatrix} \right].$$

Similarly as in Titsias and Papaspiliopoulos (2018), the auxiliary variables (y, v) can be integrated out to obtain a marginal scheme from (x_0, u_0) to (x^*, u^*) as

$$\begin{aligned}q(x^*, u^* | x_0, u_0) &= \int \tilde{\pi}(x^*, u^* | y, v; x_0) \pi(y, v | x_0, u_0) d(y, v) \\ &= \mathcal{N} \left(\begin{pmatrix} x^* \\ u^* \end{pmatrix} \middle| \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - A \begin{pmatrix} \nabla U(x_0) \\ u \end{pmatrix}, AS^{-1}A + A \right),\end{aligned}\tag{S4}$$

where $AS^{-1}A + A = 2A - A^2$ for $A = (I + S^{-1})^{-1}$. Hence the proposal scheme (S4) from the auxiliary variable argument retains the same form as (7). This discussion also confirms the previous observation that when the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$, the proposal (x^*, u^*) in (8) is always accepted, because the normal approximation $\tilde{\pi}(x; x_0)$ becomes exact and hence (x^*, u^*) is obtained from just two-block Gibbs sampling.

There is, however, a caveat in the link between (7) and (S4). Using the auxiliary variables leads the proposal (S4), with the relation $A = (I + S^{-1})^{-1}$. Because S is positive

semi-definite as a variance matrix, this relation imposes the constraint that $A \leq I$. For the proposal scheme (7), it is only required that $\mathbf{0} \leq A \leq 2I$. When $I < A \leq 2I$, the scheme (7) remains valid, but cannot be deduced from (S4). Hence (7) encapsulates a broader class of proposal distributions than directly derived via auxiliary variables.

Next we show that the over-relaxation technique (Adler, 1981; Neal, 1998) can be exploited to define an auxiliary proposal density $q(x^*, u^*|y, v, x_0, u_0)$ more flexible than above, so that the entire class of proposal distributions (7) can be recovered. By over-relaxation based on normal distributions, consider the proposal density

$$q_\alpha(x^*, u^*|y, v, x_0, u_0) = \mathcal{N}(x^*, u^*|\mu_{x_0} + \alpha((x_0^\top, u_0^\top)^\top - \mu_{x_0}), (1 - \alpha^2)A),$$

where μ_{x_0} and A are defined as in Lemma S1, and $-1 \leq \alpha \leq 1$ controls the degree of over-relaxation. Setting $\alpha = 0$ gives the previous choice $q(x^*, u^*|y, v, x_0, u_0) = \tilde{\pi}(x^*, u^*|y, v; x_0)$ and leads to the marginal proposal density (S4).

Lemma S2 *Let $A_\alpha = (1 - \alpha)A$. The marginal proposal density obtained by integrating out (y, v) from $q_\alpha(x^*, u^*|y, v, x_0, u_0)$ is*

$$\begin{aligned} q_\alpha(x^*, u^*|x_0, u_0) &= \int q_\alpha(x^*, u^*|y, v, x_0, u_0)\pi(y, v|x_0, u_0) d(y, v) \\ &= \mathcal{N}\left(\begin{pmatrix} x^* \\ u^* \end{pmatrix} \middle| \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - A_\alpha \begin{pmatrix} \nabla U(x_0) \\ u \end{pmatrix}, 2A_\alpha - A_\alpha^2\right). \end{aligned} \quad (\text{S5})$$

By the preceding result, the marginal scheme (S5) is still of the form (7), with A replaced by A_α . The matrix A_α is determined from α and S as $A_\alpha = (1 - \alpha)(I + S^{-1})^{-1}$. The constraints $-1 \leq \alpha \leq 1$ and $S \geq \mathbf{0}$ imply that $\mathbf{0} \leq A_\alpha \leq 2I$. Conversely, any matrix $\mathbf{0} \leq A \leq 2I$ can be obtained as A_α for some $-1 \leq \alpha \leq 1$ and $S \geq \mathbf{0}$. The choice $A = 2I$ corresponds to the limit case $\alpha = -1$ and $S \rightarrow \infty$. In this sense, the proposal scheme (7) with any choice $\mathbf{0} \leq A \leq 2I$ can be identified as a marginal scheme from the auxiliary variable argument while incorporating over-relaxation.

Finally, as might be noted by readers, the foregoing development (including the over-relaxation) remains valid when the momentum variable u is dropped. In this case, the

proposal density in (S4) reduces to

$$q(x^*|x_0) = \mathcal{N}(x^*|x_0 - A\nabla U(x_0), 2A - A^2),$$

where A is a $k \times k$ symmetric matrix satisfying $\mathbf{0} \leq A \leq I$ before over-relaxation. Taking $A = \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}I = (1 - \sqrt{1 - \epsilon^2})I$ leads to the proposal scheme

$$x^* = x_0 - \frac{\epsilon^2}{1 + \sqrt{1 - \epsilon^2}}\nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(0, \epsilon^2 I), \quad (\text{S6})$$

which is precisely the proposal scheme in modified pMALA with $\Sigma = I$ (or modified MALA). In general, our auxiliary variable argument can be applied with an arbitrary choice of variance matrix Σ , to obtain a proposal scheme in the form

$$x^* = x_0 - A\nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(0, 2A - A\Sigma^{-1}A), \quad (\text{S7})$$

where A is a $k \times k$ symmetric matrix satisfying $A^{-1} \geq \Sigma^{-1}$. Taking $A = \frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}\Sigma = (1 - \sqrt{1 - \epsilon^2})\Sigma$ leads to modified pMALA described in Section 2.

It is informative to compare our schemes with Titsias and Papaspiliopoulos (2018) in the Bayesian setting with $\pi(x) \propto \exp\{-U(x)\} \propto \exp\{\ell(x)\}\mathcal{N}(x|\mathbf{0}, C)$, where $\ell(x)$ is the log-likelihood and C a prior variance. As discussed in Section 2, the proposal scheme (3) in Titsias and Papaspiliopoulos (2018) differs from that in modified pMALA with general Σ , except for equivalence in the special case $C = I$, where both proposal schemes reduce to (S6). This difference can be understood as follows. Given the current value x_0 , the normal approximation of $\pi(x)$ used in Titsias and Papaspiliopoulos (2018) is

$$\begin{aligned} \tilde{\pi}_{\text{TP}}(x; x_0) &\propto \exp \left\{ \ell(x_0) + (x - x_0)^T \nabla \ell(x_0) - \frac{1}{2} x^T C^{-1} x \right\} \\ &\propto \exp \left\{ -(x - x_0)^T \nabla U(x_0) - \frac{1}{2} (x - x_0)^T C^{-1} (x - x_0) \right\}, \end{aligned} \quad (\text{S8})$$

where $\nabla \ell(x_0) = -\nabla U(x_0) + C^{-1}x_0$. Apparently, the normal density (S8) in general differs from (S2) used in our derivation unless $C = I$.

For the second-order algorithm in the Supplement of Titsias and Papaspiliopoulos (2018), the proposal scheme, after correcting a typo to match the first-order scheme (3)

when $G = \mathbf{0}$, can be written as

$$x^* = \frac{2}{\delta} C^\dagger x_0 + C^\dagger (\nabla \ell(x_0) - G x_0) + Z = x_0 - C^\dagger \nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, \frac{2}{\delta} C^{\dagger 2} + C^\dagger), \quad (\text{S9})$$

where $C^\dagger = (\frac{2}{\delta} I + C^{-1} - G)^{-1}$, and G is the Hessian $\nabla^2 \ell(x_0)$ or an approximation. For simplicity, assume that G is independent of x_0 . The corresponding approximation to the variance of the target $\pi(x)$ is then $\Sigma = (C^{-1} - G)^{-1}$. Moreover, the proposal scheme (S9), by direct calculation, can be expressed as (S7) with $\Sigma = (C^{-1} - G)^{-1}$ and $A = C^\dagger = (\Sigma^{-1} + \frac{2}{\delta} I)^{-1}$. Therefore, the second-order algorithm of Titsias and Papaspiliopoulos (2018) and modified pMALA use proposal schemes both in the class (S7), but with different choices of A matrix, after the approximate variance Σ is matched.

I.1 Proof of Lemma S2

Given the current variables (x_0, u_0) , the variables (y, v) are generated as

$$(y, v) | (x_0, u_0) \sim \mathcal{N}((x_0, u_0), S). \quad (\text{S10})$$

The variables (x^*, u^*) are then generated from q_α as

$$(x^*, u^*) | (y, v, x_0, u_0) \sim \mathcal{N}\left((1 - \alpha)\mu_{x_0} + \alpha \begin{pmatrix} x_0 \\ u_0 \end{pmatrix}, (1 - \alpha^2)A\right), \quad (\text{S11})$$

where $-1 \leq \alpha \leq 1$, and

$$A = (I + S^{-1})^{-1}, \quad \mu_{x_0} = A \left(\begin{pmatrix} x_0 - \nabla U(x_0) \\ \mathbf{0} \end{pmatrix} + S^{-1} \begin{pmatrix} y \\ v \end{pmatrix} \right). \quad (\text{S12})$$

Then (x^*, u^*) and (y, v) are jointly normal given (x_0, u_0) and hence $(x^*, u^*) | (x_0, u_0)$ is also normally distributed. It suffices to determine its mean and variance.

First, we compute $\mathbb{E}(x^*, u^* | x_0, u_0)$. By (S10) and (S12),

$$\begin{aligned} \mathbb{E}[\mu_{x_0} | x_0, u_0] &= A \left(\begin{pmatrix} x_0 - \nabla U(x_0) \\ \mathbf{0} \end{pmatrix} + S^{-1} \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} \right) \\ &= A \left((I + S^{-1}) \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - \begin{pmatrix} \nabla U(x_0) \\ u_0 \end{pmatrix} \right) = \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - A \begin{pmatrix} \nabla U(x_0) \\ u_0 \end{pmatrix}. \end{aligned} \quad (\text{S13})$$

Therefore, by (S11) and (S13),

$$\begin{aligned}\mathbb{E}(x^*, u^* | x_0, u_0) &= \mathbb{E}[\mathbb{E}(x, u^* | y, v, x_0, u_0) | x_0, u_0] \\ &= \mathbb{E}\left[(1-\alpha)\mu_{x_0} + \alpha \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} \middle| x_0, u_0\right] = \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} - A_\alpha \begin{pmatrix} \nabla U(x_0) \\ u_0 \end{pmatrix},\end{aligned}$$

where $A_\alpha = (1-\alpha)A$.

Next, we compute $\text{Var}(x^*, u^* | x_0, u_0)$. By (S11)–(S12),

$$\begin{aligned}\text{Var}[\mathbb{E}(x^*, u^* | y, v, x_0, u_0) | x_0, u_0] &= \text{Var}\left[(1-\alpha)\mu_{x_0} + \alpha \begin{pmatrix} x_0 \\ u_0 \end{pmatrix} \middle| x_0, u_0\right] \\ &= (1-\alpha)^2 \text{Var}[\mu_{x_0} | x_0, u_0] = (1-\alpha)^2 AS^{-1}A = A_\alpha S^{-1}A_\alpha,\end{aligned}\tag{S14}$$

$$\begin{aligned}\mathbb{E}[\text{Var}(x^*, u^* | y, v, x_0, u_0) | x_0, u_0] &= \mathbb{E}[(1-\alpha^2)A | x_0, u_0] \\ &= (1-\alpha^2)A = (1+\alpha)A_\alpha.\end{aligned}\tag{S15}$$

Combining (S14) and (S15) yields

$$\begin{aligned}\text{Var}(x^*, u^* | x_0, u_0) &= \mathbb{E}[\text{Var}(x^*, u^* | y, v, x_0, u_0) | x_0, u_0] + \text{Var}[\mathbb{E}(x^*, u^* | y, v, x_0, u_0) | x_0, u_0] \\ &= A_\alpha S^{-1}A_\alpha + (1+\alpha)A_\alpha.\end{aligned}$$

Finally, we show that $A_\alpha S^{-1}A_\alpha + (1+\alpha)A_\alpha = 2A_\alpha - A_\alpha^2$. Because $A = (I + S^{-1})^{-1}$, we have $A(I + S^{-1}) = I$ and hence $A^2 + AS^{-1}A = A$. Then

$$\begin{aligned}(1-\alpha)^2 AS^{-1}A &= (1-\alpha)^2 A - (1-\alpha)^2 A^2 \\ \Rightarrow A_\alpha S^{-1}A_\alpha &= (1-\alpha)A_\alpha - A_\alpha^2 \\ \Rightarrow A_\alpha S^{-1}A_\alpha + (1+\alpha)A_\alpha &= 2A_\alpha - A_\alpha^2.\end{aligned}$$

This completes the proof of Lemma S2.

II Demonstration of validity of UDL

We demonstrate that UDL is valid in leaving the augmented target $\pi(x, u)$ invariant. Similarly as HAMS, by Proposition 3, it suffices to verify that the acceptance probability

stated for UDL in Section 2 can be written in the form of generalized Metropolis–Hastings probability (21) for the associated (forward) proposal density Q .

First, we calculate the generalized Metropolis–Hastings probability (21) with the (forward) proposal density Q from UDL. The proposal scheme in UDL is defined as

Sample $Z_1, Z_2 \sim \mathcal{N}(\mathbf{0}, M)$ independently,

$$u^+ = \sqrt{c}u_0 + \sqrt{1-c}Z_1,$$

$$\tilde{u} = u^+ - \frac{\epsilon}{2}\nabla U(x_0), \quad x^* = x_0 + \epsilon M^{-1}\tilde{u}, \quad u^- = \tilde{u} - \frac{\epsilon}{2}\nabla U(x^*),$$

$$u^* = \sqrt{c}u^- + \sqrt{1-c}Z_2.$$

The noises (Z_1, Z_2) can be expressed as

$$Z_1 = \left(\frac{M}{\epsilon}(x^* - x_0) + \frac{\epsilon}{2}\nabla U(x_0) - \sqrt{c}u_0 \right) (1-c)^{-1/2}, \quad (\text{S16})$$

$$Z_2 = \left(\frac{\sqrt{c}M}{\epsilon}(x_0 - x^*) + \frac{\epsilon\sqrt{c}}{2}\nabla U(x^*) + \sqrt{c}u^* \right) (1-c)^{-1/2}. \quad (\text{S17})$$

Suppose that the mapping above from (x_0, u_0) to (x^*, u^*) is applied from $(x^*, -u^*)$ to $(x_0, -u_0)$, but using new noises (Z_3, Z_4) . By exchanging (x_0, u_0) and $(x^*, -u^*)$, the new noises (Z_3, Z_4) can be calculated as

$$Z_3 = \left(\frac{M}{\epsilon}(x_0 - x^*) + \frac{\epsilon}{2}\nabla U(x^*) + \sqrt{c}u^* \right) (1-c)^{-1/2}, \quad (\text{S18})$$

$$Z_4 = \left(\frac{\sqrt{c}M}{\epsilon}(x^* - x_0) + \frac{\epsilon\sqrt{c}}{2}\nabla U(x_0) - \sqrt{c}u_0 \right) (1-c)^{-1/2}. \quad (\text{S19})$$

Then the forward and backward transitions of the proposals for UDL can be illustrated in a similar manner to (20) as

$$\begin{pmatrix} x_0 \\ u_0 \end{pmatrix} \xrightarrow{(Z_1, Z_2)} \begin{pmatrix} x^* \\ u^* \end{pmatrix}, \quad \begin{pmatrix} x^* \\ -u^* \end{pmatrix} \xrightarrow{(Z_3, Z_4)} \begin{pmatrix} x_0 \\ -u_0 \end{pmatrix}, \quad (\text{S20})$$

where the arrows denote the *same* mapping, depending on (Z_1, Z_2) or (Z_3, Z_4) .

