Importance Sampling-based Transport Map Hamiltonian Monte Carlo for Bayesian Hierarchical Models

Kjartan Kloster Osmundsen\textsuperscript{*1}, Tore Selland Kleppe\textsuperscript{1}, and Roman Liesenfeld\textsuperscript{2}

\textsuperscript{1}Department of Mathematics and Physics, University of Stavanger, Norway
\textsuperscript{2}Institute of Econometrics and Statistics, University of Cologne, Germany

December 11, 2019

Abstract

We propose an importance sampling (IS)-based transport map Hamiltonian Monte Carlo procedure for performing full Bayesian analysis in general nonlinear high-dimensional hierarchical models. Using IS techniques to construct a transport map, the proposed method transforms the typically highly challenging target distribution of a hierarchical model into a target which is easily sampled using standard Hamiltonian Monte Carlo. Conventional applications of high-dimensional IS, where infinite variance of IS weights can be a serious problem, require computationally costly high-fidelity IS distributions. An appealing property of our method is that the IS distributions employed can be of rather low fidelity, making it computationally cheap. We illustrate our algorithm in applications to challenging dynamic state-space models, where it exhibits very high simulation efficiency compared to relevant benchmarks, even for variants of the proposed method implemented using a few dozen lines of code in the Stan statistical software.

Keywords: Hamiltonian Monte Carlo; Importance Sampling; Transport Map; Bayesian hierarchical models; State-space models; Stan

1 Introduction

Computational methods for Bayesian nonlinear/non-Gaussian hierarchical models is an active field of research, and advances in such computational methods allow researchers to build and fit progressively more complex models. Existing Markov chain Monte Carlo (MCMC) methods for such models fall broadly into four categories. Firstly, Gibbs sampling is widely used, in part due to its simple implementation (see e.g. Robert and Casella [2004]). However, a naive implementation updating latent variables in one block and

\textsuperscript{*}Corresponding author. Email: kjartan.osmundsen@gmail.com
model parameters in another block can suffer from a very slow exploration (see e.g. Jacquier et al., 1994) of the target distribution if this joint distribution implies a strong, typically nonlinear dependence structure of the variables in the two blocks. Secondly, methods that update latent variables and parameters jointly avoid the nonlinear dependence problem of Gibbs sampling. One such approach for joint updates is to use Riemann manifold Hamiltonian Monte Carlo (RMHMC) methods (see e.g. Girolami and Calderhead, 2011; Zhang and Sutton, 2014; Kleppe, 2018). However, they critically require update proposals which are properly aligned with the (typically rather variable) local geometry of the target, the generation of which can be computationally demanding for complex high-dimensional joint posteriors of the parameters and latent variables.

The third category is pseudo-marginal methods (see e.g. Andrieu et al., 2010; Pitt et al., 2012 and references therein), which bypasses the problematic parameters and latent variables dependency by targeting directly the marginal posterior of the parameters. Pseudo-marginal methods require, however, a low variance, unbiased Monte Carlo (MC) estimate of said posterior, which can often be extremely computationally demanding for high-dimensional models (see e.g. Flury and Shephard, 2011). Moreover, for models with many parameters, it can be difficult to select an efficient proposal distribution for updating the parameters if the MC estimates for the marginal posterior are noisy and/or contain many discontinuities, which is typically the case if the MC estimator is implemented using particle filtering techniques.

Finally, the fourth category is transport map/dynamic rescaling methods (see e.g. Parno and Marzouk, 2018; Hoffman et al., 2019), which rely on introducing a modified parameterization related to the original parameterization via the nonlinear transport map. The transport map is chosen so that the target distribution in the modified parameterization is more well behaved and allows MCMC sampling using standard techniques. The Dynamically rescaled Hamiltonian Monte Carlo (DRHMC) approach of Kleppe (2019) involves a recipe for constructing transport maps suitable for a large class of Bayesian hierarchical models, and where the models are fitted using the (fixed scale) No-U-Turn Sampler (NUTS) Hamiltonian Monte Carlo (HMC) algorithm (Hoffman and Gelman, 2014) implemented in Stan (Stan Development Team, 2019b).

The present paper also considers a transport map approach for Bayesian hierarchical models, and sample from the modified target using HMC methods. However, the strategy for constructing the transport map considered here is different from that of DRHMC. Specifically, DRHMC involves deriving the transport maps from the model specification itself, and in particular it requires the availability of closed-form expressions for certain precision- and Fisher information matrices associated with the model. Moreover, the DRHMC approach is in practice limited to models containing only a certain class of nonlinearities which lead to so-called constant information parameterizations.

Here, on the other hand, we consider transport maps derived from well-known importance sampling
(IS) methods for the latent variables only. This approach relies only on the ability to evaluate the log-target density (and potentially it’s derivatives) pointwise, and therefore bypasses the substantial analytic tractability requirement of DRHMC. The proposed approach is consequently more automatic in nature, and in particular applicable to a wider range of nonlinear models than DRHMC. Still, some analytical insight into the model is beneficial in terms of computational speed when choosing the initial iterates of the involved iterative processes.

A fortunate property of the proposed methodology, relative to conventional applications of high-dimensional importance sampling (see e.g. Koopman et al. 2009), is that the importance densities applied within the present framework may be of relatively low fidelity as long as they reflect the location and scale of the distribution of the latent state conditioned both on data and parameters. Since parameters and latent variables are updated simultaneously, the slow exploration of the target associated with Gibbs sampling is avoided. Moreover, being transport map-based, rather than say RMHMC-based, the proposed methodology allows for the application of standard HMC and in particular can be implemented with minimal effort in Stan.

The application of IS methods to construct transport maps also allows the proposed methodology to be interpreted as a pseudo-marginal method, namely a special case (with simulation sample size $n = 1$) of the pseudo-marginal HMC method of Lindsten and Doucet (2016). However, our focus on models with high-dimensional latent variables generally precludes the application of ‘brute force’ IS estimators that do not reflect information from the data (see, e.g., Danielsson, 1994). This is the case even for increased simulation sample size of the IS estimate, as is possible in the general setup of Lindsten and Doucet (2016).

The rest of the paper is laid out as follows: Section 2 provides some background and Section 3 introduces IS-based transport maps. Section 4 discusses specific choices of IS-based transport maps and Section 5 provides a simulation experiment where the fidelity vs computational cost tradeoff of the different transport maps is explored numerically. Finally, Section 6 presents a realistic application and Section 7 provides some discussion. The paper is accompanied by supplementary material giving further details in several regards, and the code used for the computations is available at https://github.com/kjartako/TMHMC.

2 Background

This section outlines some background on HMC and why the application of HMC in default formulations of hierarchical models is problematic. In what follows, we use $\mathcal{N}(\mathbf{x}|\mu, \Sigma)$ to denote the probability density function of a $\mathcal{N}(\mu, \Sigma)$ random vector evaluated at $\mathbf{x}$, while $\nabla_\mathbf{z}$ and $\nabla_\mathbf{z}^2$ are used, respectively, for the gradient/Jacobian and Hessian operator with respect to the vector $\mathbf{z}$. 
2.1 HMC

Over the past decade, HMC introduced by Duane et al. (1987) has been extensively used as a general-purpose MCMC method, often applied for simulating from posterior distributions arising in Bayesian models (Neal 2011). HMC offers the advantage of producing close to perfectly mixing MCMC chains by using the dynamics of a synthetic Hamiltonian system as proposal mechanism. The popular Bayesian modelling software Stan (Stan Development Team 2019b) is an easy to use HMC implementation based on the NUTS HMC algorithm of Hoffman and Gelman (2014).

Suppose one seeks to sample from an analytically intractable target distribution with density kernel \( \tilde{\pi}(q) \), \(q \in \Omega \subseteq \mathbb{R}^s\). To this end, HMC takes the variable of interest \( q \) as the ‘position coordinate’ of a Hamiltonian system, which is complemented by an (artificial) ‘momentum variable’ \( p \in \mathbb{R}^s \). The corresponding Hamiltonian function specifying the total energy of the dynamical system is given by

\[
H(q, p) = -\log \tilde{\pi}(q) + \frac{1}{2} p^\prime M^{-1} p, \tag{1}
\]

where \( M \in \mathbb{R}^{s \times s} \) is a symmetric, positive definite ‘mass matrix’ representing an HMC tuning parameter. For near-Gaussian target distributions, for instance, setting \( M \) close to the precision matrix of the target ensures the best performance. The law of motions under the dynamic system specified by the Hamiltonian \( H \) is determined by Hamilton’s equations given by

\[
\frac{d}{dt} p(t) = -\nabla_q H(q(t), p(t)) = \nabla_q \log \tilde{\pi}(q), \quad \frac{d}{dt} q(t) = \nabla_p H(q(t), p(t)) = M^{-1} p. \tag{2}
\]

It can be shown that the dynamics associated with Hamilton’s equations preserves both the Hamiltonian (i.e. \( dH(q(t), p(t)) / dt = 0 \)) and the Boltzmann distribution \( \pi(q, p) \propto \exp\{-H(q, p)\} \propto \tilde{\pi}(q) \mathcal{N}(p|0_s, M) \), in the sense that if \( [q(t), p(t)] \sim \pi(q, p) \), then \( [q(t+\tau), p(t+\tau)] \sim \pi(q, p) \) for any (scalar) time increment \( \tau \). Based on the latter property, a valid MCMC scheme for generating \( \{q^{(k)}\}_k \sim \pi(q) \) would be to alternate between the following two steps: (i) Sample a new momentum \( p^{(k)} \sim \mathcal{N}(0_s, M) \) from the \( p \)-marginal of the Boltzmann distribution; and (ii) use the Hamiltonian’s equations (2) to propagate \( [q(0), p(0)] = [q^{(k)}, p^{(k)}] \) for some increment \( \tau \) to obtain \( [q(\tau), p(\tau)] = [q^{(k+1)}, p^\ast] \) and discard \( p^\ast \). However, for all but very simple scenarios (like those with a Gaussian target \( \tilde{\pi}(q) \)) the transition dynamics according to (2) does not admit closed-form solution, in which case it is necessary to rely on numerical integrators for an approximative solution. Provided that the numerical integrator used for that purpose is symplectic, the numerical approximation error can be exactly corrected by introducing an accept-reject (AR) step, which uses the Hamiltonian to compare the total energy of the new proposal for the pair \( (q, p) \) with that of the old pair inherited from the
previous MCMC step (see, e.g., [Neal, 2011]). More specifically each iteration of the HMC algorithm involves
the following steps

- Refresh the momentum \( p^{(k)} \sim N(0_s, M) \).
- Propagate approximately the dynamics (2) from \((q(0), p(0)) = (q^{(k)}, p^{(k)})\) to obtain \((q^*, p^*) \approx (q(L\varepsilon), p(L\varepsilon))\) using \(L\) symplectic integrator steps with time-step size \(\varepsilon\).
- Set \(q^{(k+1)} = q^*\) with probability \(\min(1, \exp(H(q^{(k)}, p^{(k)}) - H(q^*, p^*))\) and \(q^{(k+1)} = q^{(k)}\) with remaining probability.

The most commonly used symplectic integrator is the Störmer-Verlet or leapfrog integrator (see, e.g., [Leimkuhler and Reich, 2004; Neal, 2011]). When implementing numerical integrators with AR-corrections it is critical that the selection of the step size accounts for the inherent trade-off between the computing time required for generating AR proposals and their quality reflected by their corresponding acceptance rates. \((q, p)\)-proposals generated by using small (big) step sizes tend to be computationally expensive (cheap) but imply a high (low) level of energy preservation and thus high (low) acceptance rates. Finally, the energy preservation properties of the symplectic integrator for any given step size critically relies on the nature of the target distribution. It is taken as a rule of thumb for the remainder of the text that high-dimensional, highly non-Gaussian targets typically require small step sizes and many steps, whereas high-dimensional near-Gaussian targets can be sampled efficiently with rather large step sizes and few steps.

2.2 Hierarchical models and HMC

Consider a stochastic model for a collection of observed data \(y\) involving a collection of latent variables \(x\) and a vector of parameters \(\theta \in \mathbb{R}^d\) with prior density \(p(\theta)\). The conditional likelihood for observations \(y\) given a value of the latent variable \(x \in \mathbb{R}^D\) is denoted by \(p(y|x, \theta)\) and the prior for \(x\) by \(p(x|\theta)\). This latent variable model is assumed to be nonlinear and/or non-Gaussian so that both the joint posterior for \((x, \theta)\) as well as the marginal posterior for \(\theta\) are analytically intractable.

The joint posterior for \((x, \theta)\) under such a latent variable model, given by \(p(x, \theta|y) \propto p(y|x, \theta)p(x|\theta)p(\theta)\), can have a complex dependence structure. In particular, when the scale of \(x|\theta, y\) varies substantially as a function of \(\theta\) in the typical range of \(p(\theta|y)\), the joint posterior will be “funnel-shaped” (see [Kleppe, 2019] Figure 1 for an illustration). In this case, the HMC algorithm, as described in Section 2.1, for \(q = (x^T, \theta^T)^T\) must be tuned for the most extremely scaled parts of the target distribution to ensure exploration of the complete target distribution. This, in turn lead to a computationally wasteful exploration of the more moderately scaled parts of the target, as the tuning parameters cannot themselves depend on \(q\) (under
regular HMC). In addition, automated tuning of integrator step sizes (and mass matrices) crucially relies on the most extremely scaled parts being visited during the initial tuning phase. If not, they may not be explored at all.

3 Transport maps based on IS densities

To counteract such undesired extreme tuning, while avoiding computationally costly \( q \)-dependent tuning such as RMHMC, the approach taken here involves “preconditioning” the original target so that the resulting modified target is close to Gaussian and thus suitable for statically tuned HMC. Such preconditioning with the aim of producing more tractable target distributions for MCMC methods have a long tradition, and prominent examples are the affine re-parameterizations common for Gibbs sampling applied to regression models (see, e.g., Gelman et al. 2014, Chapter 12). More recent approaches with such ends involve semi-parametric transport map approach of Parno and Marzouk (2018), and, neural transport as described by Hoffman et al. (2019). The approach taken here share many similarities with the dynamically rescaled HMC approach of Kleppe (2019), but the strategy for constructing the transport map considered here is very different and is applicable to more general models.

In a nutshell, a transport map, say \( T \), is a smooth bijective mapping relating the original parameterization \( q \sim \pi_q(q) \) and some modified parameterization \( q' \) via \( q = T(q') \). If \( q' \) is some random draw \( \sim \pi_{q'}(q') = \pi_q(T(q'))|\nabla_q T(q')| \), then a draw distributed according to \( \pi_q \) is achieved by simply applying the transport map to \( q' \). The aim of introducing this construction, is that \( T \) can be chosen so that \( \pi_{q'} \) is loosely speaking "more suitable for MCMC sampling". In practice, this rather vague aim is replaced by making \( \pi_{q'} \) close to a Gaussian distribution with independent components, which can be sampled very efficiently using HMC.

3.1 Transport maps for Bayesian hierarchical models

In the current situation involving a Bayesian hierarchical model, a transport map \( T \) that is non-trivial for the latent variables only,

\[
q = \begin{bmatrix} \theta \\ x \end{bmatrix} = T(q') = \begin{bmatrix} \theta \\ \gamma_{\theta}(u) \end{bmatrix}, \quad q' = \begin{bmatrix} \theta \\ u \end{bmatrix},
\]

is considered. The transport map specific to the latent variables, \( \gamma_{\theta} : \mathbb{R}^D \rightarrow \mathbb{R}^D \) is assumed to be a smooth bijective mapping for each \( \theta \). As we have \( \nabla_u \theta = 0 \) in the above transport map, it follows that \( |\nabla_q T(q')| = |\nabla_u \gamma_{\theta}(u)| \), and thus the modified target distribution has the form:

\[
\tilde{\pi}(\theta, u | y) \propto |\nabla_u \gamma_{\theta}(u)| p(\theta) [p(y|x, \theta) p(x|\theta)]_{x = \gamma_{\theta}(u)}.
\] (3)
Notice in particular that the original parameterization of the latent variables is computed in each evaluation of \( (3) \), and thus obtaining MCMC samples in the \( (\theta, x) = (\theta, \gamma_\theta(u)) \) parameterization comes at no additional cost when MCMC samples targeting \( (3) \) are available.

Further, let \( m(x|\theta) \) denote the density of \( \gamma_\theta(u) \) when \( u \sim N(0_D, I_D) \). In particular, \( m(x|\theta) \) is implicitly related to the underlying standard Gaussian distribution via the change of variable formula: \( \mathcal{N}(u|0_D, I_D) = |\nabla_u \gamma_\theta(u)| m(x|\theta)|_{x=\gamma_\theta(u)} \). Consequently, eliminating the Jacobian determinant in \( (3) \) results in

\[
\hat{\pi}(\theta, u|y) \propto \mathcal{N}(u|0_D, I_D)p(\theta)\omega_\theta(u), \quad \omega_\theta(u) = \left[ \frac{p(y|x, \theta)p(x|\theta)}{m(x|\theta)} \right]_{x=\gamma_\theta(u)}. \tag{4}
\]

Representation \( (4) \) reveal that if \( m(x|\theta) = p(x|y, \theta) \) (i.e. \( \gamma_\theta(u) \sim x|y, \theta \)), the parameters and latent variables exactly “decouples” and \( (3) \) and \( (4) \) reduces to \( \mathcal{N}(u|0_D, I_D)p(\theta|y) \) (see also Lindsten and Doucet 2016 for a similar discussion). Such a situation will be well suited for HMC sampling (provided of course that the marginal likelihood \( p(\theta|y) \) is reasonably well-behaved). Of course, such an ideal situation is in practice unattainable when the model in question is nonlinear/non-Gaussian as neither \( p(\theta|y) \) nor \( p(x|y, \theta) \) will have analytical forms. The strategy pursued here is therefore to take \( m(x|\theta) \) as an approximation to \( p(x|y, \theta) \) in order to obtain an approximate decoupling effect, i.e. so that \( \omega_\theta(u) \) is fairly flat across the region where \( \mathcal{N}(u|0_D, I_D) \) has significant probability mass.

### 3.2 Relation to importance sampling and pseudo-marginal methods

The \( \omega_\theta(u) \) of \( (4) \) is recognized to be an importance weight targeting the marginal likelihood \( p(y|\theta) \) (i.e. \( E_u(\omega_\theta(u)) = p(y|\theta) \)) when \( u \sim N(0_D, I_D) \). This observation is important for at least three reasons. Firstly, it is clear that the large literature on importance sampling- and similar methods for hierarchical models (among many others, Shephard and Pitt 1997, Richard and Zhang 2007, Rue et al. 2009, Durbin and Koopman 2012) may be leveraged to suggest suitable choices for importance density \( m(x|\theta) \) or \( \gamma_\theta(u) \).

Specific choices considered here are discussed in more detail in Section 4.

Secondly, as discussed, e.g., in Koopman et al. (2009), importance sampling-based likelihood estimates such as \( \omega_\theta(u) \) may have infinite variance and thus become unreliable, in particular in high-dimensional applications. This occurs when the tails of \( m(x|\theta) \) are thinner than those of the target distribution \( p(x|\theta, y) \propto p(y|x, \theta)p(x|\theta) \), making \( \omega_\theta(u) \) unbounded as a function of \( u \). However, under the modified target \( (4) \) the likelihood estimate is combined with the thin-tailed standard normal distribution in \( u \), which counteracts the potential unboundedness of the IS weight in the \( u \)-direction. This robustness with respect to the infinite-variance problem is also evident in the representation \( (3) \) of the target, which does not explicitly involve the importance sampling weight. Affine transport maps \( \gamma_\theta(u) \), and consequently thin-tailed Gaussian
importance densities $m(x|\theta)$, lead to the Jacobian determinant $|\nabla_u \gamma_\theta(u)|$ being constant with respect to $u$. Consequently, in this case the tail behavior of $[\mathcal{E}]$ with respect to $u$ will be the same as the tail behavior of $p(\theta, x|y)$ in $x$. Thus, the proposed methodology may be seen as a resolution of the infinite variance problems complicating the application of high-dimensional importance sampling.

Finally, the proposed methodology may be seen as a special case of the pseudo-marginal HMC (PM-HMC) method of [Lindsten and Doucet (2016)]. PM-HMC relies on joint HMC sampling of a Monte Carlo estimate of the marginal likelihood and the random variables used to generate said estimate. [Lindsten and Doucet (2016)] find a similar decoupling effect by admitting their Monte Carlo estimate be based on $n \geq 1$ importance weights (at the cost of increasing the dimensionality of $u$ in their counterpart to (4)), and are to a lesser degree reliant on choosing high-quality importance densities. In particular, [Lindsten and Doucet (2016)] use $m(x|\theta) = p(x|\theta)$ in their illustrations, which for moderately dimensional and low-signal-to noise situations will produce a good decoupling effect for moderate $n$. However, in the present work we focus on high-dimensional applications where it is well known that such “brute force” importance sampling estimators can suffer from prohibitively large variances for any practical $n$ (see, e.g., [Danielsson, 1994]), and thus focus rather on higher fidelity importance densities and $n = 1$.

[Lindsten and Doucet (2016)] also propose a symplectic integrator suitable for HMC applications with target distributions on the form (4) under the “close to decoupling” assumption. In the decoupling case $u \mapsto \omega_\theta(u) \propto 1$, the integrator reduces to a standard leapfrog integrator in the dynamics of $\theta$, whereas the dynamics of $u$ (typically high-dimensional) are simulated exactly. This integrator will be referred to as the LD-integrator in the example applications and is detailed in the supplementary material, Section A.

4 Specific choices of $m(x|\theta)$ and $\gamma_\theta(u)$

As alluded to above, taking $m(x|\theta) = p(x|\theta)$ may in cases where data $y$ are rather un-informative with respect to the latent variable $x$ lead to satisfactory results (see e.g. [Stan Development Team, 2019b], Section 2.5). However, as illustrated by e.g. [Kleppe (2019)], such procedures can lead to misleading MCMC results if data are more informative with respect to the latent variables. An even more challenging situation with $m(x|\theta) = p(x|\theta)$ is when one or more elements of $\theta$ determine how informative the data are with respect to the latent variables (e.g. $\sigma$ when $y_i \sim N(x_i, \sigma^2)$), as this may still lead to a funnel-shaped target distribution. On the other hand, as illustrated by [Kleppe (2019)], rather crude transport maps reflecting only roughly the location and scale of $p(x|y, \theta)$ may lead to dramatic speedups, and the resolution of funnel-related problems. In the rest of this section, two families of strategies for locating transport maps are discussed. Both are well known in the context of importance sampling, and are typically applicable when $p(x|\theta)$ is non-Gaussian.
4.1 $m(x|\theta)$ and $\gamma_\theta(u)$ derived from approximate Laplace approximations

As explained e.g. in [Rue et al. (2009)], the Laplace approximation (also often referred to as the second order approximation) for integrating out latent variables relies on approximating $p(x|y, \theta)$ with a $N(h_\theta, G_\theta^{-1})$ density, where

$$h_\theta = \arg \max_x \log [p(x|\theta)p(y|x, \theta)],$$
$$G_\theta = -\nabla^2_x \log [p(x|\theta)p(y|x, \theta)]_{x=h_\theta}.$$

Namely, the first and second order derivatives of $\log p(x|y, \theta)$ at the mode are matched with the same derivatives of the approximating Gaussian log-density. Due to conditional independence assumptions often involved in modelling, the negative Hessian of $-\log p(x|y, \theta)$ is typically sparse which, when exploited, can substantially speed up the associated Cholesky factorizations.