Because (Z_1, Z_2) are the only sources of randomness, the (forward) proposal density from (x_0, u_0) to (x^*, u^*) is

$$\begin{aligned} Q(x^*, u^* | x_0, u_0) &= \mathcal{N}(Z_1 | \mathbf{0}, M)\mathcal{N}(Z_2 | \mathbf{0}, M) \\ &\propto \exp \left(-\frac{1}{2}Z_1^T M^{-1} Z_1 - \frac{1}{2}Z_2^T M^{-1} Z_2 \right). \end{aligned} \quad (\text{S21})$$

Evaluation of the *same* proposal density from $(x^*, -u^*)$ to $(x_0, -u_0)$ gives

$$\begin{aligned} Q(x_0, -u_0 | x^*, -u^*) &= \mathcal{N}(Z_3 | \mathbf{0}, M) \mathcal{N}(Z_4 | \mathbf{0}, M) \\ &\propto \exp \left(-\frac{1}{2} Z_3^T M^{-1} Z_3 - \frac{1}{2} Z_4^T M^{-1} Z_4 \right). \end{aligned} \quad (\text{S22})$$

Using (S16) to (S22), the log ratio of proposal densities is

$$\begin{aligned} \log \left(\frac{Q(x_0, -u_0 | x^*, -u^*)}{Q(x^*, u^* | x_0, u_0)} \right) &= \frac{1}{2} (x^* - x_0)^T (\nabla U(x^*) + \nabla U(x_0)) \\ &- \frac{\epsilon^2}{8} ([\nabla U(x^*)]^T M^{-1} \nabla U(x^*) - [\nabla U(x_0)]^T M^{-1} \nabla U(x_0)) - \frac{1}{2} (u_0^T M^{-1} u_0 - (u^*)^T M^{-1} u^*). \end{aligned} \quad (\text{S23})$$

Furthermore, the log ratio of target densities at (x_0, u_0) and $(x^*, -u^*)$ is

$$\log \left(\frac{\pi(x^*, -u^*)}{\pi(x_0, u_0)} \right) = U(x_0) - U(x^*) + \frac{1}{2} (u_0^T M^{-1} u_0 - (u^*)^T M^{-1} u^*). \quad (\text{S24})$$

From (S23) and (S24), the generalized Metropolis–Hastings probability (21) is

$$\min \left(1, \exp \left\{ U(x_0) - U(x^*) + \frac{(x^* - x_0)^T}{2} (\nabla U(x_0) + \nabla U(x^*)) \right. \right. \\ \left. \left. - \frac{\epsilon^2}{8} ([\nabla U(x^*)]^T M^{-1} \nabla U(x^*) - [\nabla U(x_0)]^T M^{-1} \nabla U(x_0)) \right\} \right). \quad (\text{S25})$$

Second, we show that generalized Metropolis–Hastings probability (S25) reduces to the acceptance probability stated in Section 2:

$$\min \left(1, \exp(H(x_0, u^+) - H(x^*, u^-)) \right). \quad (\text{S26})$$

In fact, direct calculation using $u^- = u^+ - \frac{\epsilon}{2} (\nabla U(x^*) + \nabla U(x_0))$ yields

$$\begin{aligned} (u^-)^T M^{-1} u^- &= (u^+)^T M^{-1} u^+ + \frac{\epsilon^2}{4} (\nabla U(x_0) + \nabla U(x^*))^T M^{-1} (\nabla U(x_0) + \nabla U(x^*)) \\ &- \epsilon (u^+)^T M^{-1} (\nabla U(x_0) + \nabla U(x^*)), \end{aligned}$$

and hence

$$\begin{aligned} \frac{1}{2} (u^+)^T M^{-1} u^+ - \frac{1}{2} (u^-)^T M^{-1} u^- \\ &= \frac{\epsilon}{2} (u^+)^T M^{-1} (\nabla U(x_0) + \nabla U(x^*)) - \frac{\epsilon^2}{8} (\nabla U(x_0) + \nabla U(x^*))^T M^{-1} (\nabla U(x_0) + \nabla U(x^*)) \\ &= \frac{1}{2} (x^* - x_0)^T (\nabla U(x_0) + \nabla U(x^*)) - \frac{\epsilon^2}{8} ([\nabla U(x^*)]^T M^{-1} \nabla U(x^*) - [\nabla U(x_0)]^T M^{-1} \nabla U(x_0)). \end{aligned} \quad (\text{S27})$$

By the definition of the Hamiltonian, we have

$$H(x_0, u^+) - H(x^*, u^-) = U(x_0) - U(x^*) + \frac{1}{2}(u^+)^T M^{-1} u^+ - \frac{1}{2}(u^-)^T M^{-1} u^-.$$

Substituting (S27) into the above, we see that (S26) equals (S25).

III Generalized Metropolis–Hastings sampling

We give a broader definition of generalized Metropolis–Hastings sampling in Section 4, to accommodate both continuous and discrete variables.

Let $\pi(y)$ be a pre-specified probability density function on \mathcal{Y} , with respect to possibly a product of Lebesgue and counting measures. Assume that $J : \mathcal{Y} \rightarrow \mathcal{Y}$ is an invertible mapping, such that for any set $C \subset \mathcal{Y}$ and integrable function h ,

$$\int_{J(C)} \pi(y) dy = \int_C \pi(y) dy, \quad (\text{S28})$$

$$\int_{J(C)} h(J^{-1}y) dy = \int_C h(y) dy, \quad (\text{S29})$$

where J^{-1} denote the inverse mapping of J , and $J(C) = \{Jy : y \in C\}$. While (S28) is restated from (33), condition (S29) is analogous to saying that the Jacobian determinant of J is ± 1 in the case where \mathcal{Y} is Euclidean endowed with the Lebesgue measure. With this interpretation of Jy , generalized Metropolis–Hastings sampling is still defined as in Section 4. More importantly, Proposition 3 can be seen to remain valid, by substituting (S29) for all the change-of-variables calculation in the proof.

Next we show that the irreversible jump sampler (I-Jump) in Ma et al. (2018) can be obtained as a special case of generalized Metropolis-Hastings sampling, when a binary auxiliary variable $s \in \{1, -1\}$ is introduced for sampling from an original target density $\pi(x)$ on \mathcal{X} . Given current variables (x_0, s_0) , an iteration of I-Jump can be described as follows, where $f(\cdot|x_0)$ and $g(\cdot|x_0)$ are two possibly different proposal densities.

Irreversible jump sampler (I-Jump).

- Sample $w \sim \text{Uniform}[0, 1]$.

- If $s_0 = 1$, sample $x^* \sim f(\cdot|x_0)$ and compute

$$\rho(x^*|x_0) = \min\left(1, \frac{\pi(x^*)g(x_0|x^*)}{\pi(x_0)f(x^*|x_0)}\right);$$

else sample $x^* \sim g(\cdot|x_0)$ and compute

$$\rho(x^*|x_0) = \min\left(1, \frac{\pi(x^*)f(x_0|x^*)}{\pi(x_0)g(x^*|x_0)}\right).$$

- If $w < \rho(x^*|x_0)$, then set $(x_1, s_1) = (x^*, s_0)$; else set $(x_1, s_1) = (x_0, -s_0)$.

To recast I-Jump, consider the augmented target density $\pi(x, s) = \pi(x)/2$ on the product space $\mathcal{Y} = \mathcal{X} \times \{1, -1\}$, that is, x and s are independent and s takes value 1 or -1 with equal probabilities. The mapping defined by $J(x, s) = (x, -s)$ satisfies conditions (S28)–(S29). Define the proposal density Q as

$$Q(x^*, s^*|x_0, s_0) = \begin{cases} f(x^*|x_0), & \text{if } s^* = s_0 = 1, \\ g(x^*|x_0), & \text{if } s^* = s_0 = -1, \\ 0, & \text{if } s^* \neq s_0. \end{cases}$$

Then the acceptance probability in I-Jump can be expressed as

$$\rho(x^*, s^*|x_0, s_0) = \min\left(1, \frac{\pi(x^*, -s^*)Q(x_0, -s_0|x^*, -s^*)}{\pi(x_0, s_0)Q(x^*, s^*|x_0, s_0)}\right).$$

by noticing that $s^* = s_0$ and $\pi(x^*, -s^*)/\pi(x_0, s_0) = \pi(x^*)/\pi(x_0)$. Therefore, the I-Jump algorithm can be seen as generalized Metropolis–Hastings sampling.

As a concrete example of I-Jump, Ma et al. (2018) proposed an irreversible MALA (I-MALA) algorithm. The proposal schemes $f(\cdot|x_0)$ and $g(\cdot|x_0)$ are defined as discretizations of irreversible continuous Markov processes. Each proposal scheme can be related to (36) in our G2MS algorithm with y_0 replaced by x_0 :

$$x^* = x_0 - B\nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, B + B^T - BB^T).$$

For $B = \epsilon^2 B_0$ with $\epsilon \approx 0$, the preceding scheme is approximately

$$x^* = x_0 - \epsilon^2(D_0 + C_0)\nabla U(x_0) + Z, \quad Z \sim \mathcal{N}(\mathbf{0}, 2\epsilon^2 D_0), \tag{S30}$$

where $D_0 = (B_0 + B_0^T)/2$ is symmetric and $C_0 = (B_0 - B_0^T)/2$ is skew-symmetric. It is interesting that the form of (S30) matches the proposal schemes derived by discretizing general Markov processes in Ma et al. (2018).

Although both HAMS and I-MALA can be subsumed by generalized Metropolis–Hastings sampling, there remain important differences. The HAMS algorithm uses momentum as an auxiliary variable and hence is able to exploit symmetry in the momentum distribution, whereas I-MALA relies on lifting with a binary variable (Gustafson, 1998; Vucelja, 2016) and needs to split the original variable x to specify symmetric and skew-symmetric matrices D_0 and C_0 when defining proposal schemes based on irreversible Markov processes in x . Further research is desired to compare and connect these algorithms.

IV Proofs

IV.1 Proof of Propositions 1 and 2

The results follow from Proposition 3, by the discussion at the end of Section 4.

IV.2 Proof of Proposition 3

First, the transition kernel $K(y_1|y_0)$ can be expressed as

$$K(y_1|y_0) \, dy_1 = Q(y_1|y_0)\rho(y_1|y_0)dy_1 + (1 - r(y_0))\delta_{Jy_0}(dy_1), \quad (\text{S31})$$

where $r(y_0) = \int Q(y_1|y_0)\rho(y_1|y_0)dy_1$ and δ_y denotes point mass at y . Then for $y_1 \neq Jy_0$,

$$\pi(y_0)K(y_1|y_0) = \pi(y_0)Q(y_1|y_0)\rho(y_1|y_0).$$

Replacing (y_0, y_1) with $(J^{-1}y_1, Jy_0)$ above shows that for $Jy_0 \neq y_1$,

$$\pi(J^{-1}y_1)K(Jy_0|J^{-1}y_1) = \pi(J^{-1}y_1)Q(Jy_0|J^{-1}y_1)\rho(Jy_0|J^{-1}y_1),$$

where $\pi(J^{-1}y_1) = \pi(y_1)$ by the invariance property (33) and

$$\rho(Jy_0|J^{-1}y_1) = \min \left(1, \frac{\pi(y_0)Q(y_1|y_0)}{\pi(J^{-1}y_1)Q(Jy_0|J^{-1}y_1)} \right).$$

Hence (35) holds for $Jy_0 \neq y_1$, because

$$\begin{aligned}\pi(y_0)Q(y_1|y_0)\rho(y_1|y_0) &= \pi(J^{-1}y_1)Q(Jy_0|J^{-1}y_1)\rho(Jy_0|J^{-1}y_1) \\ &= \min(\pi(y_0)Q(y_1|y_0), \pi(J^{-1}y_1)Q(Jy_0|J^{-1}y_1)),\end{aligned}\tag{S32}$$

which holds whether $Jy_0 = y_1$ or not.

The proof that $\pi(y)$ is a stationary distribution is a generalization of Tierney (1994).

It suffices to show that for any set $C \subset \mathcal{Y}$,

$$\int_C \left(\int \pi(y_0)K(y_1|y_0) dy_0 \right) dy_1 = \int_C \pi(y_1) dy_1.\tag{S33}$$

By (S31), the left-hand side of (S33) can be calculated as

$$\begin{aligned}&\int_C \left(\int \pi(y_0)Q(y_1|y_0)\rho(y_1|y_0) dy_0 \right) dy_1 + \int_{J^{-1}(C)} (1 - r(y_0))\pi(y_0) dy_0 \\ &= \int_C \left(\int Q(Jy_0|J^{-1}y_1)\rho(Jy_0|J^{-1}y_1) dy_0 \right) \pi(J^{-1}y_1) dy_1 + \int_{J^{-1}(C)} (1 - r(y_0))\pi(y_0) dy_0 \\ &= \int_C r(J^{-1}y_1)\pi(J^{-1}y_1)|\det(J^{-1})| dy_1 + \int_{J^{-1}(C)} (1 - r(y_0))\pi(y_0) dy_0 \\ &= \int_C r(J^{-1}y_1)\pi(J^{-1}y_1)|\det(J^{-1})| dy_1 + \int_C (1 - r(J^{-1}y_0))\pi(J^{-1}y_0)|\det(J^{-1})| dy_0 \\ &= \int_C \pi(J^{-1}y_1)|\det(J^{-1})| dy_1 = \int_{J(C)} \pi(y_1) dy_1,\end{aligned}$$

which yields the right-hand side of (S33) by the invariance property (33). The first equality follows from (S32), the second from the definition of $r(\cdot)$ and the change of variables, and the third and fifth both from the change of variables.

IV.3 Proof of Corollary 1

The result follows from Corollary 3, by the discussion at the end of Section 4.

IV.4 Proof of Corollary 3

The backward proposal scheme (37) becomes $Jy_0 = (I - A)y^* + Z^*$. The new noise Z^* can be directly calculated using (39) as

$$Z^* = Jy_0 - (I - A)y^* = (2A - A^2)Jy_0 - (I - A)Z,$$

which is distributed as $\mathcal{N}(\mathbf{0}, 2A - A^2)$, if $y_0 \sim \mathcal{N}(\mathbf{0}, I)$, independently of $Z \sim \mathcal{N}(\mathbf{0}, 2A - A^2)$. Hence if $y_0 \sim \mathcal{N}(\mathbf{0}, I)$ and y^* is generated by (39) with Z independent of y_0 , then the conditional density of y^* given y_0 is $p(y^*|y_0) = \mathcal{N}(Z|\mathbf{0}, 2A - A^2)$ and the conditional density of Jy_0 given y^* is $p(Jy_0|y^*) = \mathcal{N}(Z^*|\mathbf{0}, 2A - A^2)$. By the change of variables, the conditional density of y_0 given y^* is also $p(y_0|y^*) = \mathcal{N}(Z^*|\mathbf{0}, 2A - A^2)$ because $|\det(J)| = 1$. Therefore, the acceptance probability (38) reduces to 1, because $\pi(y_0)p(y^*|y_0) = \pi(y^*)p(y_0|y^*)$: both $\pi(y_0)p(y^*|y_0)$ and $\pi(y^*)p(y_0|y^*)$ give the joint density of (y_0, y^*) .