In the present situation, obtaining the exact mode $h_\theta$ is typically not desirable from a computational perspective. Rather, given an initial guesses for $h_\theta$ and $G_\theta$, say $h_\theta^{(0)}$ and $G_\theta^{(0)}$, a sequence of gradually more refined approximate solutions $h_\theta^{(k)}$ and $G_\theta^{(k)}$ are calculated via iterations of Newton’s method for optimization or an approximation thereof (see supplementary material, Sections C and D for details specific to the models considered shortly).

Finally, for some fixed number of iterations, $K = 0, 1, 2, \ldots$, the transport map is taken to be

$$\gamma_\theta(u) = h_\theta^{(K)} + (L_\theta^{(K)})^{-T} u,$$  \hspace{1cm} (5)

where $L_\theta^{(K)}$ is the lower triangular Cholesky factor of $G_\theta^{(K)}$, so that $m(x|\theta) = N\left(x|h_\theta^{(K)}, [G_\theta^{(K)}]^{-1}\right)$. Notice in particular that the Jacobian determinant of $\gamma_\theta(u)$, required in representation (3) (or in the normalization constant of $m(x|\theta)$ in (4)), takes a particularly simple form, namely $|\nabla_u \gamma_\theta(u)| = |L_\theta^{(K)}|^{-1}$, when applying the affine transport map (5). It should be noted that the applicability of the Laplace approximation relies critically on that $p(x|y, \theta)$ is unimodal and log-concave in a region around the mode that also contains $h_\theta^{(0)}$.

Choices of $h_\theta^{(0)}$, $G_\theta^{(0)}$ and the iteration over $k$ are inherently model specific. However, for a rather general class of models, the initial guesses may be taken to be

$$G_\theta^{(0)} = G_{\theta,x} + G_{\theta,y|x}$$
$$h_\theta^{(0)} = \left(G_\theta^{(0)}\right)^{-1}(G_{\theta,x}h_{\theta,x} + G_{\theta,y|x}h_{\theta,y|x}),$$  \hspace{1cm} (6) (7)

where $h_{\theta,x}$ and $G_{\theta,x}$ are the mean and precision matrix associated with $x|\theta$. Further, $h_{\theta,y|x}$ and $G_{\theta,y|x}$ are
importance density of (4) is simply set equal to the EIS importance density, i.e. the usage of an approximate mode and Hessian). By substituting \( \omega \) connected to the proposed methodology with Laplace approximation-based transport maps. Specifically, evaluation of (3) to one.

\( H \) that do not depend on \( x \) for some fixed set of common random numbers \( u \).

The efficient importance sampler (EIS) algorithm of Richard and Zhang (2007) is a widely used technique for constructing close to optimal importance densities, typically in the context of integrating out latent variables. At its core, the EIS relies initially on eliciting a family of sampling mechanisms, say for constructing close to optimal importance densities, typically in the context of integrating out latent variables. The EIS algorithm proceeds by first sampling a collection of “common random numbers” \( Z = \{z^{(i)}\}_{i=1}^r \), \( z^{(i)} \sim N(0_D, I_D) \), \( i = 1, \ldots, r \), then selecting an initial parameter \( a^{[0]} \), and finally iterate over the below steps for \( j = 1, \ldots, J \):

- Sample latent states \( x^{(i)} = \Gamma_a(z^{(i)}) \), \( i = 1, \ldots, r \).
- Locate a new \( a^{[j]} \) as a (generally approximate) minimizer (over \( a \)) of the sample variance of the importance weights \( w^{(i)}_a = p(y|x^{(i)}, \theta)p(x^{(i)}|\theta)/m_\theta(x^{(i)}) \), \( i = 1, \ldots, r \).

An unbiased estimate of \( p(y|\theta) \) is given by the means of conventional importance sampling (Robert and Casella, 2004, Section 3.3) based on importance density \( m_{a^{[j]}|z}(x) \), with random draws (from \( m_{a^{[j]}|z}(x) \)) generated based on random numbers independent from \( z^{(i)} \), \( i = 1, \ldots, n \).

Notice that the near optimal EIS parameter \( a^{[j]} = a^{[j]}(\theta, Z) \) generally depends both on \( \theta \) and \( Z \). In the present context, for some fixed set of common random numbers \( Z \) and number of EIS iterations \( J \), the importance density of (4) is simply set equal to the EIS importance density, i.e. \( m(x|\theta) = m_{a^{[j]}(\theta, Z)}(x) \).
Notice in particular that the EIS iterations above must be repeated for each evaluation of (4), and that the common random numbers must be kept fixed during each HMC iteration (which typically involve several evaluations of (4) and its gradient), or throughout the whole MCMC simulation.

The EIS importance density is often regarded as more reliable than the Laplace approximation counterpart, as it explicitly seeks to minimize the importance weight variation across typical outcomes of importance density. In addition, the family of importance densities \( m_a(x) \) may be constructed to highly non-Gaussian densities, whereas the Laplace approximation importance density is multivariate Gaussian. On the other hand, the EIS algorithm typically is substantially more costly in a computational perspective, whether this additional computational effort pays off in terms of a better decoupling effect in (3,4) is sought to be answered here.

The sketch of the EIS algorithm above is intentionally kept somewhat vague, as the actual details, both in terms of selecting \( m_a(x) \) and how the optimization step is implemented, depends very much on the model specification at hand. A more detailed description of the EIS suitable for the models considered in the simulation study discussed shortly is given in Section 3 of the supplementary material.

### 4.3 Implementation and Tuning Parameters

The proposed methodology has been implemented in two ways. Firstly, the Laplace approximation-based methods are implemented in Stan using the modified target representation (3). This is also the case for the reference method corresponding to \( m_{\theta}(x) = p(x|\theta) \).

Secondly, we also consider a bespoke HMC implementation as outlined in Section 2.1 for \( q = (\theta^T, u^T)^T \), targeting either (3) for Laplace approximation-based methods) or (4) for EIS-based methods). This HMC method is based on the LD-integrator (see supplementary material, Section A) in order to better exploit the approximate decoupling effects in the target, and was in particular included to explore the advantage of using the LD-integrator over the leapfrog integrator in the present situation.

The mass matrix in the bespoke implementation was taken to be

\[
M = \begin{bmatrix}
\hat{M}_\theta & 0_{d \times D} \\
0_{D \times d} & I_D
\end{bmatrix},
\]

where \( \hat{M}_\theta = -\nabla^2_{\theta} \log [\hat{p}(y|\theta)p(\theta)]_{\theta=\hat{\theta}} \) and the simulated MAP \( \hat{\theta} = \arg \max_\theta \log [\hat{p}(y|\theta)p(\theta)] \) is obtained from an EIS importance sampling estimate \( \hat{p}(y|\theta) \) of \( p(y|\theta) \). Finding the approximate parameter marginal posterior precision \( \hat{M}_\theta \) is very fast and requires minimal additional effort as gradients of the importance weight with respect to \( \theta \) are already available via automatic differentiation (AD, to be discussed shortly).
Notice that the mass matrix specific to \( u \) is take to be the identity to match the precision of the \( N(0_D, I_D) \) “prior” of \( u \) in (3.1). As for the integrator step size \( \varepsilon \) and the number of integrator steps \( L \), we retain \( L \) as a tuning parameter while keeping the total integration time \( \varepsilon L \) per HMC proposal fixed at \( \approx \pi/2 \). This choice of total integration time is informed by the expectation that \( \theta, u | y \) under (3.1) will be close to a Gaussian with precision matrix \( M \). Moreover, whenever \( \tilde{\pi}(q) \) in (1) is Gaussian with precision \( M \), the dynamics (2) are periodic with period \( t = 2\pi \), and choosing a quarter of such a cycle leads to HMC proposals \( q^* \) independent of the current configuration \( q^{(k)} \) (see e.g. [Neal, 2011; Mannseth et al., 2018]). Finally, \( L \) is tuned by hand to obtain acceptance rates around 0.9.

Both implementations rely on the ability to compute gradients of log-targets (3.1) with respect to both \( \theta \) and \( u \). To this end, we rely on Automatic Differentiation (AD). In Stan, this is done automatically, whereas in the bespoke implementation, the Adept C++ automatic differentiation software library (Hogan 2014) is applied. Notice that for the Laplace approximation-based method, AD is applied to calculations of band-Cholesky factorizations, and thus there may be room for improvement in CPU times if the AD libraries supported such operations natively. The bespoke algorithm is implemented using the R (R Core Team, 2019) package Rcpp by Eddelbuettel and François (2011), which makes it possible to run compiled C++ code in R. Stan is used through its R interface rstan (Stan Development Team, 2019a), version 2.19.2. The same C++ compiler was used for both the bespoke and Stan methods. All computations are performed using R version 3.6.1 on a PC with an Intel Core i5-6500 processor running at 3.20 GHz.

5 Simulation study

This section presents applications of the proposed methodology to three non-Gaussian/nonlinear state-space latent variable models for the purpose of benchmarking against alternative methods. State-space models with univariate state were chosen as the Laplace approximation-based methods only require tri-diagonal Cholesky factorizations, which are easily implemented in the Stan language. The specific models are selected to illustrate the performance under different, empirically relevant, scenarios. In particular, the three models exhibit significantly different, and variable signal-to-noise ratios, which as discussed above may modulate the need for (non-trivial) transport map methods.

In the proceeding, different combinations of implementation (\( \in \{\text{Stan, LD}\} \)) and transport map method (\( \in \{\text{Prior, Laplace, EIS, Fisher}\} \)) are considered, where “LD” refers to the bespoke HMC implementation with LD integrator. Transport map “Prior” correspond to \( m_\theta(x) = p(x|\theta) \) and is equivalent to carrying out the simulations in an \((\theta, \eta)\)-parameterization where \( \eta = (\eta_1, \ldots, \eta_D)' \) are a-priori standard normal disturbances of the models to be discussed.
Transport map method “Fisher” corresponds to Fisher information-based DRHMC approach of [Kleppe (2019)] applied to the latent variables only (i.e. general DRHMC involves non-trivial transport maps for the parameters also). Fisher also leads to an affine transport map \( \gamma_{\theta}(u) = h_F + L_F^T u, L_F L_F^T = G_F \). Here, \( G_F \) is the sum of the a-priori precision matrix of \( x \) and the Fisher information of the observations with respect to \( x \). Notice that this method requires both that said Fisher information is constant with respect to the latent state, and that the \( p(x|\theta) \) precision matrix has closed form, where the latter requirement limits its applicability to the first two models considered below.

Methods LD-Prior and LD-Fisher were not carried out as the default tuning discussed in Section 4.3 work poorly in these cases. Moreover, Stan-EIS was also not considered as it was impractical to implement the EIS algorithm in the Stan language. For each of the three models, the LD algorithm is simulated for 1,500 iterations, where the draws from the first 500 burn-in iterations are discarded. Stan uses (the default) 2,000 burn-in steps also used for automatic tuning of the integrator step size and the mass matrix. The reported computing times are for the 1,000 sampling iterations for both methods. Further details for the different example models, including prior assumptions and details related to the Newton iterations for the Laplace maps, are found in the supplementary material, Section C.