IV.5 Proof of Lemma 1

The HAMS-A proposal described in Section 3.3 is

$$\begin{aligned}\tilde{Z} &= Z - a\nabla U(x_0) + \sqrt{ab}u_0, \quad Z \sim \mathcal{N}(\mathbf{0}, a(2 - a - b)I), \\ x^* &= x_0 + \tilde{Z}, \quad u^* = -u_0 + \sqrt{\frac{b}{a}}\tilde{Z} + \phi(\tilde{Z} + \nabla U(x_0) - \nabla U(x^*)), \\ Z^* &= \tilde{Z} - a\nabla U(x^*) - \sqrt{ab}u^*.\end{aligned}$$

We express x^* , u^* and Z^* in terms of x_0 , u_0 , Z and $\nabla U(x^*)$:

$$x^* = x_0 - \nabla U(x_0) + \sqrt{ab}u_0 + Z, \tag{S34}$$

$$u^* = [\phi - \phi a - \sqrt{ab}]\nabla U(x_0) - \phi \nabla U(x^*) + [\phi\sqrt{ab} + b - 1]u_0 + \left[\phi + \sqrt{\frac{b}{a}}\right]Z, \tag{S35}$$

$$\begin{aligned}Z^* &= [ab + \phi a\sqrt{ab} - \phi\sqrt{ab} - a]\nabla U(x_0) + (\sqrt{ab}\phi - a)\nabla U(x^*) \\ &\quad + [2\sqrt{ab} - \phi ab - b\sqrt{ab}]u_0 + \left[1 - \phi\sqrt{ab} - b\right]Z.\end{aligned} \tag{S36}$$

Suppose that the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, \gamma^{-1}I)$. Then x^* , u^* and Z^* from (S34)–

(S36) can be expressed in terms of only x_0, u_0 and Z as

$$x^* = (-a\gamma + 1)x_0 + \sqrt{ab}u_0 + Z, \quad (\text{S37})$$

$$u^* = \underbrace{[a\phi\gamma^2 - (a\phi + \sqrt{ab})\gamma]x_0}_{(i)} + \underbrace{[-\phi\sqrt{ab}\gamma + \phi\sqrt{ab} + b - 1]u_0}_{(iii)} + \underbrace{\left[-\phi\gamma + \phi + \sqrt{\frac{b}{a}} \right] Z}_{(v)}, \quad (\text{S38})$$

$$\begin{aligned} Z^* &= \underbrace{[(a^2 - \phi a \sqrt{ab})\gamma^2 + (ab - 2a + \phi a \sqrt{ab})\gamma]x_0}_{(ii)} \\ &\quad + \underbrace{[(\phi ab - a \sqrt{ab})\gamma + 2\sqrt{ab} - b\sqrt{ab} - \phi ab]u_0}_{(iv)} \\ &\quad + \underbrace{[(\phi\sqrt{ab} - a)\gamma + 1 - b - \phi\sqrt{ab}]Z}_{(vi)}. \end{aligned} \quad (\text{S39})$$

The quantity inside the exponential in (28) is

$$\begin{aligned} H(x_0, u_0) - H(x^*, u^*) &+ \frac{Z^T Z - (Z^*)^T Z^*}{2a(2 - a - b)} \\ &= \frac{\gamma}{2} x_0^T x_0 - \frac{\gamma}{2} (x^*)^T x^* + \frac{1}{2} u_0^T u_0 - \frac{1}{2} (u^*)^T u^* + \frac{Z^T Z}{2a(2 - a - b)} - \frac{(Z^*)^T Z^*}{2a(2 - a - b)}. \end{aligned} \quad (\text{S40})$$

Substituting (S37)–(S39) into the above shows that (S40) can be expressed as a quadratic form in x_0, u_0 and Z :

$$(x_0^T, u_0^T, Z^T) G(\gamma) (x_0^T, u_0^T, Z^T)^T,$$

where $G(\gamma)$ is a 3×3 block matrix. For $i, j = 1, 2, 3$, the (i, j) th block of $G(\gamma)$ is of the form $g_{ij}(\gamma)I$, where $g_{ij}(\gamma)$ is a scalar, polynomial of γ , with coefficients depending on (a, b, ϕ) .

Now we compute the coefficients of the leading terms (terms corresponding to highest power of γ) of $g_{11}(\gamma)$, $g_{22}(\gamma)$ and $g_{33}(\gamma)$. Because we focus on only the leading terms, it is sufficient to examine (S37)–(S39) and account for the coefficients of x_0, u_0, Z , labeled as $(i), \dots, (v)$, which lead to the highest power of γ in $g_{11}(\gamma)$, $g_{22}(\gamma)$ and $g_{33}(\gamma)$. The coefficient

of the leading term of $g_{11}(\gamma)$ associated with $x_0^T x_0$ is

$$\begin{aligned}
& -\frac{(i)^2}{2} - \frac{(ii)^2}{2a(2-a-b)} = -\frac{1}{2}(a\phi)^2\gamma^4 - \frac{(a^2 - \phi a \sqrt{ab})^2 \gamma^4}{2a(2-a-b)} \\
& = \frac{\gamma^4}{2(2-a-b)}(-a^2\phi^2(2-a-b) + 2\phi a^2\sqrt{ab} - \phi^2 a^2 b - a^3) \\
& = \frac{\gamma^4 a^2}{2(2-a-b)}(\phi^2(a-2) + \phi 2\sqrt{ab} - a).
\end{aligned} \tag{S41}$$

The coefficient of the leading term of $g_{22}(\gamma)$ associated with $u_0^T u_0$ is

$$\begin{aligned}
& -\frac{(iii)^2}{2} - \frac{(iv)^2}{2a(2-a-b)} = -\frac{1}{2}(\phi\sqrt{ab})^2\gamma^2 - \frac{(\phi ab - a\sqrt{ab})^2 \gamma^2}{2a(2-a-b)} \\
& = \frac{\gamma^2}{2(2-a-b)}(2\phi ab\sqrt{ab} - a^2 b - \phi^2 ab^2 - 2\phi^2 ab + \phi^2 a^2 b + \phi^2 ab^2) \\
& = \frac{\gamma^2 ab}{2(2-a-b)}(\phi^2(a-2) + \phi 2\sqrt{ab} - a).
\end{aligned} \tag{S42}$$

The coefficient of the leading term of $g_{33}(\gamma)$ associated with $Z^T Z$ is

$$\begin{aligned}
& -\frac{(v)^2}{2} - \frac{(vi)^2}{2a(2-a-b)} = -\frac{1}{2}\phi^2\gamma^2 - \frac{(\phi\sqrt{ab} - a)^2 \gamma^2}{2a(2-a-b)} \\
& = \frac{\gamma^2}{2(2-a-b)}(2\phi\sqrt{ab} - a - \phi^2 b - 2\phi^2 + a\phi^2 + b\phi^2) \\
& = \frac{\gamma^2}{2(2-a-b)}(\phi^2(a-2) + \phi 2\sqrt{ab} - a).
\end{aligned} \tag{S43}$$

Notice that (S41)–(S43) involve ϕ only through the same quadratic function of ϕ :

$$h(\phi) = \phi^2(a-2) + \phi 2\sqrt{ab} - a.$$

For $a > 0, b \geq 0$ and $a+b \leq 2$, we have $h(\phi) \leq 0$, because $(2\sqrt{ab})^2 + 4a(a-2) = 4a(a+b-2) \leq 0$. Hence $|h(\phi)|$ is minimized at $\phi = -2\sqrt{ab}/(2(a-2)) = \sqrt{ab}/(2-a)$.

IV.6 Proof of Lemma 2

We use the following choice of A in (10): $a_1 = 2-a$, $a_2 = \sqrt{ab}$, $a_3 = 2-b$ with the constraints on a, b that $a > 0, b \geq 0$ and $a+b \leq 2$. The noise terms are proportional: $Z_2 = -\sqrt{b/a}Z_1$. The new noises Z_1^* and Z_2^* , defined by (15), (24), and (17), can be expressed in terms of

$u_0, \nabla U(x_0), \nabla U(x^*)$ and Z_1 as

$$\begin{aligned} Z_1^* &= \underbrace{\sqrt{ab}(b - \phi\sqrt{ab})}_{\theta_1} u_0 + \underbrace{(a + ab - 2 - \phi(a - 1)\sqrt{ab})}_{\theta_2} \nabla U(x_0) \\ &\quad + \underbrace{(a - 2 + \phi\sqrt{ab})}_{\theta_3} \nabla U(x^*) + \underbrace{(b + 1 - \phi\sqrt{ab})}_{\theta_4} Z_1, \\ Z_2^* &= \underbrace{(2 - b)(b - \phi\sqrt{ab})}_{\psi_1} u_0 + \underbrace{(\sqrt{ab}(1 - b) - \phi(a - 1)(2 - b))}_{\psi_2} \nabla U(x_0) \\ &\quad + \underbrace{(-\sqrt{ab} + \phi(2 - b))}_{\psi_3} \nabla U(x^*) + \underbrace{(\sqrt{b/a}(1 - b) - \phi(2 - b))}_{\psi_4} Z_1. \end{aligned}$$

Suppose there exists $r \in \mathbb{R}$ such that $Z_2^* = rZ_1^*$ for arbitrary values of x_0, u_0 and Z_1 . Then the coefficients, denoted as $\theta_1, \dots, \theta_4, \psi_1, \dots, \psi_4$, satisfy

$$r\theta_1 = \psi_1, \quad r\theta_2 = \psi_2, \quad r\theta_3 = \psi_3, \quad r\theta_4 = \psi_4. \quad (\text{S44})$$

We study the following possibilities.

First, suppose that $\theta_1 \neq 0$. Then $r = \frac{\psi_1}{\theta_1} = \frac{2-b}{\sqrt{ab}}$ by (S44). Substituting this into $r\theta_4 = \psi_4$ in (S44) yields

$$\begin{aligned} r\theta_4 = \psi_4 &\Rightarrow \frac{2-b}{\sqrt{ab}}(b + 1 - \phi\sqrt{ab}) = \sqrt{b/a}(1 - b) - \phi(2 - b) \\ &\Rightarrow \frac{(2-b)(b+1)}{\sqrt{ab}} = \sqrt{b/a}(1 - b) \Rightarrow (2-b)(b+1) = b(1 - b) \\ &\Rightarrow b - b^2 + 2 = b - b^2 \Rightarrow 0 = 2, \end{aligned}$$

which is a contradiction. Hence $\theta_1 = \psi_1 = 0$, which gives two possibilities: either $b = 0$ or $\phi = \sqrt{b/a}$.

Next suppose that $b = 0$. Then $\theta_4 = 1$ and $\psi_4 = -2\phi$, and hence $r = \psi_4/\theta_4 = -2\phi$ by (S44). Moreover, $\theta_2 = a - 2$ and $\psi_2 = -2\phi(a - 1)$, and

$$r\theta_2 = \psi_2 \Rightarrow -2\phi(a - 2) = -2\phi(a - 1),$$

which implies that $\phi = 0$. Thus if $b = 0$, then $\phi = 0$ as well. This gives the trivial case that $r = 0$ and $Z_2^* \equiv 0$.

Finally suppose that $\phi = \sqrt{b/a}$. Then $Z_2^* = rZ_1^*$ is satisfied with $r = -\sqrt{b/a}$ by the following calculation:

$$\theta_1 = \psi_1 = 0,$$

$$\theta_2 = a + ab - 2 - b(a - 1) = a + b - 2,$$

$$\psi_2 = \sqrt{ab}(1 - b) - \sqrt{\frac{b}{a}}(a - b)(2 - b) = -\sqrt{\frac{b}{a}}(a + b - 2) = r\theta_2,$$

$$\theta_3 = a - 2 + b, \quad \psi_3 = -\sqrt{ab} + \sqrt{\frac{b}{a}}(2 - b) = -\sqrt{\frac{b}{a}}(b - 2 + a) = r\theta_3,$$

$$\theta_4 = b + 1 - b = 1, \quad \psi_4 = \sqrt{\frac{b}{a}}(1 - b) - \sqrt{\frac{b}{a}}(2 - b) = -\sqrt{\frac{b}{a}} = r\theta_4.$$

Therefore $Z_2^* = rZ_1^*$ if and only if $r = -\sqrt{b/a}$ and $\phi = \sqrt{b/a}$, which also includes the trivial case, $r = \phi = b = 0$.

IV.7 Proof of Lemma 3

By the rejection-free property, $(x_1, u_1) = (x^*, u^*)$ when the target density $\pi(x)$ is $\mathcal{N}(\mathbf{0}, I)$.

We give a proof for HAMS-A and HAMS-B separately.

For HAMS-A, the lag-1 auto-covariance matrix is

$$C_A = \text{Cov}((x^*, u^*), (x_0, u_0)) = \begin{pmatrix} (1-a)I & \sqrt{ab}I \\ -\sqrt{ab}I & (b-1)I \end{pmatrix}.$$

The eigenvalues of C_A are the eigenvalues of C_A with $I = 1$, each with multiplicities k . Henceforth we assume $I = 1$. The two eigenvalues of C_A are

$$\lambda_1 = \frac{1}{2}(b - a + \sqrt{\Delta}), \quad \lambda_2 = \frac{1}{2}(b - a - \sqrt{\Delta}),$$

where

$$\Delta = (a + b - 2)^2 - 4ab = \{2 - (\sqrt{a} - \sqrt{b})^2\}\{2 - (\sqrt{a} + \sqrt{b})^2\}.$$

Given $a \in (0, 2)$, we show that the choice of $b \in (0, 2 - a)$ which minimizes $\max(|\lambda_1|, |\lambda_2|)$ is $b^* = (\sqrt{2} - \sqrt{a})^2$, where $|\cdot|$ denotes the modulus. For this choice b^* , $\Delta = 0$ and the two eigenvalues are identical, $\lambda_1^* = \lambda_2^* = 1 - \sqrt{2a}$. We distinguish three cases.

(i) Suppose $(\sqrt{a} + \sqrt{b})^2 > 2$. Then λ_1 and λ_2 are complex, and

$$\begin{aligned} |\lambda_1|^2 = |\lambda_2|^2 &= \lambda_1 \lambda_2 = b + a - 1 \\ &> (\sqrt{2} - \sqrt{a})^2 + a - 1 = (\sqrt{2a} - 1)^2 = \lambda_1^{*2}. \end{aligned}$$

(ii) Suppose $(\sqrt{a} + \sqrt{b})^2 < 2$ and $b \geq a$. Then $\lambda_1 (> 0)$ and λ_2 are real, and $\max(|\lambda_1|, |\lambda_2|) = \lambda_1$. For fixed a , the derivative of λ_1 with respect to b is

$$\frac{d\lambda_1}{db} = \frac{1}{2} \left(1 + \frac{b-a-2}{\sqrt{\Delta}} \right) \leq \frac{1}{2} \frac{(2-a-b)+(b-a-2)}{\sqrt{\Delta}} = \frac{-a}{\sqrt{\Delta}} < 0,$$

where the first inequality uses $\sqrt{\Delta} \leq 2 - a - b$. Then λ_1 is decreasing in b , which is upper-bounded by $b^* = (\sqrt{2} - \sqrt{a})^2$. Hence $\lambda_1 > \lambda_1^*$.

(iii) Suppose $(\sqrt{a} + \sqrt{b})^2 < 2$ and $b \leq a$. Then λ_1 and $\lambda_2 (< 0)$ are real, and $\max(|\lambda_1|, |\lambda_2|) = -\lambda_2$. For fixed a , the derivative of λ_2 with respect to b is

$$\frac{d\lambda_2}{db} = \frac{1}{2} \left(1 - \frac{b-a-2}{\sqrt{\Delta}} \right) = \frac{1}{2} \left(1 + \frac{2+a-b}{\sqrt{\Delta}} \right) > 0.$$

Then λ_2 is increasing in b , which is upper-bounded by $\min(a, b^*)$. If $b^* \leq a$, then $|\lambda_2| = -\lambda_2 > -\lambda_2^* = |\lambda_2^*|$. If $a < b^*$, then $|\lambda_2| = -\lambda_2$ is greater than the value of $-\lambda_2$ corresponding $b = a$, which is identical to the value of λ_1 (due to $b = a$) and still greater than $|\lambda_1^*|$ by the conclusion from (iii).

Combining the three cases shows that $\max(|\lambda_1|, |\lambda_2|) \geq |\lambda_1^*| = |\lambda_2^*|$.

For HAMS-B, we work with the equations (13)–(14) with $a_1 = 2 - a$, $a_3 = 2 - b$ and $a_2 = \sqrt{ab}$, that is, before the reparametrization $ab = \tilde{a}\tilde{b}$ and $a(2 - a - b) = \tilde{a}(2 - \tilde{a} - \tilde{b})$. Then the lag-1 auto-covariance matrix is

$$C_B = \text{Cov}((x^*, u^*), (x_0, u_0)) = \begin{pmatrix} (-1+a)I & \sqrt{ab}I \\ -\sqrt{ab}I & (1-b)I \end{pmatrix}.$$

The eigenvalues of C_B are the same as those of C_A . Hence the maximum modulus of eigenvalues is also minimized by the choice $b = (\sqrt{2} - \sqrt{a})^2$. By the reparametrization $ab = \tilde{a}\tilde{b}$ and $a(2 - a - b) = \tilde{a}(2 - \tilde{a} - \tilde{b})$, the resulting choice is $\tilde{b} = \frac{\tilde{a}(2-\tilde{a})}{(\sqrt{2}+\sqrt{2-\tilde{a}})^2}$, which gives the desired expression with (\tilde{a}, \tilde{b}) relabeled as (a, b) .

Algorithm 4: HAMS-A/HAMS-B (with preconditioning non-simplified)

Initialize x_0, u_0

for $t = 0, 1, 2, \dots, N_{iter}$ **do**

 Sample $w \sim \text{Uniform}[0, 1]$ and $\zeta \sim \mathcal{N}(\mathbf{0}, I)$

 Transform $\tilde{x}_t = L^T x_t$

$$\tilde{x}^* = \tilde{x}_t - aL^{-1}\nabla U(x_t) + \sqrt{ab}u_t + \sqrt{a(2-a-b)}\zeta$$

 Propose $x^* = (L^T)^{-1}\tilde{x}^*$

if HAMS-A **then**

$$\begin{aligned} &\text{Propose } u^* = \left(\frac{2b}{2-a} - 1\right)u_t - \frac{\sqrt{ab}}{2-a}L^{-1}(\nabla U(x_t) + \nabla U(x^*)) + \frac{2\sqrt{b(2-a-b)}}{2-a}\zeta \\ &\quad \zeta^* = \left(1 - \frac{2b}{2-a}\right)\zeta - \frac{\sqrt{a(2-a-b)}}{2-a}L^{-1}(\nabla U(x_t) + \nabla U(x^*)) + \frac{2\sqrt{b(2-a-b)}}{2-a}u_t \end{aligned}$$

end

if HAMS-B **then**

$$\begin{aligned} &\text{Propose } u^* = u_t - \frac{\sqrt{ab}}{2-a}L^{-1}(\nabla U(x_t) + \nabla U(x^*)) \\ &\quad \zeta^* = \zeta - \frac{\sqrt{a(2-a-b)}}{2-a}L^{-1}(\nabla U(x_t) + \nabla U(x^*)) \end{aligned}$$

end

$$\rho = \exp \left\{ H(x_t, u_t) - H(x^*, u^*) + \frac{1}{2}\zeta^T\zeta - \frac{1}{2}(\zeta^*)^T\zeta^* \right\}$$

if $w < \min(1, \rho)$ **then**

$$| \quad (x_{t+1}, u_{t+1}) = (x^*, u^*) \quad \# \text{ Accept}$$

else

$$| \quad (x_{t+1}, u_{t+1}) = (x_t, -u_t) \quad \# \text{ Reject}$$

end

end

IV.8 Simplification of preconditioning for Algorithm 3

As discussed in Section 3.6 for preconditioning, we apply the linear transformations $\tilde{x} = L^T x$ and $\nabla U(\tilde{x}) = L^{-1}\nabla U(x)$ to HAMS-A/B in Algorithm 2. We show the the resulting algorithm, stated as Algorithm 4 here, can be rearranged in an equivalent but computationally more efficient form as Algorithm 3.

Suppose that the equivalence holds for (x_t, u_t) . By the relation $\nabla U(\tilde{x}_t) = L^{-1}\nabla U(x_t)$

and the definition of ξ in Algorithm 3, we have

$$\begin{aligned}\tilde{x}^* &= \tilde{x}_t - a\nabla U(\tilde{x}_t) + \xi \\ &= \tilde{x}_t - aL^{-1}\nabla U(x_t) + \sqrt{ab}u_t + \sqrt{a(2-a-b)}\zeta.\end{aligned}$$

Hence, when the proposal is accepted, $x_{t+1} = x^* = (L^\top)^{-1}\tilde{x}^*$ in both algorithms. By the relation $\tilde{\xi} = \nabla U(\tilde{x}) + L^{-1}\nabla U(x^*) = L^{-1}(\nabla U(x_t) + \nabla U(x^*))$, we see that when the proposal is accepted, the expressions of u_{t+1} are the same in both algorithms. When the proposal is rejected, $(x_{t+1}, u_{t+1}) = (x_t, -u_t)$ is also the same in the two algorithms.

To show the equivalence holds for (x_{t+1}, u_{t+1}) , it remains to check that the acceptance probabilities are equal in the two algorithms. We need to show

$$U(x_t) - U(x^*) + \frac{1}{2-a}(\tilde{\xi})^\top(\xi - \frac{a}{2}\tilde{\xi}) = H(x_t, u_t) - H(x^*, u^*) + \frac{1}{2}\zeta^\top\zeta - \frac{1}{2}(\zeta^*)^\top\zeta^*,$$

which is equivalent to

$$\frac{2}{2-a}(\tilde{\xi})^\top(\xi - \frac{a}{2}\tilde{\xi}) = u_t^\top u_t - (u^*)^\top u^* + \zeta^\top\zeta - (\zeta^*)^\top\zeta^*,$$

because $H(x_t, u_t) - H(x^*, u^*) = U(x_t) - U(x^*) + \frac{1}{2}u_t^\top u_t - \frac{1}{2}(u^*)^\top u^*$.

Consider the algorithm HAMS-B. We use the following fact

$$u_t^\top u_t - (u^*)^\top u^* = (u_t - u^*)^\top(u_t + u^*), \quad \zeta^\top\zeta - (\zeta^*)^\top\zeta^* = (\zeta_t - \zeta^*)^\top(\zeta_t + \zeta^*). \quad (\text{S45})$$

By direct calculation, we have

$$u_t - u^* = \frac{\sqrt{ab}}{2-a}L^{-1}(\nabla U(x_t) + \nabla U(x^*)) = \frac{\sqrt{ab}}{2-a}\tilde{\xi}, \quad (\text{S46})$$

$$(u_t - u^*)^\top(u_t + u^*) = \frac{\sqrt{ab}}{2-a}(\tilde{\xi})^\top\left(2u_t - \frac{\sqrt{ab}}{2-a}\tilde{\xi}\right), \quad (\text{S47})$$

and

$$\zeta - \zeta^* = \frac{\sqrt{a(2-a-b)}}{2-a}L^{-1}(\nabla U(x_t) + \nabla U(x^*)) = \frac{\sqrt{a(2-a-b)}}{2-a}\tilde{\xi}, \quad (\text{S48})$$

$$(\zeta - \zeta^*)^\top(\zeta + \zeta^*) = \frac{\sqrt{a(2-a-b)}}{2-a}(\tilde{\xi})^\top\left(2\zeta - \frac{\sqrt{a(2-a-b)}}{2-a}\tilde{\xi}\right). \quad (\text{S49})$$

Combining (S45)–(S48) yields

$$\begin{aligned}
& u_t^T u_t - (u^*)^T u^* + \zeta^T \zeta - (\zeta^*)^T \zeta^* \\
&= (\tilde{\xi})^T \left(\frac{2\sqrt{ab}}{2-a} u_t + \frac{2\sqrt{a(2-a-b)}}{2-a} \zeta - \left(\frac{ab}{(2-a)^2} + \frac{a(2-a-b)}{(2-a)^2} \right) \tilde{\xi} \right) \\
&= \frac{2}{2-a} (\tilde{\xi})^T \left(\sqrt{ab} u_t + \sqrt{a(2-a-b)} \zeta - \frac{a}{2} \tilde{\xi} \right) \\
&= \frac{2}{2-a} (\tilde{\xi})^T \left(\xi - \frac{a}{2} \tilde{\xi} \right).
\end{aligned} \tag{S50}$$

Hence the acceptance probabilities match for HAMS-B in Algorithms 3 and 4.

Finally consider the algorithm HAMS-A. Define intermediate variables

$$\begin{aligned}
u^\dagger &= \left(\frac{2b}{2-a} - 1 \right) u_t + \frac{2\sqrt{b(2-a-b)}}{2-a} \zeta, \\
\zeta^\dagger &= \left(1 - \frac{2b}{2-a} \right) \zeta + \frac{2\sqrt{b(2-a-b)}}{2-a} u_t.
\end{aligned}$$

Then the following identities hold:

$$(u^\dagger)^T u^\dagger + (\zeta^\dagger)^T \zeta^\dagger = u_t^T u_t + \zeta^T \zeta, \tag{S51}$$

$$\sqrt{ab} u^\dagger + \sqrt{a(2-a-b)} \zeta^\dagger = \sqrt{ab} u_t + \sqrt{a(2-a-b)} \zeta (= \xi). \tag{S52}$$

Identity (S51) follows, because after expanding the inner products on the left hand side, the cross terms cancel out and the squared terms have coefficients

$$\left(\frac{2b}{2-a} - 1 \right)^2 + \left(\frac{2\sqrt{b(2-a-b)}}{2-a} \right)^2 = 1.$$

Identity (S52) follows because by direct calculation

$$\begin{aligned}
u^\dagger - u_t &= \frac{2\sqrt{2-a-b}}{2-a} (\sqrt{2-a-b} u_t + \sqrt{b} \zeta), \\
\zeta^\dagger - \zeta &= \frac{2\sqrt{b}}{2-a} (-\sqrt{b} \zeta + \sqrt{2-a-b} u_t).
\end{aligned}$$

Moreover, it can be verified by definition that

$$u^\dagger - u^* = \frac{\sqrt{ab}}{2-a} \tilde{\xi}, \quad \zeta^\dagger - \zeta^* = \frac{\sqrt{a(2-a-b)}}{2-a} \tilde{\xi}.$$

Then (S45)–(S50) remain valid with u_t and ζ replaced by u^\dagger and ζ^\dagger . From these equations together with the identities (S51)–(S52), we find

$$\begin{aligned} u_t^T u_t - (u^*)^T u^* + \zeta^T \zeta - (\zeta^*)^T \zeta^* \\ = (u^\dagger)^T u^\dagger - (u^*)^T u^* + (\zeta^\dagger)^T \zeta^\dagger - (\zeta^*)^T \zeta^* \\ = \frac{2}{2-a} (\tilde{\xi})^T \left(\sqrt{ab} u^\dagger + \sqrt{a(2-a-b)} \zeta^\dagger - \frac{a}{2} \tilde{\xi} \right) \\ = \frac{2}{2-a} (\tilde{\xi})^T \left(\xi - \frac{a}{2} \tilde{\xi} \right). \end{aligned}$$

Hence the acceptance probabilities match for HAMS-A in Algorithms 3 and 4.

V Details for simulation studies

V.1 Expressions for stochastic volatility model

The stochastic volatility model is defined as

$$\begin{aligned} x_t &= \phi x_{t-1} + \eta_t, \quad t = 2, \dots, T, \quad x_1 \sim \mathcal{N}\left(0, \frac{\sigma^2}{1-\phi^2}\right), \\ y_t &= z_t \beta \exp(x_t/2), \quad z_t \stackrel{iid}{\sim} \mathcal{N}(0, 1), \quad \eta_t \stackrel{iid}{\sim} \mathcal{N}(0, \sigma^2), \quad t = 1, \dots, T. \end{aligned}$$

Denote $\mathbf{x} = (x_1, \dots, x_T)^T$, $\mathbf{y} = (y_1, \dots, y_T)^T$, $\mathbf{z} = (z_1, \dots, z_T)^T$ and $\theta = (\beta, \sigma, \phi)^T$. The joint density of $(\mathbf{x}, \mathbf{y}, \theta)$ is

$$\begin{aligned} p(\mathbf{x}, \mathbf{y}, \theta) &= \pi(\theta) \cdot p(x_1) \underbrace{\prod_{t=2}^T p(x_t | x_{t-1}, \phi, \sigma)}_{\mathcal{N}(\mathbf{x} | \mathbf{0}, C)} \underbrace{\prod_{t=1}^T p(y_t | x_t, \beta)}_{\mathcal{N}(\mathbf{y} | \mathbf{0}, \beta^2 \exp(\mathbf{x}))} \\ &\propto \pi(\theta) |\det(C)|^{-1/2} \exp \left\{ -\frac{1}{2} \mathbf{x}^T C^{-1} \mathbf{x} \right\} \beta^{-T} \exp \left\{ -\frac{1}{2} \sum_{t=1}^T (x_t + \beta^{-2} y_t^2 \exp(-x_t)) \right\}. \end{aligned}$$

The matrix C and its inverse are given by

$$C = \frac{\sigma^2}{1-\phi^2} \begin{pmatrix} 1 & \phi & \phi^2 & \cdots & \phi^{T-2} & \phi^{T-1} \\ \phi & 1 & \phi & \cdots & \phi^{T-3} & \phi^{T-2} \\ \phi^2 & \phi & 1 & \cdots & \phi^{T-4} & \phi^{T-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \phi^{T-2} & \phi^{T-3} & \phi^{T-4} & \cdots & 1 & \phi \\ \phi^{T-1} & \phi^{T-2} & \phi^{T-3} & \cdots & \phi & 1 \end{pmatrix}$$

$$\iff C^{-1} = \frac{1}{\sigma^2} \begin{pmatrix} 1 & -\phi & 0 & \cdots & 0 & 0 \\ -\phi & 1 + \phi^2 & -\phi & \cdots & 0 & 0 \\ 0 & -\phi & 1 + \phi^2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 + \phi^2 & -\phi \\ 0 & 0 & 0 & \cdots & -\phi & 1 \end{pmatrix}.$$

The conditional posterior of the latent variables is

$$p(\mathbf{x}|\mathbf{y}, \theta) \propto \exp \left\{ -\frac{1}{2} \mathbf{x}^T C^{-1} \mathbf{x} \right\} \exp \left\{ -\frac{1}{2} \sum_{t=1}^T (x_t + \beta^{-2} y_t^2 \exp(-x_t)) \right\}.$$

Then the negative log-density (or potential function) is

$$U(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T C^{-1} \mathbf{x} + \frac{1}{2} \sum_{t=1}^T (x_t + \beta^{-2} y_t^2 \exp(-x_t)),$$

where dependency on (\mathbf{y}, θ) is suppressed in the notation. The gradient is

$$\nabla U(\mathbf{x}) = C^{-1} \mathbf{x} - \frac{1}{2} \beta^{-2} \mathbf{y} \exp(-\mathbf{x}) + \frac{1}{2} \mathbf{1},$$

where $\mathbf{1}$ is a vector of all 1's. The hessian is

$$\nabla^2 U(\mathbf{x}) = C^{-1} + \frac{1}{2} \text{diag}[\beta^{-2} \mathbf{y}^2 \exp(-\mathbf{x})].$$

The square \mathbf{y}^2 is taken component-wise. Using the relation between \mathbf{y} and \mathbf{x} , the diagonal elements in the second term can be expressed as

$$\beta^{-2} \mathbf{y}^2 \exp(-\mathbf{x}) = \beta^{-2} \exp(-\mathbf{x}) \mathbf{z}^2 \beta^2 \exp(\mathbf{x}) = \mathbf{z}^2.$$

Hence

$$\mathbb{E}[\nabla^2 U(\mathbf{x})] = C^{-1} + \frac{1}{2}I,$$

which leads to the preconditioning in Section 5.1. The expectation above is taken over the marginal distribution of \mathbf{z} .