5.1 Stochastic Volatility Model

The first example model is the discrete-time stochastic volatility (SV) model for financial returns given by [Taylor (1986)]

\[
\begin{align*}
y_t &= \exp(x_t/2)e_t, \quad e_t \sim \text{iid } N(0,1), \; t = 1, \ldots, D, \\
x_t &= \gamma + \delta x_{t-1} + \nu \eta_t, \quad \eta_t \sim \text{iid } N(0,1), \; t = 2, \ldots, D,
\end{align*}
\]

where \( y_t \) is the return observed on day \( t \), \( x_t \) is the latent log-volatility with initial condition \( x_1 \sim N(\gamma/[1 - \delta], \nu^2/[1 - \delta^2]) \), while \( e_t \) and \( \eta_t \) are mutually independent innovations. The data consists of daily log-returns on the U.S. dollar against the U.K. Pound Sterling from October 1, 1981 to June 28, 1985 with \( D = 945 \).

Under this SV model the data density \( p(y_t|x_t) = \mathcal{N}(y_t|0, \exp\{x_t\}) \) is fairly uninformative about the states \( x_t \), with a Fisher information (w.r.t. \( x_t \)) which is independent of \( \theta \) and given by \(-E[\nabla^2_x \log p(y_t|x_t)] = 1/2\), whereas the states are fairly volatile under typical estimates for \( \theta \). This low signal-to-noise ratio together with a shape of the data density which is independent of the parameters implies that the conditional posterior of the innovations \( \eta \) given \( \theta \) are close to a normal distribution regardless of \( \theta \), leading to a correspondingly well-behaved joint posterior of \( \theta \) and \( \eta \). Hence, this represents a scenario where the Stan-Prior sampling on the joint space of \( \theta \) and \( \eta \) used as a benchmark can be expected to exhibit a comparably good performance.
Table 1: Simulation study results for the SV model (8,9). ESS corresponds to the effective sample size (out of 1,000 iterations) and ESS/s is the number of effective samples produced per second of computing time. The columns “Min”, “Mean” correspond to the minimum, mean across 8 independent replicas of the experiment. Burn-in iterations are not included in the reported CPU times. The tuning parameters are: LD-EIS: $J = 2$, $r = 6$, $\varepsilon = 0.4$ and $L = 4$. LD-Laplace: $K = 2$, $\varepsilon = 0.4$ and $L = 4$. Stan-Laplace: $K = 0$.

|               | LD-EIS |             | Stan-Prior |             | LD-Laplace |             | Stan-Laplace |             | Stan-Fisher |             |
|---------------|--------|-------------|------------|-------------|------------|-------------|--------------|-------------|-------------|-------------|
|                | Min    | Mean        | Min        | Mean        | Min        | Mean        | Min          | Mean        | Min         | Mean        |
| CPU time (s)   | 276.5  | 278         | 12.4       | 15          | 10.6       | 10.6        | 9.7          | 16.7        | 6.1         | 7.6         |
| $\gamma$      |        |             |            |             |            |             |              |             |             |             |
| Post. mean    | -0.021 | -0.021      | -0.021     | -0.021      | -0.021     | -0.021      | -0.021       | -0.021      | -0.021      | -0.021      |
| Post. std.    | 0.012  | 0.01        | 0.011      | 0.011       | 0.011      | 0.011       | 0.011        | 0.011       | 0.011       | 0.011       |
| ESS           | 201    | 337         | 237        | 348         | 275        | 354         | 268          | 494         | 218         | 321         |
| ESS/s         | 0.7    | 1.2         | 18.1       | 23.5        | 25.7       | 33.3        | 16.7         | 37.2        | 5.6         | 27          |
| $\delta$      |        |             |            |             |            |             |              |             |             |             |
| Post. mean    | 0.98   | 0.98        | 0.98       | 0.98        | 0.98       | 0.98        | 0.98         | 0.98        | 0.98        | 0.98        |
| Post. std.    | 0.01   | 0.01        | 0.01       | 0.01        | 0.01       | 0.01        | 0.01         | 0.01        | 0.01        | 0.01        |
| ESS           | 269    | 380         | 192        | 309         | 320        | 363         | 290          | 423         | 239         | 319         |
| ESS/s         | 1      | 1.4         | 15.3       | 20.6        | 30.1       | 34.1        | 13.9         | 32          | 5           | 27          |
| $\nu$         |        |             |            |             |            |             |              |             |             |             |
| Post. mean    | 0.15   | 0.15        | 0.15       | 0.15        | 0.15       | 0.15        | 0.15         | 0.15        | 0.15        | 0.15        |
| Post. std.    | 0.03   | 0.03        | 0.03       | 0.03        | 0.03       | 0.03        | 0.03         | 0.03        | 0.03        | 0.03        |
| ESS           | 363    | 503         | 243        | 332         | 360        | 512         | 274          | 431         | 226         | 293         |
| ESS/s         | 1.3    | 1.8         | 16.7       | 23          | 33.9       | 48.1        | 14.1         | 32.8        | 3.8         | 25.6        |

For the Fisher transport map method, $G_F = G_{\theta,x} + G_{\theta,y|x}$, and as suggested by Table 4 of Kleppe (2019), we set $h_F = 0_d$.

Table 1 shows the HMC posterior mean and standard deviation for the parameters, which are sample averages computed from 8 independent replications. It also reports the effective sample size (ESS) (Geyer, 1992) and the ESS per second of CPU time (ESS/s), where the latter will be the main performance measure (provided of course that the MCMC method properly explores the target distribution) considered here. Several settings of the tuning parameters (i.e. some subset of $r$, $J$, $K$, and $L$) where considered, and the presented results are the best considered, in terms of ESS/s. Table 1 indicates firstly that all five methods produce a good exploration of the target distribution with posterior moments being essentially the same. For the Stan-based methods, there is substantial variation in the CPU times due to variation in the automatic tuning of the integrator step size $\varepsilon$ over the replica. Judging from the ESS values, on average there is not much to be gained from introducing the Laplace approximation- and EIS-based transport map for this model. This finding mirrors to some extent what was found by Kleppe (2019, Section 5.2), and is also as expected since the observations carry very little information regarding the states. In terms of ESS/s, there is no uniform winner, but the computational overhead of locating the EIS importance density is clearly not worthwhile for this model, relative to the computationally cheaper Laplace- and Fisher transport maps.
5.2 Gamma Model for Realized Volatilities

The second example model is a dynamic state-space model for the realized variance of asset returns (see, e.g., Golosnoy et al. 2012, and references therein). It has the form

\[ y_t = \beta \exp(x_t) e_t, \quad e_t \sim \text{iid } G(1/\tau, \tau), \quad t = 1, \ldots, D, \]

\[ x_t = \delta x_{t-1} + \nu \eta_t, \quad \eta_t \sim \text{iid } N(0, 1), \quad t = 2, \ldots, D, \]

where \( y_t \) is the daily realized variance measuring the latent integrated variance \( \beta \exp(x_t) \), and \( G(1/\tau, \tau) \) denotes a Gamma-distribution for \( e_t \) normalized such that \( E(e_t) = 1 \) and \( \text{Var}(e_t) = \tau \). The innovations \( e_t \) and \( \eta_t \) are independent and the initial condition for the log-variance is \( x_1 \sim N(0, \nu^2/[1 - \delta^2]) \). This Gamma volatility model is applied to a data set consisting of \( D = 2,514 \) observations of the daily realized variance for the American Express stock (more information concerning the data is given in Section 6; \( y_t \) here is identical to the 1,1-element of realized covariance matrices \( Y_t \)).

In contrast to the SV model, this Gamma model applied to the realized variance data has both a considerably higher signal-to-noise ratio and a shape of the data density \( x_t \mapsto p(y_t|x_t, \theta) \) which depends on the parameters. In particular, the Fisher information of its data density with respect to \( x_t \) is \( 1/\tau \) with an estimate of \( \tau \simeq 0.13 \) (see Table 2), while the estimated volatility of the states is roughly as large as under the SV model. Hence, it can be expected that the conditional posterior of the innovations \( \eta \) given \( \theta \) deviates distinctly from a Gaussian form and exhibits nonlinear dependence on \( \theta \), which makes the Gamma model a more challenging scenario for the Stan-Prior benchmark than the SV model.

The same initial guess \( h^{(0)} \) in the Laplace scaling as for the SV model above was applied, and also here \( G_F \) coincides with \( G_{\theta, x} + G_{\theta, y|x}. \) Choosing \( h_F = 0 \) leads to poor results, and we therefore set \( h_F \) equal to \( (7) \) (see also Kleppe 2019, Equation 20). Consequently, Stan-Fisher coincides with Stan-Laplace, \( K = 0 \) (which was also found to be the optimal Stan-Laplace method in this situation). The remaining experiment setup is also identical to that for the SV model, and the results are given in Table 2. Stan-Prior produces substantially lower ESSes than the EIS- and Laplace methods, which we attribute to the failure to take the higher information content from the observations into account in the transport map. LD-Laplace and Stan-Laplace are the winners in terms of ESS/s and again it is not beneficial to opt for the presumably more accurate and expensive EIS-transport map over the cruder and computationally faster Laplace-approximation.
Table 2: Simulation study results for the Gamma model \([10,11]\). ESS corresponds to the effective sample size (out of 1,000 iterations) and ESS/s is the number of effective samples produced per second of computing time. The columns “Min”, “Mean” correspond to the minimum, mean across 8 independent replicas of the experiment. Burn-in iterations are not included in the reported CPU times. The tuning parameters are: LD-EIS: \(J = 2\), \(r = 5\), \(\varepsilon = 0.64\) and \(L = 3\), LD-Laplace: \(K = 1\), \(\varepsilon = 0.64\) and \(L = 3\). Stan-Laplace: \(K = 0\). Notice that Stan-Fisher and Stan-Laplace coincide in this case.

5.3 Constant Elasticity of Variance Diffusion Model

The last example model is a time-discretized version of the constant elasticity of variance (CEV) diffusion model for short-term interest rates (Chan et al., 1992), extended by a measurement error to account for microstructure noise (Aït-Sahalia, 1999; Kleppe and Skaug, 2016). The resulting model for the interest rate \(y_t\) observed at day \(t\) with a corresponding latent state \(x_t > 0\), is described as

\[
y_t = x_t + \sigma_y e_t, \quad e_t \sim \text{iid } N(0, 1), \quad t = 1, \ldots, D,
\]

\[
x_t = x_{t-1} + \Delta(\alpha - \beta x_{t-1}) + \sigma_x x_{t-1}^\gamma \sqrt{\Delta} \eta_t, \quad \eta_t \sim \text{iid } N(0, 1), \quad t = 2, \ldots, D,
\]

where \(e_t\) and \(\eta_t\) are mutually independent and \(\Delta = 1/252\). The parameters are \(\theta = (\alpha, \beta, \gamma, \sigma_x, \sigma_y)\) and the initial condition \(x_1 \sim N(y_1, 0.01^2)\). The data consist of \(D = 3,082\) daily 7-day Eurodollar deposit spot rates from January 2, 1983 to February 25, 1995 (see Aït-Sahalia, 1996 for a description of this data set).