For the parameters, the priors are

$$\pi(\beta) \propto \beta^{-1}, \quad \sigma^2 \sim \text{Inv-}\chi^2(10, 0.05), \quad \frac{\phi + 1}{2} \sim \text{Beta}(20, 1.5).$$

Then σ and ϕ are also transformed by $\sigma = \exp(\gamma)$ and $\phi = \tanh(\alpha)$. The resulting potential for the transformed parameters is

$$U(\beta, \alpha, \gamma) = (T+1) \log \beta - 20.5 \log(1 + \tanh \alpha) - 2 \log(1 - \tanh \alpha) \frac{1}{2} \mathbf{x}^T C^{-1} \mathbf{x} + \frac{1}{2} \sum_{t=1}^T \beta^{-2} y_t^2 \exp(-x_t),$$

where dependency on (\mathbf{y}, \mathbf{x}) is suppressed in the notation. The gradient is

$$\begin{aligned} \frac{\partial U(\beta, \alpha, \gamma)}{\partial \beta} &= \frac{T+1}{\beta} - \frac{\sum_{t=1}^T y_t^2 \exp(-x_t)}{\beta^3}, \\ \frac{\partial U(\beta, \alpha, \gamma)}{\partial \alpha} &= 22.5 \tanh \alpha - 18.5 - \exp(-2\gamma) x_1^2 \tanh \alpha (1 - \tanh^2 \alpha), \\ &\quad - \exp(-2\gamma) \sum_{t=2}^T (x_t - \tanh \alpha x_{t-1}) x_{t-1} (1 - \tanh^2 \alpha), \\ \frac{\partial U(\beta, \alpha, \gamma)}{\partial \gamma} &= -\mathbf{x}^T C^{-1} \mathbf{x} - \frac{1}{2} \exp(-2\gamma) + 10 + T. \end{aligned}$$

Finally the expected hessian computed with respect to the marginals of \mathbf{x} and \mathbf{z} is

$$\mathbb{E}[\nabla^2 U(\beta, \alpha, \gamma)] = \begin{pmatrix} (2T-1)/\beta & 0 & 0 \\ 0 & \exp(-2\gamma) + 2T & 2 \tanh \alpha \\ 0 & 2 \tanh \alpha & 21.5 - 19.5 \tanh^2 \alpha + (T-1)(1 - \tanh^2 \alpha) \end{pmatrix}.$$

When sampling the parameters, we use $M = \Sigma^{-1} = \mathbb{E}[\nabla^2 U(\beta, \alpha, \gamma)]$ for preconditioning.

V.2 Expressions for log-Gaussian Cox model

Denote $\mathbf{x} = (x_{ij}), \mathbf{y} = (y_{ij}), i, j = 1, \dots, m$ and let C be the matrix corresponding to the covariance function as described in Section 5.2. The joint posterior density is

$$p(\mathbf{x}, \sigma^2, \beta | \mathbf{y}) \propto \pi(\sigma^2) \pi(\beta) (\det|C|)^{-1/2} \exp\left\{-\frac{1}{2} \mathbf{x}^\top C^{-1} \mathbf{x}\right\} \exp\left\{\sum_{i,j} (y_{ij}(x_{ij} + \mu) - n^{-1} \exp(x_{ij} + \mu))\right\}.$$

The potential function from the conditional posterior of the latent variables given $(\mathbf{y}, \sigma^2, \beta)$ is

$$U(\mathbf{x}) = \frac{1}{2} \mathbf{x}^\top C^{-1} \mathbf{x} - \sum_{i,j} (y_{ij} x_{ij} - n^{-1} \exp(x_{ij} + \mu)),$$

where dependency on $(\mathbf{y}, \sigma^2, \beta)$ is suppressed in the notation. The gradient is

$$\nabla U(\mathbf{x}) = C^{-1} \mathbf{x} - \mathbf{y} + n^{-1} \exp(\mathbf{x} + \mu).$$

The hessian is

$$\nabla^2 U(\mathbf{x}) = C^{-1} + n^{-1} \text{diag}[\mathbf{x} + \mu].$$

Because marginally $\mathbf{x} \sim \mathcal{N}(0, C)$, we take the expectation

$$\mathbb{E}[\nabla^2 U(\mathbf{x})] = C^{-1} + n^{-1} \text{diag}[\sigma^2/2 + \mu],$$

which is used for preconditioning in Section 5.2.

For the parameters, we use the priors $\sigma^2 \sim \text{Gamma}(2, 0.5)$ and $\beta \sim \text{Gamma}(2, 0.5)$ and the transformations $\sigma^2 = \exp(\varphi_1), \beta = \exp(\varphi_2)$. Then the potential function from the conditional posterior of transformed parameters given (\mathbf{y}, \mathbf{x}) is

$$U(\varphi_1, \varphi_2) = \frac{1}{2} (\exp(\varphi_1) + \exp(\varphi_2)) - 2(\varphi_1 + \varphi_2) + \frac{1}{2} \mathbf{x}^\top C^{-1} \mathbf{x} + \frac{1}{2} \log \det(C),$$

where dependency on (\mathbf{y}, \mathbf{x}) is suppressed in the notation. The gradient is

$$\frac{\partial U(\varphi_1, \varphi_2)}{\partial \varphi_1} = \frac{\exp(\varphi_1)}{2} - 2 + \frac{n}{2} - \frac{1}{2} \mathbf{x}^\top C^{-1} \mathbf{x},$$

$$\frac{\partial U(\varphi_1, \varphi_2)}{\partial \varphi_2} = \frac{\exp(\varphi_2)}{2} - 2 + \frac{1}{2} \text{tr}\left(\frac{\partial C}{\partial \varphi_2}\right) - \frac{1}{2} \mathbf{x}^\top C^{-1} \frac{\partial C}{\partial \varphi_2} C^{-1} \mathbf{x},$$

where

$$\frac{\partial C}{\partial \varphi_2}[(i, j), (i', j')] = m^{-1} \exp(\varphi_1) \exp(-\varphi_2) \sqrt{(i - i')^2 + (j - j')^2} \exp(-\sqrt{(i - i')^2 + (j - j')^2}/(m \exp(\varphi_2))).$$

The marginal expected hessian is

$$\mathbb{E}[\nabla^2 U(\varphi_1, \varphi_2)] = \begin{pmatrix} \frac{1}{2}(\exp(\varphi_1) + n) & \frac{1}{2}\text{tr}(C^{-1} \frac{\partial C}{\partial \varphi_2}) \\ \frac{1}{2}\text{tr}(C^{-1} \frac{\partial C}{\partial \varphi_2}) & \frac{1}{2}(\exp(\varphi_1) + \text{tr}(C^{-1} \frac{\partial C}{\partial \varphi_2} C^{-1} \frac{\partial C}{\partial \varphi_2})) \end{pmatrix}.$$

When sampling the parameters, we use $M = \Sigma^{-1} = \mathbb{E}[\nabla^2 U(\varphi_1, \varphi_2)]$ for preconditioning.

V.3 Step size tuning

As mentioned in Section 5, we periodically adjust step size ϵ based on the acceptance rate during the burn-in period. When acceptance is too low (smaller than a lower threshold), we decrease ϵ by the mapping $\epsilon \leftarrow \max(1 - \sqrt{1 - \epsilon}, \frac{\epsilon}{1 + \delta})$; when acceptance is too high (larger than a upper threshold), we increase ϵ by the mapping $\epsilon \leftarrow \epsilon + \epsilon \cdot \min(1 - \epsilon, \delta)$, where δ is an adjustment value taken to be $\delta = 0.2$ in all our simulations. The increase and decrease mappings are, by design, inverse of each other, as illustrated in Figure S1. The two mappings are mostly linear, but are curved when ϵ is close to 1 to ensure that ϵ is always between 0 and 1 after the update.

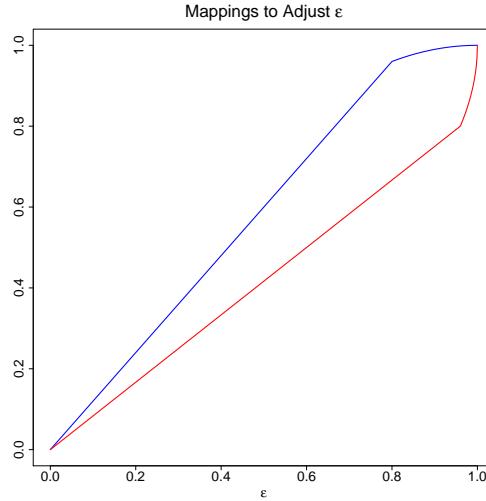


Figure S1: Tuning of step size ϵ with $\delta = 0.2$. Blue curve is mapping used to increase ϵ . Red curve is mapping used to decrease ϵ

VI Additional simulation results

We present an experiment with a multivariate normal distribution, and additional simulation results including pMALA* and GMC from the experiments with the stochastic volatility model and log-Gaussian Cox model.

VI.1 Multivariate normal distribution

Consider the problem of sampling from a 100 dimensional normal distribution with high correlations: $\pi(x) = \mathcal{N}(\mathbf{0}, C)$ where the entries of C are

$$C[i, j] = 0.9^{|i-j|}, \quad i, j = 1, \dots, 100.$$

We do not employ any preconditioning here, although we still refer to pMALA and pMALA* as such. This experiment is used to compare different algorithms when the variance of the target distribution may not be readily approximated. Hence potential advantages associated with the rejection-free property are removed from HAMS-A/B.

In terms of tuning, we set $\epsilon = 0.19$ for HAMS-A, HAMS-B, UDL, GMC, pMALA and pMALA* to maintain acceptance rates around 70%. Through empirical trials we find that HAMS-A, UDL and GMC have good performance using a large carryover (c value), while HAMS-B favors a relatively small carryover. Hence we set $c = 0.95$ for HAMS-A, UDL and GMC, $c = 0.25$ for HAMS-B. For HMC, we set $nleap = 50$ and $\epsilon = 0.17$ which also yields a 70% acceptance rate. For RWM, we set $\epsilon = 0.06$ and the resulting acceptance is around 40%. To account for the additional computation cost due to leapfrog steps, HMC is run for 200 iterations and all other methods are run for $200 \times 50 = 10000$ iterations. The simulation process is repeated for 100 times with a fixed starting value of $\mathbf{0}$.

Figure S2 shows boxplots of sample means and variances of 100 coordinates and sample covariances of 100 coordinates with the first coordinate after centered about the true values. Hence deviations from 0 (marked by red lines) show divergence from the truth. From the boxplots, we see that HAMS-A, UDL and GMC are comparable to each other. They are mostly accurate in the means and covariances while slightly underestimate the variances. Sample means of HAMS-B are correctly centered but exhibit more variation.

HAMS-B underestimates the variances more than HAMS-A, UDL, and GMC, and also the covariances associated with the first several coordinates. Compared to HAMS-B, pMALA shows similar underestimation of variances and covariances, but has an even wider spread in sample means. For pMALA*, because $\epsilon = 0.18$ is small, its performance is similar to that of the unmodified pMALA. While HMC is good in terms of sample means, it underestimates variances and is inaccurate in covariances with a considerable number of outliers. RWM performs poorly to capture neither variance nor covariance.

Figure S3 shows trace plots of first 2000 iterations (first 40 iterations for HMC) from an individual run. The first two coordinates are plotted and red ellipses mark regions containing 95% probability of the marginal target density. HAMS-A best fills up the area. UDL and GMC are also reasonable but leave a small part in the upper right blank. HAMS-B, pMALA and pMALA* all cover smaller areas with parts of the corners missing. The HMC trace misses the top right quadrant and its movement is only aligned to the long axis of the ellipse. RWM performs poorly and covers the least amount of the area.

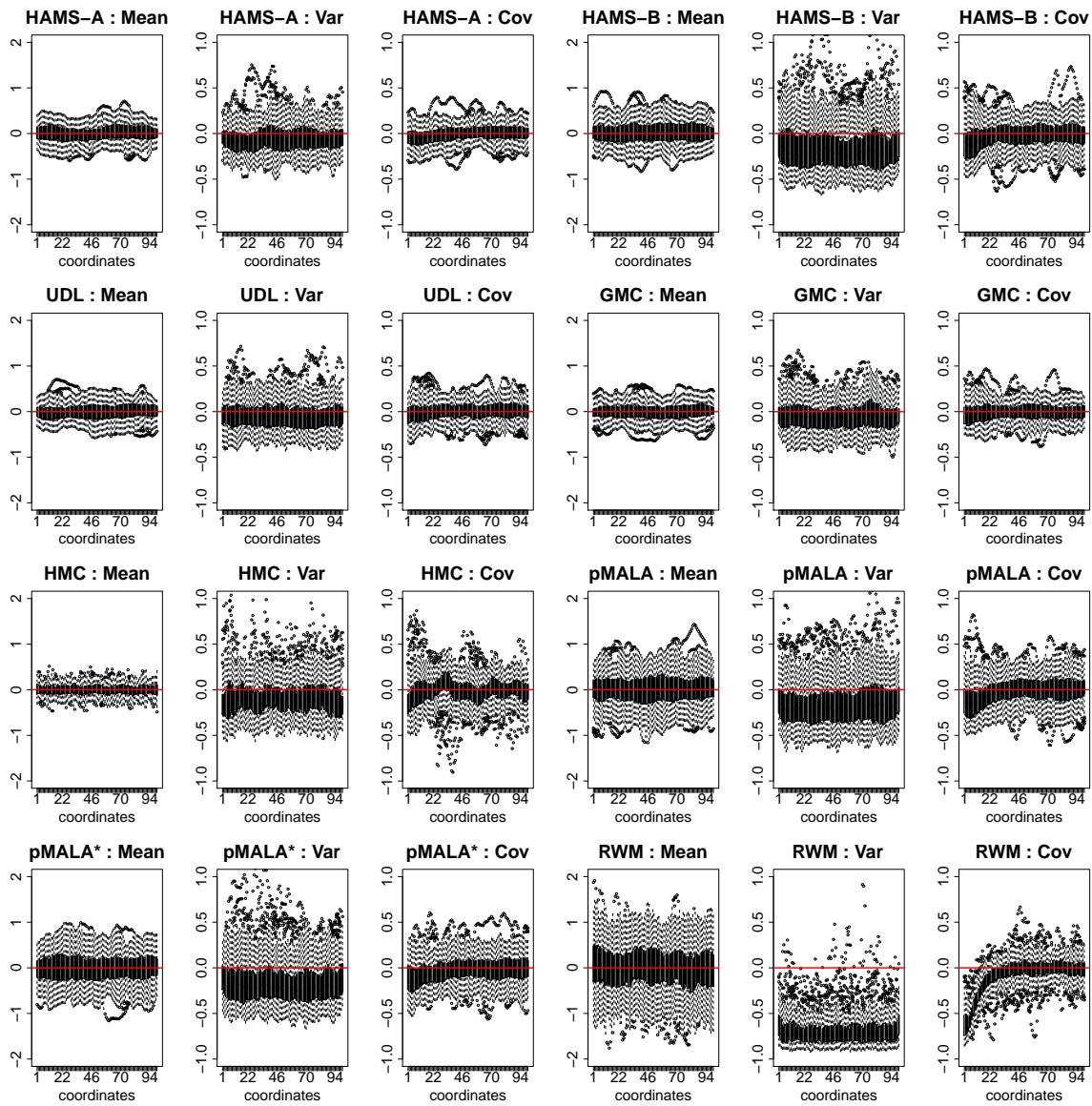


Figure S2: Time-adjusted and centered boxplots of sample means, variances, and covariances of 100 coordinates over 100 repetitions for sampling from the multivariate normal distribution. Red lines indicate zero.

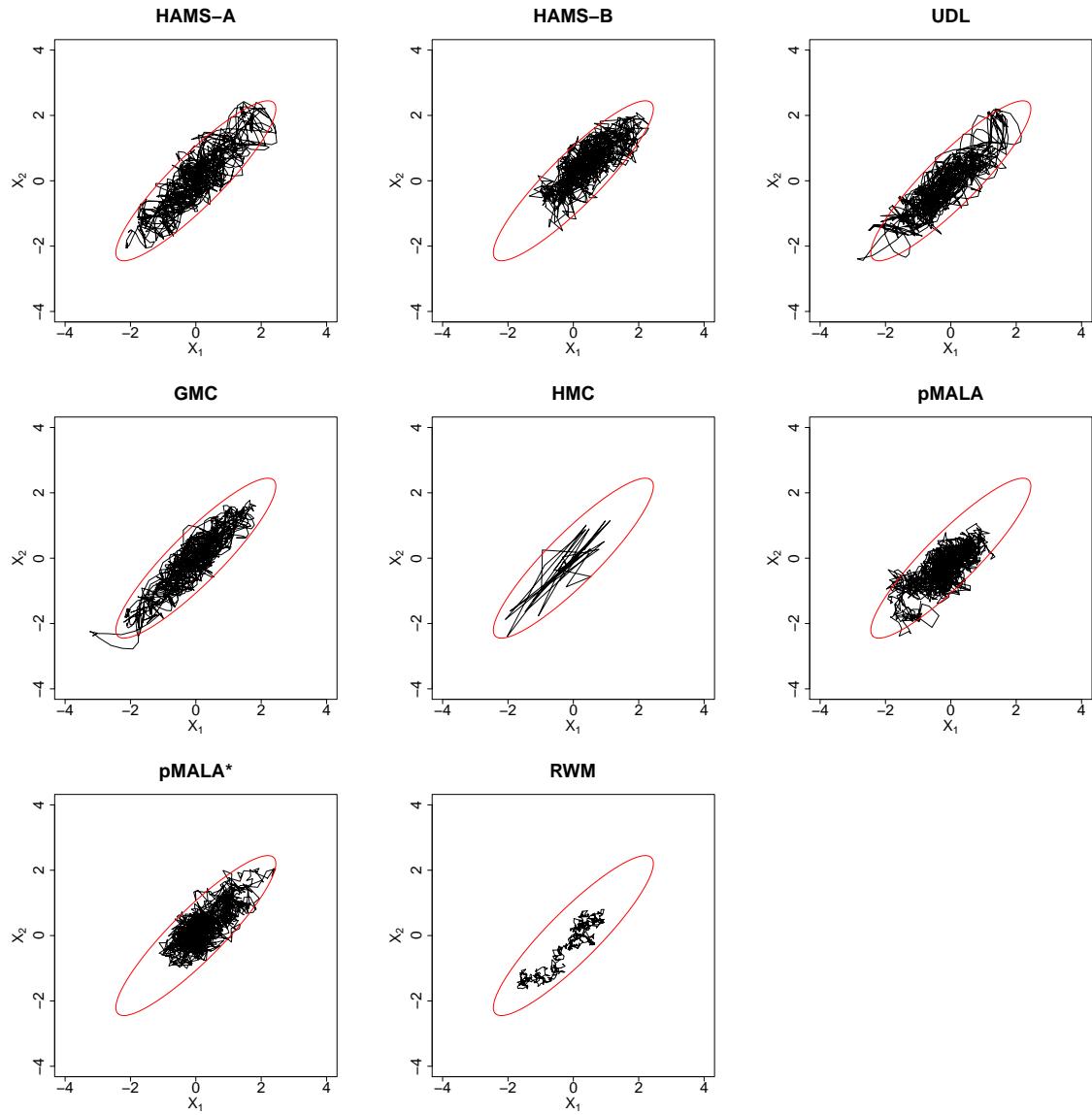


Figure S3: Time-adjusted trace plots of the first two coordinates from first 2000 iterations (first 40 iterations for HMC) for sampling from the multivariate normal distribution. Red ellipses indicate 95% probability regions.

VI.2 Stochastic volatility model

Consider the setting in Section 5.1. For sampling latent variables only, Figure S4 shows the average acceptance rates (red curves) and step sizes ϵ (black curves) during the burn-in period, using the tuning procedure described in Section V.3. The upper and lower thresholds of acceptance rates for such adjustments are marked by the dashed lines. From Figure S4, our tuning procedure seems effective in obtaining desirable acceptance rates for each algorithm. Furthermore, larger step sizes are achieved for HAMS-A, HAMS-B, and pMALA* than other methods, while similar acceptance rates are obtained. A possible explanation is that these three methods use coefficient $\frac{\epsilon^2}{1+\sqrt{1-\epsilon^2}}$ instead of $\frac{\epsilon^2}{2}$ for gradient updates and satisfy the rejection-free property (i.e., proposals are always accepted) for a normal target density with pre-specified variance. Hence relatively large step sizes are allowed for these methods together with reasonable acceptance rates, when the target density is not far from such a normal density. The differences in step sizes associated with the rejection-free property can be seen to underlie advantages of HAMS-A/B as well as improvement of pMALA* over pMALA in our results.

From Table S1 (expanded from Table 1), GMC has similar performance to UDL, while pMALA* improves upon pMALA considerably. The time-adjusted centered boxplots of sample means in Figure S5 (expanded from Figire 2) also confirm that pMALA* performs better than the original pMALA. Figure S6 shows time-adjusted averages (over repeated runs) of sample means for all latent variables. The curves are shifted (centered relative to the dashed lines) so that the overall shapes can be compared between methods. All methods yield similar average sample means including RWM. Figure S7 shows time-adjusted variances in the log scale (over repeated runs) of sample means. It is clear that HAMS-A and HAMS-B have the smallest variances, and hence are more consistent across repeated runs than other methods. pMALA* has slightly larger variance, followed by GMC, UDL, pMALA, HMC and RWM. Additional trace plots and ACFs are shown in Figures S8 – S10, for different latent variables than in Figure 1.

Results of posterior sampling are presented in Table S2 (expanded from Table 2). While pMALA* and GMC have reasonable sample means, they also have more variability than

our methods. pMALA* has large standard deviation in β while GMC has large standard deviation in both β and ϕ . Such behaviors are also observed in Figure S12.

Finally, trace plots of each parameter from an individual run are shown in Figure S12. These trace plots are divided into four stages by blue vertical lines. In the first stage, we apply no preconditioning and adjust step size ϵ . In the second stage we fix ϵ and collect samples for crude parameter estimates; we then evaluate preconditioning matrices using the sample means of parameters from the second stage and fix them. In the third stage we apply preconditioning and adjust ϵ . In the fourth stage, we fix ϵ and continue applying preconditioning to collect working samples.

Method	Time (s)	ESS (min, median, max)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	98.7	(2420, 3660, 6668)	24.51
HAMS-B	99.6	(1915, 3404, 6229)	19.23
UDL	98.4	(657, 1020, 1661)	6.68
GMC	85.0	(752, 1249, 1914)	8.85
HMC	1250.1	(1125, 3698, 11240)	0.90
pMALA	120.5	(374, 610, 990)	3.11
pMALA*	122.6	(1740, 2879, 5429)	14.19
RWM	51.7	(7, 12, 20)	0.14

Table S1: Runtime and ESS comparison (including GMC and pMALA*) for sampling latent variables in the stochastic volatility model. Results are averaged over 50 repetitions.

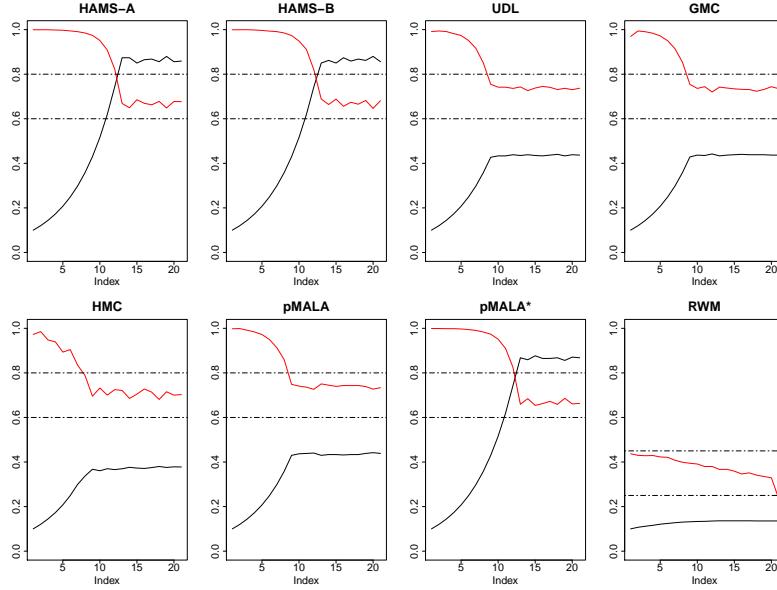


Figure S4: Average step sizes (black) and acceptance rates (red) for sampling latent variables in the stochastic volatility model. For every 250 iterations, acceptance rates are calculated and step sizes adjusted. Results are averaged over 50 repetitions.

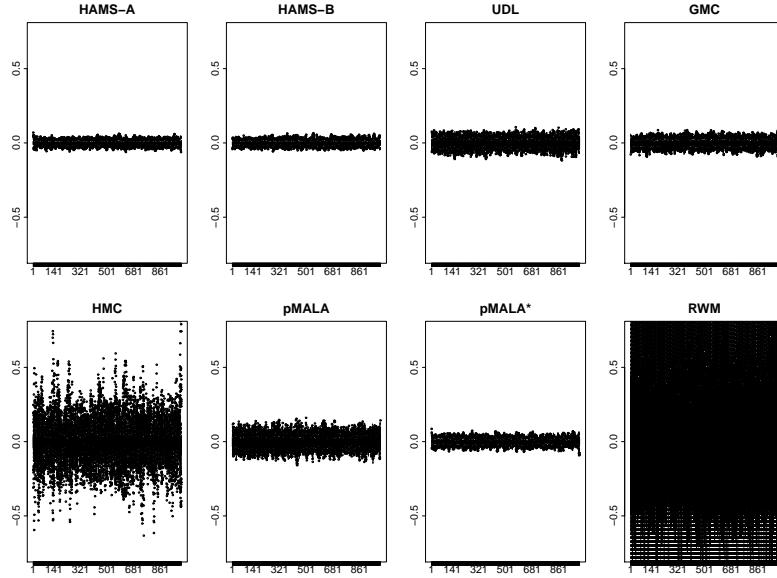


Figure S5: Time-adjusted and centered boxplots of sample means of all latent variables over 50 repetitions for sampling latent variables in the stochastic volatility model.

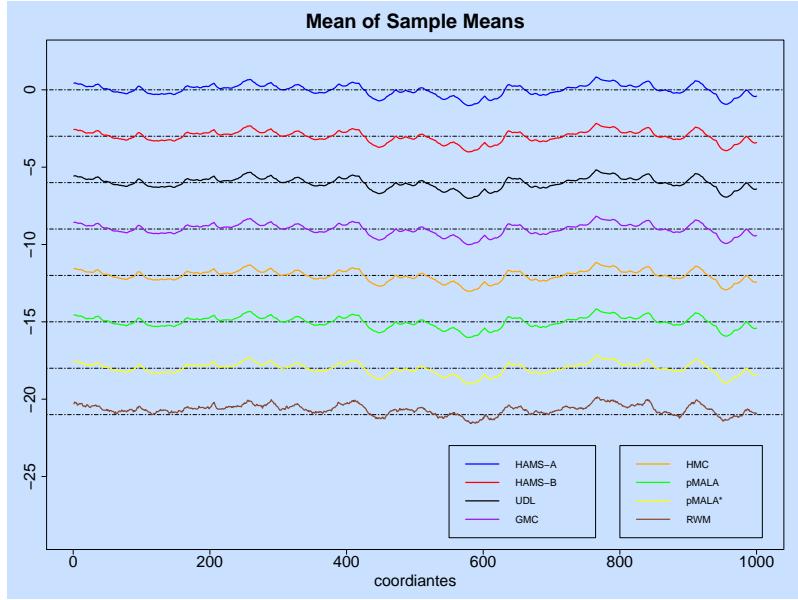


Figure S6: Time-adjusted averages of sample means (shifted) of all latent variables over 50 repetitions for sampling latent variables in the stochastic volatility model.