The estimated standard deviation of the noise component \(\sigma_y\) is very small with an estimate of 0.0005 (see
|                  | LD-EIS |               | LD-Laplace |               | Stan-Laplace |
|------------------|--------|--------------|-----------|--------------|--------------|
|                  | Min    | Mean         | Min       | Mean         | Min          | Mean         |
| CPU time (s)     | 615.6  | 618.8        | 60.3      | 60.6         | 482.2        | 515.7        |
| $\alpha$ Post. mean | 0.01   | 0.01         | 0.01      | 0.01          | 0.01          |
| $\alpha$ Post. std. | 0.01   | 0.01         | 0.01      | 0.01          | 0.01          |
| $\alpha$ ESS     | 869    | 984          | 876       | 972          | 1000         | 1000         |
| $\alpha$ ESS/s   | 1.4    | 1.6          | 14.5      | 16           | 1.9          | 1.9          |
| $\beta$ Post. mean | 0.17   | 0.17         | 0.17      | 0.17          | 0.17          |
| $\beta$ Post. std. | 0.17   | 0.17         | 0.17      | 0.17          | 0.17          |
| $\beta$ ESS     | 707    | 963          | 745       | 957          | 1000         | 1000         |
| $\beta$ ESS/s   | 1.1    | 1.6          | 12.4      | 15.8         | 1.9          | 1.9          |
| $\gamma$ Post. mean | 1.18   | 1.18         | 1.18      | 1.18          | 1.18          |
| $\gamma$ Post. std. | 0.06   | 0.06         | 0.06      | 0.06          | 0.06          |
| $\gamma$ ESS     | 759    | 957          | 1000      | 1000         | 631          | 852          |
| $\gamma$ ESS/s   | 1.2    | 1.5          | 16.4      | 16.5         | 1.3          | 1.6          |
| $\sigma_x$ Post. mean | 0.41   | 0.41         | 0.41      | 0.41          | 0.41          |
| $\sigma_x$ Post. std. | 0.06   | 0.06         | 0.06      | 0.06          | 0.06          |
| $\sigma_x$ ESS     | 769    | 946          | 1000      | 1000         | 650          | 890          |
| $\sigma_x$ ESS/s   | 1.2    | 1.5          | 16.4      | 16.5         | 1.3          | 1.7          |
| $\sigma_y$ Post. mean | 0.0005 | 0.0005       | 0.0005    | 0.0005        | 0.0005       |
| $\sigma_y$ Post. std. | 0.0002 | 0.0002       | 0.0002    | 0.0002        | 0.0002       |
| $\sigma_y$ ESS     | 769    | 963          | 1000      | 1000         | 1000         | 1000         |
| $\sigma_y$ ESS/s   | 1.2    | 1.6          | 16.4      | 16.5         | 1.9          | 1.9          |

Table 3: Simulation study results for the CEV model (12,13). ESS corresponds to the effective sample size (out of 1,000 iterations) and ESS/s is the number of effective samples produced per second of computing time. The columns “Min”, “Mean” correspond to the minimum, mean across 8 independent replicas of the experiment. Burn-in iterations are not included in the reported CPU times. The tuning parameters are: LD-EIS: $J = 1$, $r = 7$, $\epsilon = 0.57$ and $L = 3$. LD-Laplace: $K = 2$, $\epsilon = 0.57$ and $L = 3$, Stan-Laplace: $K = 1$.

Table 3) so that the data density $x_t \mapsto p(y_t|x_t, \theta)$ is strongly peaked at $x_t = y_t$ and by far more informative about $x_t$ than in the SV- and Gamma model with a Fisher information given by $1/\sigma_y^2$. Also, the volatility of the states is not constant and depends, unlike in the previous models, nonlinearly on the level of the states. As a result, the posterior of $\eta$ and $\theta$ strongly deviates from being Gaussian. Consequently, Stan-Prior fails to produce meaningful results and is therefore not reported on. Moreover, since the prior on $x$ is nonlinear and its precision matrix does not seem to have closed-form, Fisher-scaling is not feasible.

Table 3 reports results for LD-EIS, LD-Laplace and Stan-Laplace, and it is seen that all three methods produce reliable results. In terms of ESS per computing time, the LD-Laplace is a factor 5-10 faster than the other methods, where the difference between LD-Laplace and Stan-Laplace is due to the substantially higher number of integrator steps required for Stan-Laplace.

The same model and data set was also considered by Kleppe (2018, Section 5), who compare the modified
Cholesky Riemann manifold HMC algorithm and a Gibbs sampling procedure. Both methods were implemented in C++ and thus the orders of magnitude of produced ESS per computing time are comparable to the present situation. It is seen that for the “most difficult” parameters $\gamma$, $\sigma_x$, the proposed methodology is roughly two order of magnitude faster than the Riemann manifold HMC method and roughly three orders of magnitude faster than the Gibbs sampler.

5.4 Summary from simulation experiment

For models with higher signal-to-noise ratios than the SV model, the proposed methodology produces large speedups (or makes challenging models feasible as for the CEV model) relative to the benchmarks, even if the per evaluation cost of the modified target is higher than in the default parameterization. For the considered models, the EIS transport map is not competitive relative to the Laplace approximation counterpart due to the relatively higher computational cost. For the Laplace-based methods, it is seen that relatively few Newton iterations is optimal in an ESS per computing time perspective. Overall, and very much in line with [Kleppa (2019)], this is indicative that rather crude representations of the location and scale of $p(x|y, \theta)$ are sufficient. Moreover, this latter observation ties in with the second point discussed in Section 3.2. Due to the thin-tailed Gaussian distribution entering explicitly in representation (4) of the modified target, the importance sampling rule of thumb that you should seek high-fidelity approximations to $p(x|y, \theta)$ as the importance density is less relevant in the present situation.

With respect to the choice of integrator, it is seen that the LD-integrator and the leapfrog-integrator-based Stan produces similar raw ESSes, but that that the LD-integrator in general requires non-trivially fewer integration steps to accomplish this. E.g., the reported (automatically tuned) Stan-Laplace results for the CEV model required on average 63 leapfrog steps whereas the corresponding (manually tuned) number for LD-Laplace was 3. For the two other models, the performance of the LD integrator is roughly on par with Stan when Laplace scaling was employed. Further, the LD integrator generally needs more refined Laplace maps (higher $K$) to work satisfactory, whereas under Stan, more crude Laplace transport maps are permissible.

6 High-dimensional application

6.1 Model

To illustrate the proposed methodology in a high-dimensional situation, we consider the dynamic inverted Wishart model for realized covariance matrices proposed in [Grothe et al. (2019) Section 6]. More specifically,
for a time series of \( r \times r \) symmetric positive definite observed realized covariance matrices \( \mathbf{Y}_t, t = 1, \ldots, D \), the observations are modeled conditionally inverse-Wishart distributed,

\[
p(\mathbf{Y}_t | \Sigma_t, \nu) \propto |\mathbf{Y}_t|^{-\frac{\nu + r + 1}{2}} \exp\left(-\frac{1}{2} \text{tr} (\Sigma_t \mathbf{Y}_t^{-1})\right),
\]

so that \( E(\mathbf{Y}_t) = (\nu + r + 1)^{-1} \Sigma_t \). Here, the degrees of freedom \( \nu > r + 1 \) is a parameter, and \( \Sigma_t \) is a (latent) time-varying scale matrix, given by

\[
\Sigma_t = \mathbf{H} \mathbf{D}_t \mathbf{H}^T, \quad \mathbf{D}_t = \text{diag}(\exp(x_{1,t}), \ldots, \exp(x_{r,t})),
\]

where \( \mathbf{H} \) is a lower triangular matrix with ones along the main diagonal and unrestricted parameters \( h_{i,j}, i > j, 1 \leq j < r \) below the main diagonal. Moreover, \( \mathbf{x}_s = \{x_{s,t}\}_{t=1}^D, s = 1, \ldots, r \) are latent Gaussian AR(1) processes

\[
x_{s,t} = \mu_s + \delta_s (x_{s,t-1} - \mu_s) + \sigma_s \eta_{s,t}, \quad t = 2, \ldots, D, \quad s = 1, \ldots, r,
\]

where \( \eta_{s,t} \sim \text{iid } \mathcal{N}(0,1), \quad t = 1, \ldots, D, \quad s = 1, \ldots, r. \) In total, the model contains \( 1 + 3r + r(r - 1)/2 \) parameters \( \theta = (\nu, \mu_{1:r}, \delta_{1:r}, \sigma_{1:r}, h_{2:r,1}, h_{3:r,2}, \ldots, h_{r,r-1}). \) Further details concerning the model specification and priors can be found in the supplementary material (Section D).

A fortunate property of this model is that the conditional posterior of the latent states are independent over \( s \), i.e. \( p(\mathbf{x}_{1:D} | \theta, \mathbf{Y}_{1:D}) = \prod_{s=1}^{r} p(\mathbf{x}_s | \theta, \mathbf{Y}_{1:D}) \). This implies that the transport map for \( \mathbf{x} \) also may be split into \( r \) individual transport maps, say \( \mathbf{x}_s = \gamma_{\theta,s}(\mathbf{u}_s), \quad \mathbf{u}_s = \{u_{s,t}\}_{t=1}^D, \quad s = 1, \ldots, r, \) without losing fidelity. The (combined) transport map becomes \( \gamma_{\theta}(\mathbf{u}) = [(\gamma_{\theta,1}(\mathbf{u}_1))^T, \ldots, (\gamma_{\theta,r}(\mathbf{u}_r))^T]^T \), where \( \mathbf{u} = [\mathbf{u}_1^T, \ldots, \mathbf{u}_r^T]^T \), and in particular \( |\nabla_{\mathbf{u}} \gamma_{\theta}(\mathbf{u})| = \prod_{s=1}^{r} |\nabla_{\mathbf{u}_s} \gamma_{\theta,s}(\mathbf{u}_s)| \) due to the block-diagonal nature of the Jacobian of \( \gamma_{\theta} \).

Further, each of the factors of the conditional posterior have a shape corresponding that of a state-space model with univariate state-process \( \mathbf{x}_s \):

\[
p(\mathbf{x}_s | \theta, \mathbf{Y}_{1:D}) \propto p(\mathbf{x}_s | \theta) \prod_{t=1}^{D} \exp\left(\frac{\nu}{2} x_{s,t} - \frac{1}{2} \text{exp}(x_{s,t})\right), \quad \tilde{y}_{s,t} = (\mathbf{H}_{1:s,s})^T \mathbf{Y}_t^{-1} \mathbf{H}_{1:s,s}, \quad s = 1, \ldots, r.
\]

Thus, individual transport maps \( \gamma_{\theta,s} \) may be constructed to target \( [17] \) as described in the previous Sections. In particular, individual Laplace approximation-based maps, \( \gamma_{\theta,s} \), involve only tri-diagonal Cholesky factorizations. It is, however, worth noticing that the proposed methodology does not rely on such a conditional independence structure in order to be applicable per se.
Table 4: Effective sample sizes and CPU times for the inverse Wishart model (14-16). The parameters are grouped, and the reported ESS figures are (min, max) across each group. All of the results are averages across 8 independent replica of each experiment. Here, $u_{s,1}$ is the first element in $u_s$. Under Prior transport map, $u_{1:5,1}$ is identical to $\eta_{1:5,1}$ in (16).

The observed Fisher information (w.r.t. $x_{s,t}$) of the marginal “measurement densities” $\propto \exp(\frac{\nu}{2} x_{s,t} - \frac{y_{s,t}}{2} \exp(x_{s,t}))$ equals $\nu/2$, with an estimate of $\nu \simeq 33.6$ for the data set considered here (see Table 5 in supplementary material). Thus, the signal to noise ratio here is similar to that of the Gamma model considered in section 5.2. As the LD- and Stan- results are similar for the Gamma model, we consider only Stan for this model, as it entails only a few dozen lines of Stan code and tuning is fully automated. EIS was found not to be competitive and is not considered here. The initial guess $h_0(\theta)$ under Laplace scaling is given by (7), whereas $G_0^{(K)} = G_0^{(0)}$ given in (6). This (fixed) matrix was also used as the scaling matrix in the approximate Newton iterations for $K = 0, 1, 2$ (see supplementary material, Section D for more details).