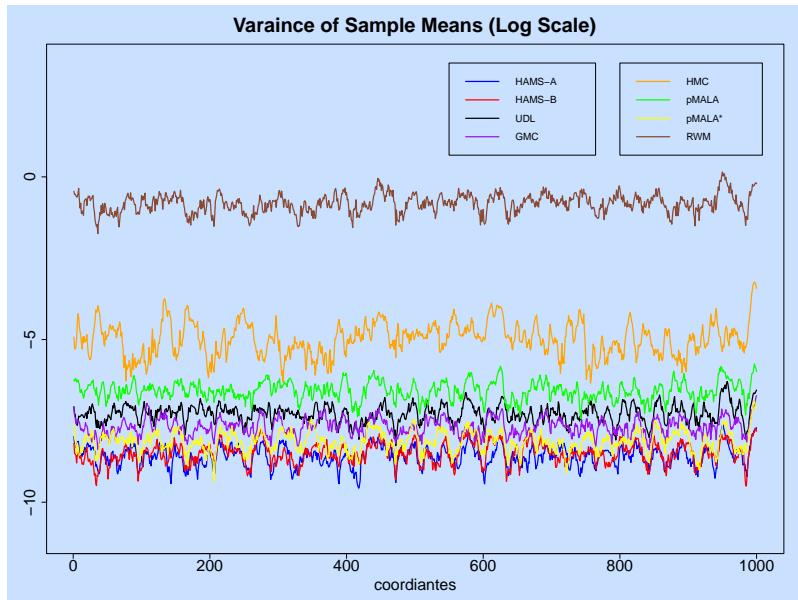


Figure S7: Time-adjusted variances of sample means (log-scale) of all latent variables over 50 repetitions for sampling latent variables in the stochastic volatility model.

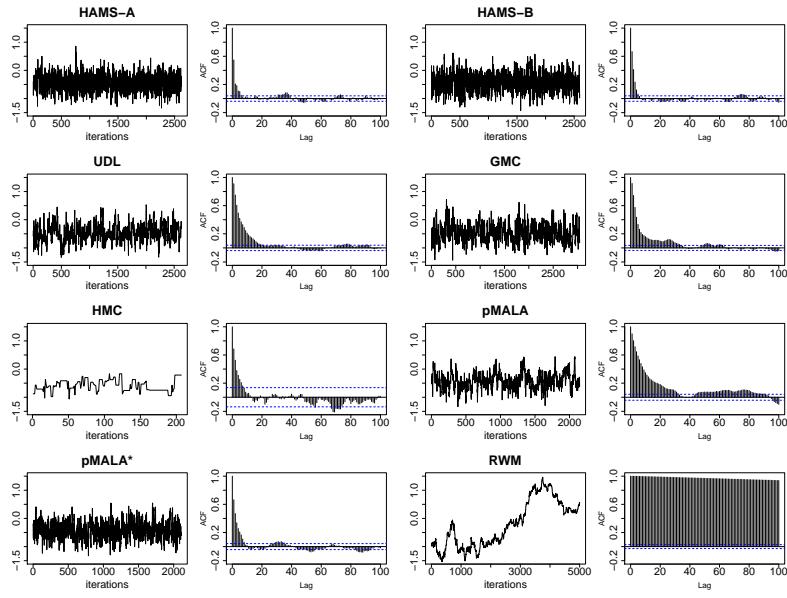


Figure S8: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the stochastic volatility model.

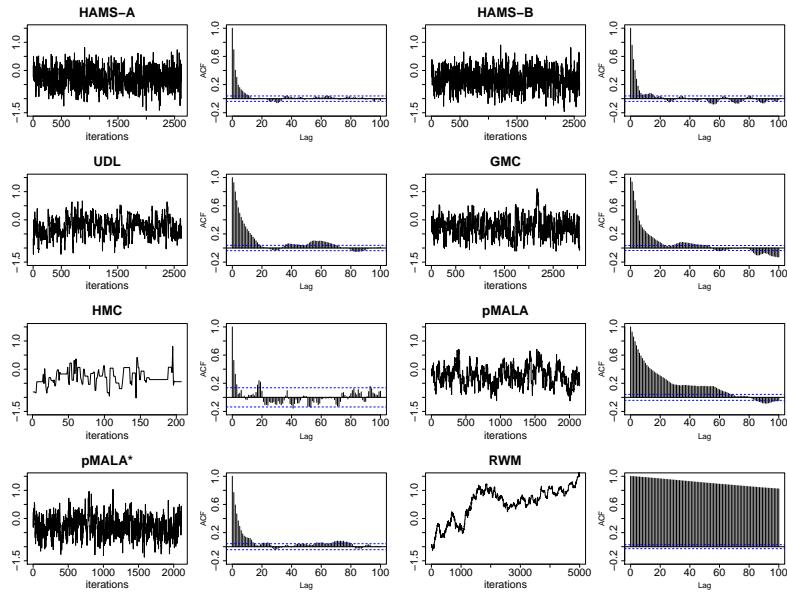


Figure S9: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the stochastic volatility model.

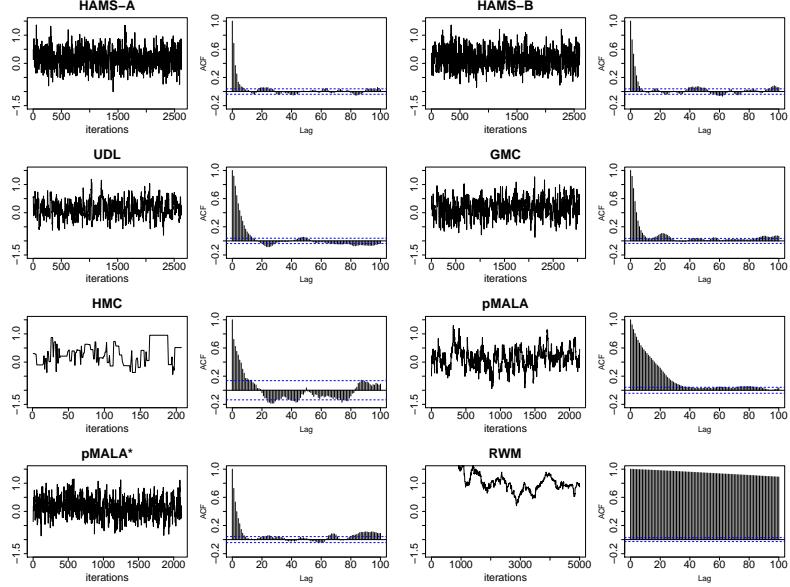


Figure S10: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the stochastic volatility model.

Method	Time (s)	β (sd)	Sample Mean σ (sd)	ϕ (sd)	ESS (β, σ, ϕ)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	1951.3	0.68 (0.034)	0.19 (0.006)	0.98 (0.001)	(30, 73, 220)	0.015
HAMS-B	1942.3	0.68 (0.037)	0.19 (0.007)	0.98 (0.001)	(25, 59, 188)	0.013
UDL	1945.8	0.68 (0.039)	0.20 (0.008)	0.98 (0.002)	(29, 37, 87)	0.015
GMC	1968.2	0.67 (0.059)	0.20 (0.007)	0.98 (0.003)	(35, 58, 169)	0.018
HMC	20920.2	0.69 (0.050)	0.19 (0.014)	0.98 (0.003)	(19, 12, 78)	0.001
pMALA	2013.0	0.68 (0.040)	0.20 (0.005)	0.98 (0.001)	(15, 30, 76)	0.008
pMALA*	2015.2	0.70 (0.054)	0.19 (0.006)	0.98 (0.001)	(23, 53, 149)	0.012
RWM	1311.1	0.76 (0.050)	0.47 (0.229)	0.51 (0.149)	(89, 12, 7)	0.006

Table S2: Comparison of posterior sampling (including GMC and pMALA*) in the stochastic volatility model. Standard deviations of sample means are in parentheses. Results are averaged over 20 repetitions.

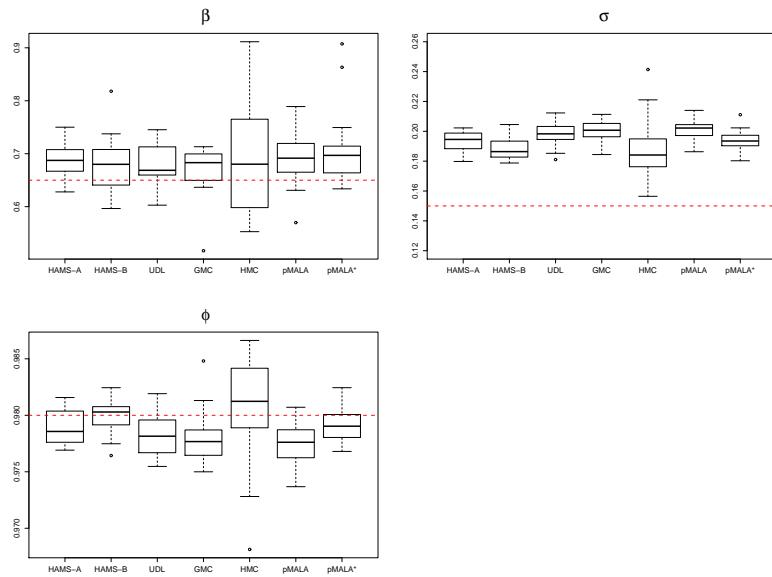


Figure S11: Time-adjusted boxplots of sample means of parameters over 20 repetitions for posterior sampling in the stochastic volatility model. The data generating parameter values are marked by red lines.

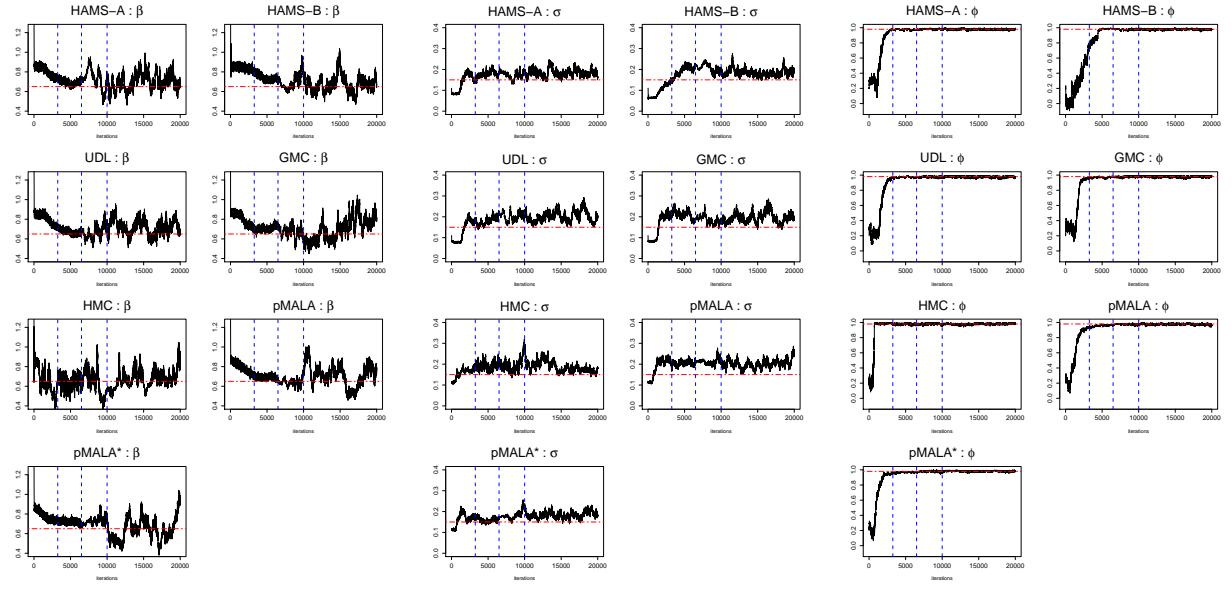
(a) Trace plots of β (b) Trace plots of σ (c) Trace plots of ϕ

Figure S12: Trace plots from an individual run for posterior sampling in the stochastic volatility model. Data generating parameter values are marked by red horizontal lines. There are four stages divided by blue vertical lines. The first two are without preconditioning, with 3250 iterations each. The last two are with preconditioning, with 3500 and 10000 iterations respectively. The first three stages are counted as burn-in.

VI.3 Log-Gaussian Cox model

We report additional simulation results for the log-Gaussian Cox model discussed in Section 5.2. The overall conclusions remain similar as in the stochastic volatility model. When only sampling latent variables, pMALA* improves upon the original pMALA, and GMC shows comparable performance to UDL. While all methods have similar average sample means, HAMS-A and HAMS-B have the smallest variance. For posterior sampling results, pMALA* inflates the standard deviations of sample means, but brings the estimates more aligned with HAMS. Compared to the stochastic volatility model, the effect of preconditioning can be seen more clearly from the trace plots in Figure S21.

Method	Time (s)	ESS (min, median, max)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	81.0	(803, 1655, 5461)	9.91
HAMS-B	78.8	(619, 1376, 4831)	7.86
UDL	78.8	(322, 622, 1761)	4.08
GMC	81.9	(359, 742, 2081)	4.38
HMC	1285.9	(935, 1621, 4523)	0.73
pMALA	116.4	(184, 340, 1002)	1.58
pMALA*	115.9	(600, 1197, 4275)	5.17
RWM	51.1	(8, 13, 22)	0.16

Table S3: Runtime and ESS comparison (including GMC and pMALA*) for sampling latent variables in the log-Gaussian Cox model ($n = 1024$). Results are averaged over 50 repetitions.

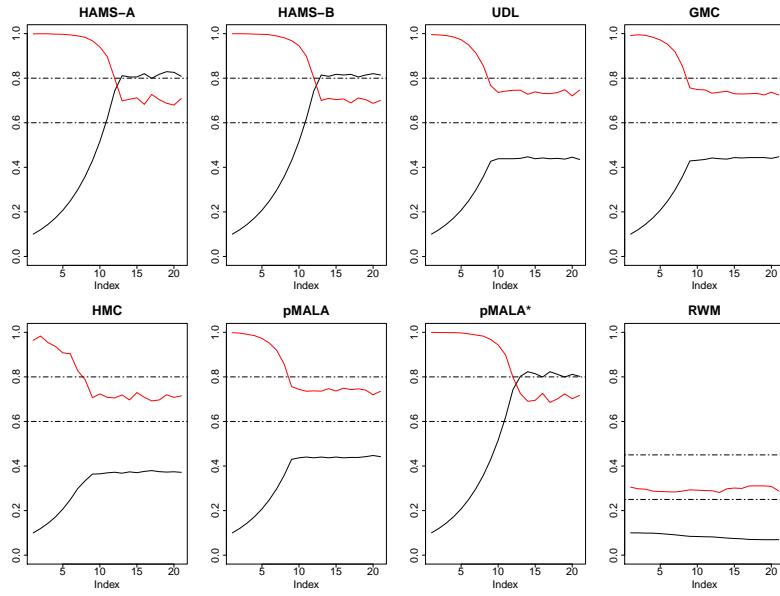


Figure S13: Average step sizes (black) and acceptance rates (red) for sampling latent variables in the log-Gaussian Cox model ($n = 1024$). For every 250 iterations, acceptance rates are calculated and step sizes adjusted. Results are averaged over 50 repetitions.

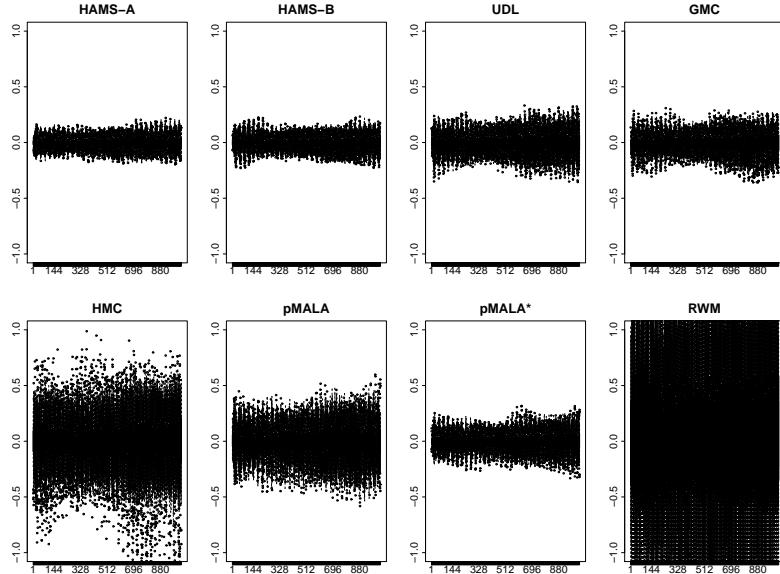


Figure S14: Time-adjusted and centered boxplots of sample means of all latent variables over 50 repetitions for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

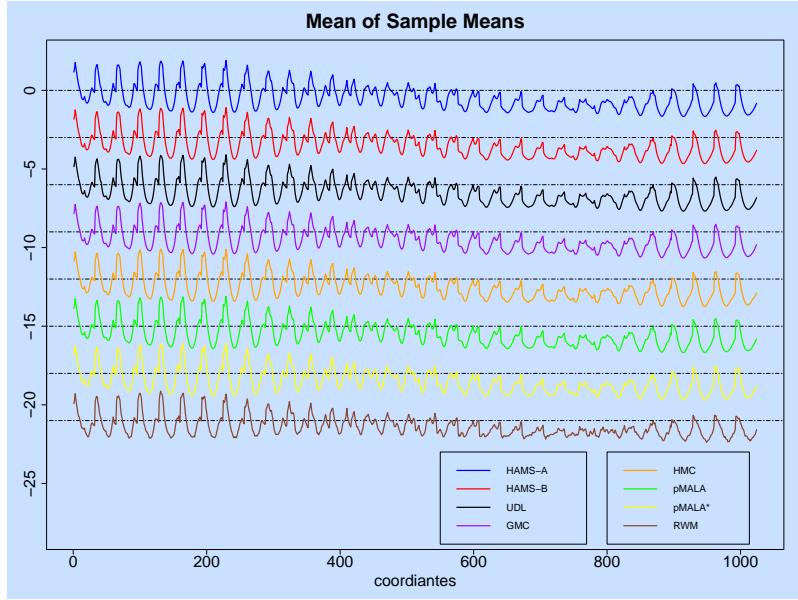


Figure S15: Time-adjusted averages of sample means (shifted) of all latent variables over 50 repetitions for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

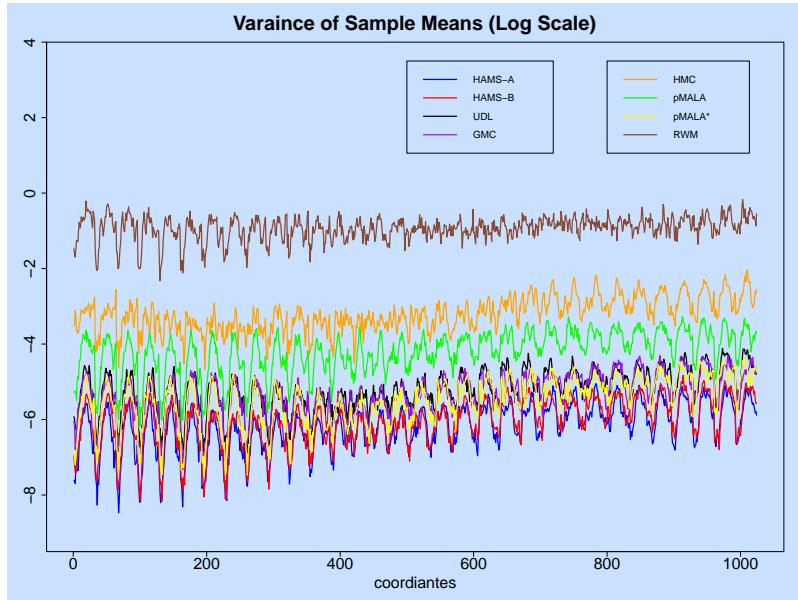


Figure S16: Time-adjusted variances of sample means (log-scale) of all latent variables over 50 repetitions for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

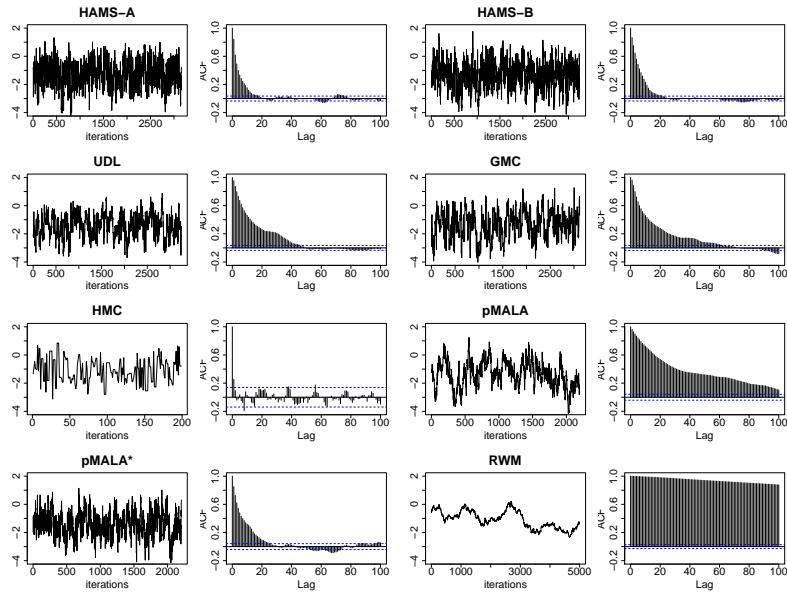


Figure S17: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

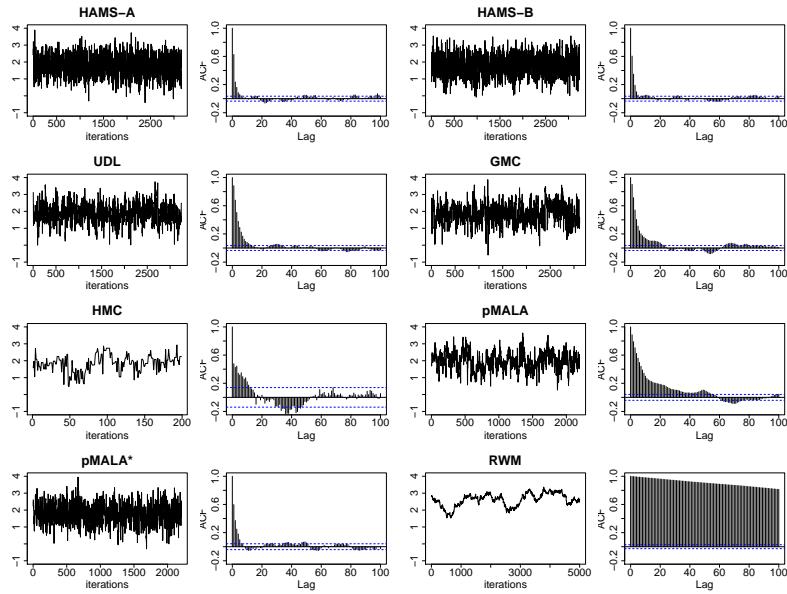


Figure S18: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

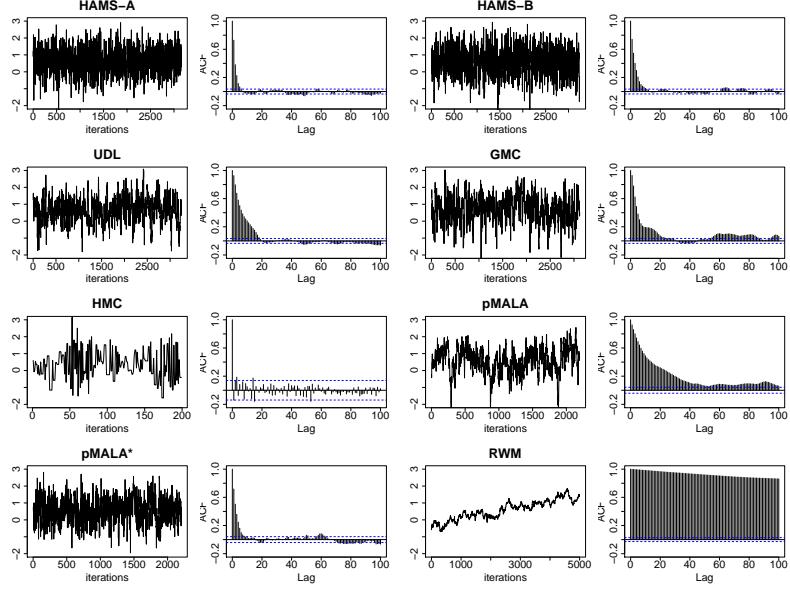


Figure S19: Time-adjusted trace and ACF plots of one latent variable from an individual run for sampling latent variables in the log-Gaussian Cox model ($n = 1024$).

Method	Time (s)	Sample Mean σ^2 (sd)	β (sd)	ESS (σ^2, β)	$\frac{\text{minESS}}{\text{Time}}$
HAMS-A	2766.8	3.90 (0.155)	0.68 (0.073)	(978, 207)	0.075
HAMS-B	2762.8	3.93 (0.190)	0.69 (0.106)	(838, 263)	0.095
UDL	2759.1	3.79 (0.171)	0.59 (0.105)	(755, 246)	0.089
GMC	2763.8	3.81 (0.156)	0.61 (0.132)	(884, 142)	0.051
HMC	25386.0	3.88 (0.084)	0.75 (0.113)	(2253, 139)	0.005
pMALA	2755.3	3.76 (0.189)	0.57 (0.101)	(528, 178)	0.065
pMALA*	2758.4	3.89 (0.223)	0.69 (0.138)	(623, 182)	0.066
RWM	1752.2	3.70 (0.662)	1.26 (1.434)	(226, 87)	0.050

Table S4: Comparison of posterior sampling (including GMC and pMALA*) in the log-Gaussian Cox model ($n = 256$). Standard deviations of sample means are in parentheses. Results are averaged over 20 repetitions.

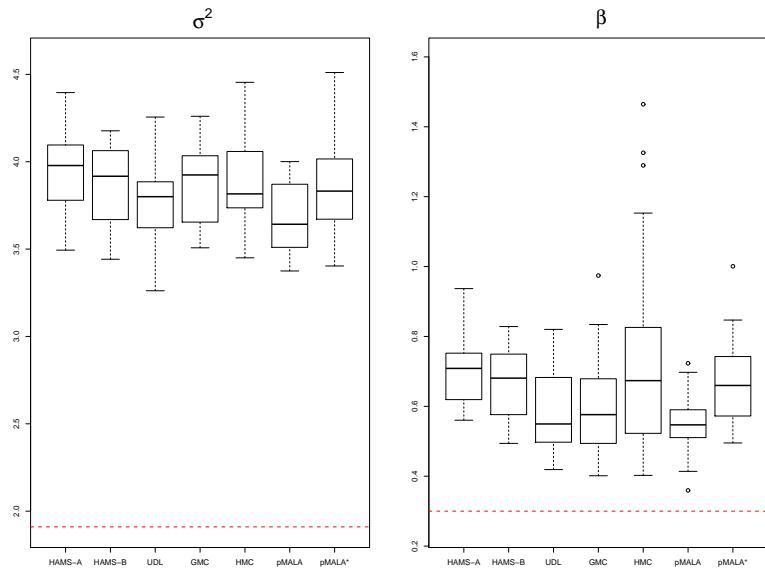
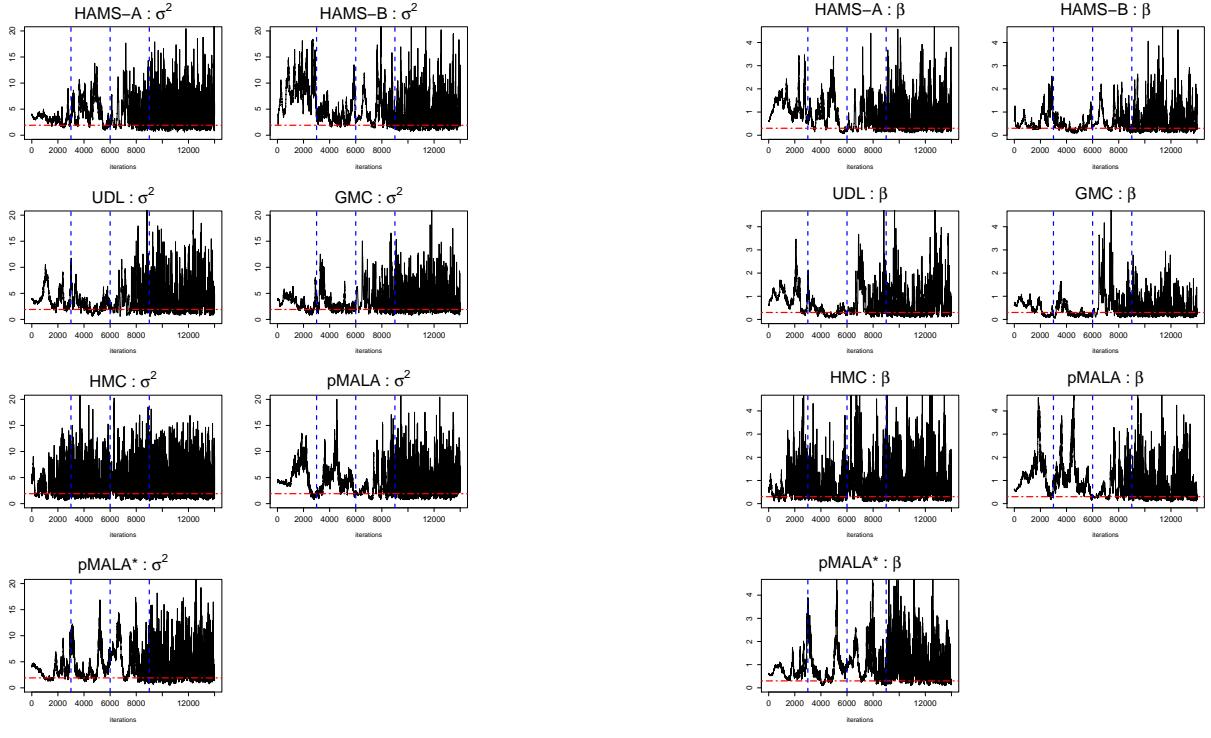


Figure S20: Time-adjusted boxplots of sample means of parameters over 20 repetitions for posterior sampling in the log-Gaussian Cox model ($n = 256$). The data generating parameter values are marked by red lines. Due to skewness of posterior densities (Figure 6), the posterior modes are reasonably close to the true parameter values, while the posterior means are not, especially for σ^2 .



(a) Trace plots of σ^2

(b) Trace plots of β

Figure S21: Trace plots from an individual run for posterior sampling in the log-Gaussian Cox model ($n = 256$). Data generating parameter values are marked by red horizontal lines. There are four stages divided by blue vertical lines. The first two are without preconditioning, with 3000 iterations each. The last two are with preconditioning, with 3000 and 5000 iterations respectively. The first three stages are counted as burn-in.