### 6.2 Data and results

The data set of $D = 2,514$ observations of daily realized covariance matrices of $r = 5$ stocks (American Express, Citigroup, General Electric, Home Depot, and IBM) spanning Jan. 1st, 2000 to Dec. 31, 2009 is described in detail in Golosncoy et al. (2012). The same model and data set was considered in Grothe et al. (2019), where Gibbs sampling procedures were considered. From Grothe et al. (2019), it is seen that even with close to iid sampling from $p(x_{1:r} | \theta, Y_{1:D})$, the chains for $\nu$ and $\sigma_s$, $s = 1, \ldots, r$ mix rather poorly under Gibbs sampling.

The ESSes for the parameters and the first elements of $x_s$ and $u_s$, and CPU times for Stan-Prior and Stan-Laplace are given in Table 4. Corresponding posterior means and standard deviations for Stan-Laplace ($K = 0$) are given in Table 5 in the supplementary material and these are very much in line with Grothe et al. (2019, Table 5).

From Table 4 it is seen that the proposed methodology Stan-Laplace outperforms the benchmark Stan-Prior, both in terms CPU time (the modified target is highly non-Gaussian and thus requires many integration
Figure 1: Posterior mean and standard deviation of $u_s$, $s = 1, \ldots, 5$, for the inverse Wishart model (14-16) under Laplace transport map with $K = 0$. The results are for a single representative simulation replica with 1000 sampling iterations.

steps) and ESS. Indeed, Stan-Laplace with $K = 0$ is at least a full order of magnitude faster in terms of ESS per CPU time than Stan-Prior for the “difficult” parameters $\nu$ and $\sigma_s$, $s = 1, \ldots, r$. The added per evaluation computational cost of the more accurate Laplace approximations ($K = 1$ and $K = 2$) is not worthwhile, and this again corroborates the finds above that only crude location- and scale information with respect to $p(x_{1:r} | \theta, Y_{1:D})$ is needed. Figure 1 depicts the posterior mean and (marginal) standard deviation of each $u_s$, for Stan-Laplace with $K = 0$. It is seen that the posterior standard deviations are close to 1, which one would expect in the case of close to perfect decoupling, i.e. is indicative that any funnel effects have been removed. The posterior means, on the other hand, are somewhat off 0, which is related both to the usage of the initial guess (7) and the fact that (17) is non-Gaussian and thus cannot be exactly decoupled using a Gaussian importance density. Figures 2,3 in the supplementary material shows corresponding plots for $K = 2$ and $K = 10$, and it is seen that the posterior means of $u_s$ are closer to zero, but some deviation still exact due to the non-Gaussian target.

Comparing the computational performance to the Gibbs sampler in Grothe et al. (2019), it is seen that Stan-Laplace is also roughly an order of magnitude faster than a Gibbs sampler. This comparison is somewhat complicated by that Grothe et al. (2019) employ parallel processing (over $s$) when sampling the latent states $x_s$, and that the computations in Grothe et al. (2019) are done in MATLAB, whereas Stan is based on compiled C++ code. In this consideration, also the fact that a model with 20 parameters and 12,570 latent variables can be fitted using a few minutes of CPU time and minimal coding efforts in Stan must be weighed against the typically time consuming and error-prone development efforts to develop Gibbs
samplers tailored for any given model.

7 Discussion

The paper proposes and evaluates importance sampler-based transport map HMC for Bayesian hierarchical models. The methodology relies on using off-the-shelf importance sampling strategies for high-dimensional latent variables to construct a modified target distribution that is easily sampled using (fixed metric) HMC. Indeed, as illustrated, the proposed methodology can lead to large speedups relative to relevant benchmarks for models with high-dimensional latent variables, while still being easily implemented using e.g. Stan.

Two strategies for selecting the involved importance samplers were considered in order to assess the optimal accuracy versus computational cost-tradeoff. The main insight in this regard is that only rather crude importance densities/transport maps (e.g. Laplace or DRHMC-type) are required when these are applied in the present framework. This observation is very much to the contrary to the importance sampling literature at large, where typically very accurate importance densities are required to produce reliable approximations to marginal likelihood functions when integrating over high-dimensional latent variables.

The proposed methodology, with Laplace transport maps and few or no Newton iterations lead to similar transport maps as those used in DRHMC in the cases where DRHMC is applicable. Thus the Laplace transport map approach may, in a rather broad sense, be seen as a generalization of DRHMC to models with nonlinear structures where DRHMC is not applicable.

Finally, there is scope for future research in developing software that can encompass a large class of models, and which implements the proposed methodology in a user-friendly manner. In particular, such software should include a sparse Cholesky algorithm for more general sparsity structures so that Laplace-based transport maps for e.g. multivariate latent state dynamic models and spatial models can be considered.

References

Aït-Sahalia, Y. (1996). Testing continuous-time models of the spot interest rate. The Review of Financial Studies 9(2), 385–426.

Aït-Sahalia, Y. (1999). Transition densities for interest rate and other nonlinear diffusions. The Journal of Finance 54(4), 1361–1395.

Andrieu, C., A. Doucet, and R. Holenstein (2010). Particle Markov chain Monte Carlo methods. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72(3), 269–342.
Chan, K. C., G. A. Karolyi, F. A. Longstaff, and A. B. Sanders (1992). An empirical comparison of alternative models of the short-term interest rate. *The Journal of Finance* 47(3), 1209–1227.

Danielsson, J. (1994). Stochastic volatility in asset prices: Estimation with simulated maximum likelihood. *Journal of Econometrics* 64, 375–400.

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

Durbin, J. and S. J. Koopman (2012). *Time Series Analysis by State Space Methods* (2 ed.). Number 38 in Oxford Statistical Science. Oxford University Press.

Eddelbuettel, D. and R. François (2011). Rcpp: Seamless R and C++ Integration. *Journal of Statistical Software* 40(8), 1–18.

Flury, T. and N. Shephard (2011). Bayesian inference based only on simulated likelihood: particle filter analysis of dynamic economic models. *Econometric Theory* 27(Special Issue 05), 933–956.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. Rubin (2014). *Bayesian Data Analysis* (3 ed.). CRC Press.

Geyer, C. (1992). Practical Markov chain Monte Carlo. *Statistical Science* 7(4), 473–483.

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

Golosnoy, V., B. Gribisch, and R. Liesenfeld (2012). The conditional autoregressive Wishart model for multivariate stock market volatility. *Journal of Econometrics* 167(1), 211–223.

Gómez-Rubio, V. and H. Rue (2018). "markov chain monte carlo with the integrated nested laplace approximation". *Statistics and Computing* 28(5), 1033–1051.

Grothe, O., T. S. Kleppe, and R. Liesenfeld (2019). The Gibbs sampler with particle efficient importance sampling for state-space models. *Econometric Reviews* 38(10), 1152–1175.

Hoffman, M., P. Sountsov, J. V. Dillon, I. Langmore, D. Tran, and S. Vasudevan (2019). NeuTra-lizing Bad Geometry in Hamiltonian Monte Carlo Using Neural Transport. arXiv:1903.03704.

Hoffman, M. D. and A. Gelman (2014). The No-U-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* 15(1), 1593–1623.
Hogan, R. J. (2014). Fast reverse-mode automatic differentiation using expression templates in c++. ACM Transactions on Mathematical Software (TOMS) 40(4), 26.

Jacquier, E., N. G. Polson, and P. E. Rossi (1994). Bayesian analysis of stochastic volatility models. Journal of Business & Economic Statistics 12(4), 371–89.

Kleppe, T. S. (2018). Modified Cholesky Riemann manifold Hamiltonian Monte Carlo: exploiting sparsity for fast sampling of high-dimensional targets. Statistics and Computing 28(4), 795–817.

Kleppe, T. S. (2019). Dynamically rescaled Hamiltonian Monte Carlo for Bayesian hierarchical models. Journal of Computational and Graphical Statistics 28(3), 493–507.

Kleppe, T. S. and H. J. Skaug (2016). Bandwidth selection in pre-smoothed particle filters. Statistics and Computing 26(5), 1009–1024.

Koopman, S. J., N. Shephard, and D. Creal (2009). Testing the assumptions behind importance sampling. Journal of Econometrics 149(1), 2 – 11.

Leimkuhler, B. and S. Reich (2004). Simulating Hamiltonian dynamics. Cambridge University Press.

Lindsten, F. and A. Doucet (2016). Pseudo-Marginal Hamiltonian Monte Carlo. arXiv preprint arXiv:1607.02516.

Mannseth, J., T. S. Kleppe, and H. J. Skaug (2018). On the application of improved symplectic integrators in Hamiltonian Monte Carlo. Communications in Statistics-Simulation and Computation 47(2), 500–509.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo 2, 113–162.

Parno, M. and Y. Marzouk (2018). Transport map accelerated Markov chain Monte Carlo. SIAM/ASA Journal on Uncertainty Quantification 6(2), 645–682.

Pitt, M. K., R. dos Santos Silva, P. Giordani, and R. Kohn (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. Journal of Econometrics 171(2), 134 – 151.

R Core Team (2019). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.

Richard, J.-F. and W. Zhang (2007). Efficient high-dimensional importance sampling. Journal of Econometrics 141(2), 1385–1411.
Robert, C. and G. Casella (2004). *Monte Carlo methods* (2 ed.). Springer, Berlin.

Rue, H., S. Martino, and N. Chopin (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 71(2), 319–392.

Shephard, N. and M. K. Pitt (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika* 84(3), 653–667.

Stan Development Team (2019a). RStan: the R interface to Stan, Version 2.19.2. http://mc-stan.org.

Stan Development Team (2019b). *Stan user’s guide, version 2.21*. http://mc-stan.org.

Taylor, S. J. (1986). *Modelling Financial Time Series*. Wiley, Chichester.

Tierney, L. and J. B. Kadane (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association* 81(393), 82–86.

Zhang, Y. and C. Sutton (2014). Semi-separable Hamiltonian Monte Carlo for inference in Bayesian hierarchical models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger (Eds.), *Advances in Neural Information Processing Systems* 27, pp. 10–18. Curran Associates, Inc.
Supplementary Material for “Importance Sampling-based Transport map Hamiltonian Monte Carlo for Bayesian Hierarchical Models”

Equation numbers <18 refer to the equations in the main text.

A The Lindsten and Doucet (2016)-integrator

The the pseduo-marginal HMC (PM-HMC) algorithm of Lindsten and Doucet (2016) can be viewed as a standard HMC algorithm for simulating the random vector $q = (\theta', u')'$ from the modified target densities (3) or (4). Proceeding with representation (4), the Hamiltonian is taken to be

$$H(\theta, u, p_\theta, p_u) = -\log \omega_\theta(u) - \log p(\theta) + \frac{1}{2}u'u + \frac{1}{2}p_\theta M_\theta^{-1}p_\theta + \frac{1}{2}p'u'p_u,$$

(18)

where $p_\theta \in \mathbb{R}^d$ and $p_u \in \mathbb{R}^D$ are the artificial momentum variables specific to $\theta$ and $u$, respectively. Note that for this form of the extended Hamiltonian the mass matrix $(M)$ of the compound vector $(\theta', u')'$ is selected to be block diagonal, where the mass matrix specific to $\theta$ is denoted by $M_\theta \in \mathbb{R}^{d \times d}$, while the mass for $u$ is set equal to the identity in order to match the a-priori precision matrix of $u$. Straight forward modifications of (18) and the proceeding theory applies if representation (3) is computationally more convenient.

Applying Hamilton’s equations (2) to the extended Hamiltonian (18), for $q = (\theta', u')'$ and $p = (p_\theta', p_u')'$, we get the following equations of motion

$$\begin{align*}
\frac{d}{dt} & \begin{pmatrix}
\theta \\
p_\theta \\
u \\
p_u
\end{pmatrix} =
\begin{pmatrix}
M_\theta^{-1}p_\theta \\
\nabla_\theta \log p(\theta) + \nabla_\theta \log \omega_\theta(u) \\
p_u \\
-u + \nabla_u \log \omega_\theta(u)
\end{pmatrix}.
\end{align*}$$

(19)

Equation (19) shows that the Hamiltonian transition dynamics of $(\theta, p_\theta)$ and $(u, p_u)$ are linked together via their joint dependence on the importance weight $\omega_\theta(u)$. However, this link vanishes as the MC variance of the MC estimator $\text{Var}_u[\omega_\theta(u)]$ tend to zero. In fact, an ‘exact’ MC estimate with zero MC variance implies that $\nabla_u \log \omega_\theta(u) = 0_D$, in which case the transition dynamics of $(\theta, p_\theta)$ would be completely decoupled from that of $(u, p_u)$ and would be (marginally) the dynamics of the ‘ideal’ HMC algorithm for $p(\theta|y)$. Moreover, the resulting marginal $(u, p_u)$-dynamics would reduce to that of a harmonic oscillator
with analytical solutions given by $u(t) = \cos(t)u(0) + \sin(t)p_u(0)$ and $p_u(t) = \cos(t)p_u(0) - \sin(t)u(0)$.

In order to approximate the Hamiltonian transition dynamics (19), Lindsten and Doucet (2016) develop a symplectic integrator which for exact likelihood estimates produces exact simulations for the dynamics of $(u, p_u)$ and reduces for $(\theta, p_{\theta})$ to the conventional leapfrog integrator. They derive this integrator for the special case where the mass matrix $M_{\theta}$ in (18) and (19) is restricted to be the identity. For the more general case with an unrestricted $M_{\theta}$ this integrator for approximately advancing the dynamics from time $t = 0$ to time $t = \varepsilon$ is given by

$$\theta(\varepsilon/2) = \theta(0) + (\varepsilon/2)M^{-1}_{\theta}\theta(0), \tag{20}$$

$$u(\varepsilon/2) = \cos(\varepsilon/2)u(0) + \sin(\varepsilon/2)p_u(0), \tag{21}$$

$$p_u^* = \cos(\varepsilon/2)p_u(0) - \sin(\varepsilon/2)u(0), \tag{22}$$

$$p_u^{**} = p_u^* + \varepsilon \nabla_u \{ \log \omega_{\theta(\varepsilon/2)}(u(\varepsilon/2)) \}, \tag{23}$$

$$p_{\theta}(\varepsilon) = p_{\theta}(0) + \varepsilon \nabla_{\theta} \{ \log p[\theta(\varepsilon/2)] + \log \omega_{\theta(\varepsilon/2)}(u(\varepsilon/2)) \}, \tag{24}$$

$$\theta(\varepsilon) = \theta(\varepsilon/2) + (\varepsilon/2)M^{-1}_{\theta}\theta(\varepsilon), \tag{25}$$

$$u(\varepsilon) = \cos(\varepsilon/2)u(\varepsilon/2) + \sin(\varepsilon/2)p_u^{**}, \tag{26}$$

$$p_u(\varepsilon) = \cos(\varepsilon/2)p_u^{**} - \sin(\varepsilon/2)u(\varepsilon/2). \tag{27}$$

## B The EIS principle

In order to minimize the variance of IS estimates for the likelihood $p(y|\theta) = \int p(y|x, \theta)p(x|\theta)dx$ of non-Gaussian and/or nonlinear latent variable models, EIS aims at sequentially constructing an IS density which approximates, as closely as possible, the (infeasible) optimal IS density $m^*(x|\theta) \propto p(y|x, \theta)p(x|\theta)$, which would reduce the variance of likelihood estimates to zero.

With reference to the likelihood it is assumed that the conditional data density $p(y|x, \theta)$ and the prior for the latent variables $p(x|\theta)$ under the latent variable model can be factorized as functions in $x = (x_1, \ldots, x_D)$ into

$$p(y|x, \theta) = \prod_{t=1}^{D} g_t(x_t, \delta), \quad p(x|\theta) = \prod_{t=1}^{D} f_t(x_t|x_{(t-1)}, \delta), \tag{28}$$

where $x_{(t)} = (x_1, \ldots, x_t)$ with $x_{(1)} = x$ and $\delta = (\theta, y)$. Such factorizations can be found for a broad class of models, including dynamic non-Gaussian/nonlinear state-space models for time series, non-Gaussian/nonlinear models with a latent correlation structure for cross-sectional data as well as static hierarchical models without
latent correlation for which \( f_t(x_t|x_{t-1}, \delta) = f_t(x_t, \delta) \). E.g., variants of EIS for univariate and multivariate linear Gaussian states subject to nonlinear measurements are given in Liesenfeld and Richard (2003, 2006) and for more general nonlinear models in Kleppe et al. (2014); Moura and Turatti (2014). EIS implementations with more flexible IS densities such as mixture of normal distributions are found in Kleppe and Liesenfeld (2014), Scharth and Kohn (2016), Grothe et al. (2019), and Liesenfeld and Richard (2010) use truncated normal distributions. Applications of EIS to models with non-Markovian latent variables for spatial data are provided in Liesenfeld et al. (2016, 2017). In our applications we consider univariate time series models, which is why we use \( t \) to index the elements in \( x \) and restrict \( x_t \) in (28) to be one-dimensional.

EIS-MC estimation of likelihood functions \( p(y|\theta) \) associated with (28) is based upon an IS density \( m \) for \( x \) which is decomposed conformably with the factorization in (28) into

\[
m(x|a) = \prod_{t=1}^{D} m_t(x_t|x_{t-1}, a_t),
\]

with conditional densities \( m_t \) such that

\[
m_t(x_t|x_{t-1}, a_t) = \frac{k_t(x_t, a_t)}{\chi_t(x_{t-1}, a_t)} , \quad \chi_t(x_{t-1}, a_t) = \int k_t(x_t, a_t) dx_t,
\]

where \( K = \{ k_t(\cdot, a_t), a_t \in A_t \} \) is a preselected parametric class of density kernels indexed by auxiliary parameters \( a_t \) and with a point-wise computable integrating factor \( \chi_t \). As required for the proposed methodology, it is assumed that the IS density (29) can be simulated by sequentially generating draws from the conditional densities (30) using smooth deterministic functions \( \gamma_t \) such that \( x_t = \gamma_t(a_t, v_t) \) for \( t = 1, \ldots, D \), where \( v_t \sim N(0,1) \).

From (28)-(30) results the following factorized IS representation of the likelihood:

\[
p(y|\theta) = \int \left[ \chi_1(a_1, \delta) \prod_{t=1}^{D} \omega_t(x_t(a_t), a_{t+1}(a_t), \delta) \right] m(x|a) dx,
\]

where the period-\( t \) IS weight is given by

\[
\omega_t(x_t, a_{t+1}, \delta) = \frac{g_t(x_t, \delta) f_t(x_t|x_{t-1}, \delta) \chi_{t+1}(x_t, a_{t+1}, \delta)}{k_t(x_t, a_t)},
\]

with \( \chi_{D+1}(\cdot) \equiv 1 \). For any given \( a = (a_1, \ldots, a_D) \in A = \times_{t=1}^{D} A_t \), the corresponding MC likelihood estimate
is given by

\[ \hat{p}(y|\theta, u) = \omega(x, a), \quad \omega(x, a) = \prod_{t=1}^{D} \omega_t(x_{(t)}, a_{(t+1)}), \]  

(33)

where \( x \) is a draw simulated from the sequential IS density \( m(x|a) \) in (29) (which is obtained by transforming \( u \) using the sequence of smooth deterministic functions \( \gamma_t \)).

In order to minimize the MC variance of the likelihood estimate (33), EIS aims at selecting values for the auxiliary parameters \( a \) that minimize period-by-period the MC variance of the IS weights \( \omega_t \) in (32) with respect to \( m(x|a) \). This requires that the kernels \( k_t(x, a) \) as functions in \( x_{(t)} \) provide the best possible fit to the products \( g_t(x_t, \delta)f_t(x_t|x_{(t-1)}, \delta)\chi_{t+1}(x_{(t)}, a_{(t+1)}) \). For an approximate solution to this minimization problem under the preselected class of kernels \( K \), EIS solves the following back-recursive sequence of least squares (LS) approximation problems:

\[
\begin{align*}
(\hat{c}_t, \hat{a}_t) = \arg \min_{c_t, a_t \in \mathcal{A}} \sum_{i=1}^{r} \left\{ \log \left[ g_t(x^{(i)}_t, \delta) f_t(x^{(i)}_t|x^{(i)}_{(t-1)}, \delta) \chi_{t+1}(x^{(i)}_{(t)}, \hat{a}_{(t+1)}) \right] \right. \\
&- c_t - \log k_t(x^{(i)}_{(t)}, a_t) \right\}^2, \quad t = D, D - 1, \ldots, 1,
\end{align*}
\]

(34)

where \( c_t \) represents an intercept, and \( \{x^{(i)}_t\}_{i=1}^{r} \) denote \( r \) iid draws simulated from \( m(x|a) \) itself. Thus, the EIS-optimal values for the auxiliary parameters \( \hat{a} \) result as a fixed-point solution to the sequence \( \{\hat{a}^{[0]}, \hat{a}^{[1]}, \ldots\} \) in which \( \hat{a}^{[j]} \) is given by (34) under draws from \( m(x|\hat{a}^{[j-1]}) \). In order to ensure convergence to a fixed-point solution it is critical that all the \( x \) draws simulated for the sequence \( \{\hat{a}^{[j]}\} \) be generated by using the smooth deterministic functions \( \gamma_t \) to transform a single set of \( rD \) Common Random Numbers (CRNs), say \( z \sim N(0, rD, I_{rD}) \). To initialize the fixed-point iterations \( j = 0, \ldots, J \), the starting value \( \hat{a}^{[0]} \) can be found, e.g., from an analytical local approximation (such as Laplace) of the EIS targets \( \ln(g_t f_t \chi_{t+1}) \) in (34).

Convergence of the iterations to a fixed-point solution is typically fast to the effect that a value for the number of iterations \( J \) between 2 and 4 often suffices to produce a (close to) optimal solution (Richard and Zhang 2007). The MC-EIS likelihood estimate, for a given \( \theta \), is then calculated by substituting in (33) the EIS-optimal value \( \hat{a} \) for \( a \). In order to highlight its dependence on \( \theta \) and \( z \) we shall use \( \hat{a} = a(\theta, z) \) to denote the EIS-optimal value.

The selection of the parametric class \( K \) of EIS density kernels \( k_t \) is inherently specific to the latent variable model under consideration as those kernels are meant to provide a functional approximation in \( x_{(t)} \) to the product \( g_t f_t \chi_{t+1} \). In the applications below, we consider models with data densities \( g_t \) which are log-concave in \( x_t \) and Gaussian conditional densities for \( x_t \) with a Markovian structure so that \( f_t(x_t|x_{(t-1)}, \delta) = \)
This suggests selection of the $k_t$’s as Gaussian kernels and to exploit that such kernels are closed under multiplication in order to construct the $k_t$’s as the following parametric extensions of the prior densities $f_t$:

$$k_t(x_t, x_{t-1}, \mathbf{a}_t) = f_t(x_t|x_{t-1}, \mathbf{a}_t) \xi_t(x_t, \mathbf{a}_t),$$

(35)

where $\xi_t$ is a Gaussian kernel in $x_t$ of the form $\xi_t(x_t, \mathbf{a}_t) = \exp\{a_1 t x_t + a_2 x_t^2\}$ with $\mathbf{a}_t = (a_{1t}, a_{2t})$. In this case the EIS approximation problems (34) take the form of simple linear LS-problems where $\log[\hat{g}_t(x_t^{(i)}, \delta)]\chi_{t+1}(x_t^{(i)}, \hat{a}_{t+1})$ are regressed on a constant, $x_t^{(i)}$ and $[x_t^{(i)}]^2$. In fact, (34) reduces to linear LS regressions for all kernels $k_t$ chosen within the exponential family (Richard and Zhang, 2007), which simplifies implementation. However, it is important to note that EIS is by no means restricted to the use of IS densities from the exponential family nor to models with low-order Markovian specifications for the latent variables.

The EIS approach as outlined above differs from standard IS in that it uses IS densities whose parameters $\hat{\mathbf{a}} = \mathbf{a}(\theta, \mathbf{z})$ are (conditional on $\theta$) random variables as they depend via the EIS fixed-point regressions (34) on the CRNs $\mathbf{z}$. This calls for specific rules for implementing EIS which ensure that the resulting MC likelihood estimates meet the qualifications needed for their use within PM-HMC. In order to ensure that the EIS likelihood estimate (33) based on the random numbers $\mathbf{u}$ is unbiased the latter need to be a set of random draws different from the CRNs $\mathbf{z}$ used to find $\hat{\mathbf{a}}$ (Kleppe and Liesenfeld, 2014). Note also that since $\hat{\mathbf{a}}$ is an implicit function of $\theta$, maximal accuracy requires us to rerun the EIS fixed-point regressions for any new value of $\theta$. In order to ensure that the resulting EIS likelihood estimate (33) as a function of $\hat{\mathbf{a}}$ is smooth in $\theta$, $\hat{\mathbf{a}}$ itself needs to be a smooth function of $\theta$. This can be achieved by presetting the number of fixed-point iterations $J$ across all $\theta$-values to a fixed number, rather than using a stopping rule based on a relative-change threshold.

The EIS-specific tuning parameters are the number of $\mathbf{x}^{(i)}$-draws $r$ used to run the EIS optimization process, the number of fixed-point iterations on the EIS regressions $J$, and the number of $\mathbf{x}^{(i)}$-draws $n$ for the likelihood estimate (33). Those parameters should be selected to balance the trade-off between EIS computing time and the quality of the resulting EIS density with respect to the MC accuracy. In particular, for $r$ it is recommended to select it as small as possible while retaining the EIS fixed-point regressions numerically stable and the parameter $J$ should be set such that it is guaranteed that the fixed-point sequence $\{\mathbf{a}^{[j]}\}_j$ approximately converge for the $\theta$ values in the relevant range of the parameter space. In our applications, where the selected class of kernels $\mathcal{K}$ imply that the EIS regressions are linear in the EIS parameters $\mathbf{a}_t$, we find that a $J$ set equal to 1 or 2 and an $r$ about 2 times the number of parameters in $(\mathbf{a}_t, \mathbf{c}_t)$ suffice. We obtain EIS kernels $k_t$ providing highly accurate approximations to the targeted product...
\( g_t \chi_{t+1} \), with an \( R^2 \) of the EIS regressions in the final iteration typically larger than 0.95.

## C Details related to the example models in Section 5

### C.1 SV model

For the SV model, the standard prior assumptions for the parameters \( \theta = (\gamma, \delta, \nu) \) are the following: for \( \gamma \) we use a flat prior, for \((\delta + 1)/2\) a Beta \( B(\alpha, \beta) \) with \( \alpha = 20 \) and \( \beta = 1.5 \), and for \( \nu^2 \) a scaled inverted-\( \chi^2 \) prior \( p_0 s_0 / \chi^2(p_0) \) with \( p_0 = 10 \) and \( s_0 = 0.01 \). For numerical stability we use the parametrization \( \theta^* = (\gamma, \text{arctanh} \delta, \log \nu^2) \) together with the priors for \( \theta^* \) to run the HMC algorithms, where the priors are derived from those on \( \theta \).

For the Laplace transport map, \( G^{(0)}_\theta \) and \( h^{(0)}_\theta \) are taken to be identical to \((6,7)\). More refined solutions are found using Newton iterations;

\[
\begin{align*}
    h^{(k)}_\theta &= h^{(k-1)}_\theta + \left[ \nabla_x^2 \log [p(x|\theta)p(y|x, \theta)]_{x=h^{(k-1)}_\theta} \right]^{-1} \left\{ \nabla_x \log [p(x|\theta)p(y|x, \theta)]_{x=h^{(k-1)}_\theta} \right\}, \\
    G^{(k)}_\theta &= \nabla_x^2 \log [p(x|\theta)p(y|x, \theta)]_{x=h^{(k-1)}_\theta}.
\end{align*}
\]

for \( k = 1, 2, \ldots, K \). Further modifications, including changing to \( G^{(k)}_\theta = \nabla_x^2 \log [p(x|\theta)p(y|x, \theta)]_{x=h^{(k)}_\theta} \) (at the cost of one additional Cholesky factorization), or keeping \( G^{(k)}_\theta = G^{(0)}_\theta \) (costs only a single Cholesky factorization) both in the transport map and as the scaling matrix in the Newton iterations was tried, but did not produce better results.

It is straightforward to show that \( G_{\theta,y|x} = 0.5 I_D \) is also the Fisher information of \( p(y|x) \) with respect to \( x \) (i.e. \( p(y|x) \) is a constant information parameterization). Hence also Stan-Laplace \( K = 0 \) may be interpreted as a special case of DRHMC \((\text{Kleppe} \ 2019)\).  

### C.2 Gamma model

For the Gamma model, the priors on the parameters \( \theta = (\tau, \beta, \delta, \nu) \) are as follows; we use flat priors for \( \log \tau \) as well as \( \log \beta \), a Beta \( B(\alpha, \beta) \) with \( \alpha = 20 \) and \( \beta = 1.5 \) for \((\delta + 1)/2\), and a scaled inverted-\( \chi^2 \) for \( \nu^2 \) with \( p_0 s_0 / \chi^2(p_0) \) and \( p_0 = 10, s_0 = 0.01 \). For the LD computations we use the parameterization \( \theta^* = (\log \tau, \log \beta, \text{arctanh} \delta, \log \nu^2) \).

For this model, the same strategy for calculating the Laplace transport map as for the SV model was used. Notice that here \( G_{\theta,y|x} = \tau^{-1} I_D \) is also the Fisher information of \( p(y|x) \) with respect to \( x \). Hence, Stan-Laplace, \( K = 0 \) may be interpreted as a DRHMC method.
In the Stan implementation, KL Laplace,

Table 5: Posterior mean and standard deviations for the inverse Wishart model (14-16) based on Stan-

For the CEV model, for C.3 CEV model

and should be close to standard normal when the transport map produces a sufficient de-coupling effect.

The LD computations are conducted on the following transformed parameters:

for \(k\)

Further modifications, including changing to \(G^{(k)}_\theta = \nabla^2_x \log [p(x|\theta)p(y|x, \theta)]_{x=h_\theta^{(k-1)}}\) (at the cost of one additional Cholesky factorization) did not improve the fit sufficiently to warrant the additional computation.

D Details related to the realized volatility model in Section \[6\]

The (normalized) observation density is given by:

In the Stan implementation, \(\prod_{t=1}^{D} |Y_t|\) and \(Y_t^{-1}\), \(t = 1, \ldots, D\) where precomputed.
Figure 2: Posterior mean and standard deviation of $u_s$, $s = 1, \ldots, 5$, for the inverse Wishart model (14-16) under Laplace transport map with $K = 2$.

Figure 3: Posterior mean and standard deviation of $u_s$, $s = 1, \ldots, 5$, for the inverse Wishart model (14-16) under Laplace transport with $K = 10$. 
The (independent) priors used to complete the model specification in Section 6.1 are as follows: \( \mu_s \sim N(0, 25) \), \( \delta_s \sim \text{uniform}(-1, 1) \), \( \sigma_s^2 \sim p_0 s_0 / \chi_p^2 \) where \( p_0 = 4 \) and \( s_0 = 0.25 \), \( h_{i,j} \sim N(0, 100) \). Finally, a flat prior on \((6.0, \infty)\) was chosen for \( \nu \).

Posterior means and standard deviations of the parameters and the first elements in \( x_s \) and \( u_s \) are given in Table 5. The results are very much in line with those of Grothe et al. (2019).

The Laplace transport maps for each of \( x_s \), \( s = 1, \ldots, r \) are constructed as follows; the initial guesses for \( h_\theta^{(0)} \) and \( G_\theta^{(0)} \) are those given in (6,7), applied to (17). The mean is further refined via the following approximate Newton iteration

\[
h_\theta^{(k)} = h_\theta^{(k-1)} + \left[ G_\theta^{(0)} \right]^{-1} \left\{ \nabla_x \log \left[ p(x|\theta)p(y|x, \theta) \right]_{x=h_\theta^{(k-1)}} \right\},
\]

whereas \( G_\theta^{(k)} = G_\theta^{(0)} \) is kept fixed which result in that only a single Cholesky factorization is required. Figures 2,3 show the posterior mean and standard deviations of \( u_s \) over time \( t \) for Stan-Laplace, \( K = 2 \) and \( K = 10 \) respectively. It is seen that even with the approximate Newton iteration, the iteration makes \( u_s \) have a mean close to zero, where the remaining deviation from zero for \( K = 10 \) iterations in Figure 3 is presumably due to the non-quadratic nature of the "measurement density" in (17) (in addition to Monte Carlo variation).

References

Grothe, O., T. S. Kleppe, and R. Liesenfeld (2019). The Gibbs sampler with particle efficient importance sampling for state-space models. *Econometric Reviews* 38(10), 1152–1175.

Kleppe, T. S. (2019). Dynamically rescaled Hamiltonian Monte Carlo for Bayesian hierarchical models. *Journal of Computational and Graphical Statistics* 28(3), 493–507.

Kleppe, T. S. and R. Liesenfeld (2014). Efficient importance sampling in mixture frameworks. *Computational Statistics & Data Analysis* 76, 449 – 463.

Kleppe, T. S., J. Yu, and H. J. Skaug (2014). Maximum likelihood estimation of partially observed diffusion models. *Journal of Econometrics* 180(1), 73 – 80.

Liesenfeld, R. and J.-F. Richard (2003). Univariate and multivariate stochastic volatility models: estimation and diagnostics. *Journal of Empirical Finance* 10(4), 505–531.

Liesenfeld, R. and J.-F. Richard (2006). Classical and Bayesian analysis of univariate and multivariate stochastic volatility models. *Econometric Reviews* 25(2-3), 335–360.
Liesenfeld, R. and J.-F. Richard (2010). Efficient estimation of probit models with correlated errors. *Journal of Econometrics* 156(2), 367–376.

Liesenfeld, R., J.-F. Richard, and J. Vogler (2016). Likelihood evaluation of high-dimensional spatial latent gaussian models with non-gaussian response variables. In *Spatial Econometrics: Qualitative and Limited Dependent Variables*, pp. 35–77. Emerald Group Publishing Limited.

Liesenfeld, R., J.-F. Richard, and J. Vogler (2017). Likelihood-based inference and prediction in spatio-temporal panel count models for urban crimes. *Journal of Applied Econometrics* 32(3), 600–620.

Lindsten, F. and A. Doucet (2016). Pseudo-Marginal Hamiltonian Monte Carlo. arXiv preprint arXiv:1607.02516.

Moura, G. V. and D. E. Turatti (2014). Efficient estimation of conditionally linear and Gaussian state space models. *Economics Letters* 124(3), 494 – 499.

Richard, J.-F. and W. Zhang (2007). Efficient high-dimensional importance sampling. *Journal of Econometrics* 141(2), 1385–1411.

Scharth, M. and R. Kohn (2016). Particle efficient importance sampling. *Journal of Econometrics* 190(1), 133 – 147.