Geophysical Journal International

Geophys. J. Int. (2022) **228,** 213–239
Advance Access publication 2021 July 30
GJI Seismology

https://doi.org/10.1093/gji/ggab298

Bayesian seismic tomography using normalizing flows

Xuebin Zhao, Andrew Curtis and Xin Zhang

School of GeoSciences, University of Edinburgh, Edinburgh, United Kingdom E-mail: xuebin.zhao@ed.ac.uk

Accepted 2021 July 27. Received 2021 July 22; in original form 2020 December 30

SUMMARY

We test a fully non-linear method to solve Bayesian seismic tomographic problems using data consisting of observed traveltimes of first-arriving waves. Rather than using Monte Carlo methods to sample the posterior probability distribution that embodies the solution of the tomographic inverse problem, we use variational inference. Variational methods solve the Bayesian inference problem under an optimization framework by seeking the best approximation to the posterior distribution from a family of distributions, while still providing fully probabilistic results. We introduce a new variational method for geophysics—normalizing flows. The method models the posterior distribution by using a series of invertible and differentiable transforms—the flows. By optimizing the parameters of these transforms the flows are designed to convert a simple and analytically known probability distribution into a good approximation of the posterior distribution. Numerical examples show that normalizing flows can provide an accurate tomographic result including full uncertainty information while significantly decreasing the computational cost compared to Monte Carlo and other variational methods. In addition, this method provides analytic solutions for the posterior distribution rather than an ensemble of posterior samples. This opens the possibility that subsequent calculations that use the posterior distribution might be performed analytically.

Key words: Seismic tomography; Image processing; Inverse theory; Probability distributions.

1 INTRODUCTION

Seismic traveltime tomography is commonly performed to image the Earth's interior structures and to infer subsurface properties. It can be formulated as an inverse problem that estimates parameters of interest (typically underground seismic velocity maps) given observed data (Curtis & Snieder 2002). Traveltime tomography has been successfully applied to many geophysical problems at different scales: for example at local scale (Aki *et al.* 1977; Thurber 1983; Mordret *et al.* 2014), regional scale (Spakman 1991; Curtis *et al.* 1998; Gorbatov *et al.* 2000; Simons *et al.* 2002) and global scale (Dziewonski & Woodhouse 1987; Inoue *et al.* 1990; Trampert & Woodhouse 1995; Shapiro & Ritzwoller 2002; Meier *et al.* 2007a, b). The method is also used to produce underground images in industrial geophysics (de Ridder & Dellinger 2011; Mordret *et al.* 2013; de Ridder *et al.* 2014; Allmark *et al.* 2018).

In each case we invert for the underground velocity maps using traveltimes between source and receiver pairs. Traveltime data can usually be obtained by picking times of seismic wave arrivals from seismograms obtained in one of the following three ways: recordings of earthquakes, recordings of active sources and—as will be used herein—from ambient noise using seismic interferometry (Campillo & Paul 2003; Curtis *et al.* 2006; Wapenaar *et al.* 2010a, b; Galetti & Curtis 2012). Ambient noise tomography using traveltime data estimated from seismic interferometry received much attention in the past two decades (Shapiro *et al.* 2005; Sabra *et al.* 2005; Villasenor *et al.* 2007; Rawlinson *et al.* 2008; Zheng *et al.* 2010; Nicolson *et al.* 2012, 2014; Galetti *et al.* 2017), because earthquakes are distributed irregularly in space and time, and some regions have low levels of seismicity so could not be imaged in any detail using earthquake-based methods. Recordings of ambient noise at pairs of receivers may be converted to seismograms which emulate those that would have been recorded if a source was fired at the location of one of the receivers and was recorded by the other. The imagined source is referred to as a virtual source. This means that real receivers can be used as virtual sources (and, in fact, vice versa—Curtis *et al.* 2009) so receiver arrays may be designed such that they act both as receivers and sources. Techniques to convert the ambient noise into virtual source seismograms vary around a standard theme described in Bensen *et al.* (2008), and in this study we used the traveltime data obtained by Galetti *et al.* (2015, 2017).

Seismic tomographic methods can be divided into two categories. In the first, the non-linear model-data relationship (the *forward* function) is linearized and hence approximated. Given a reference model, the locally best-fitting solution is then found by iteratively minimizing a predefined misfit function between the observed data and data simulated from an Earth velocity model (Iyer & Hirahara 1993; Rawlinson

et al. 2010). Although these methods are often computationally efficient, in significantly non-linear problems they require a good initial model estimate to avoid finding local minima. In addition, since tomographic problems are often ill-conditioned, additional regularization terms are introduced to stabilize the solutions (Tarantola 2005; Loris *et al.* 2007). However, regularization often creates a biased result that suppresses important information in the data (Zhdanov 2002). What is more, in a linearized framework it is hard to estimate uncertainty in the tomographic results as illustrated in Galetti *et al.* (2015), which is especially important for risk evaluation and decision making (Arnold & Curtis 2018).

The second category—referred to as fully non-linear tomography—has drawn much attention recently. It can provide probabilistic solutions by evaluating the *posterior* probability density function (pdf) that describes the full, non-unique solution to the tomographic problem as defined by Bayes' theorem without linearization and without additional regularization—referred to as Bayesian inference. Bayesian inference updates the *prior* probability distribution (the information about parameters known before inversion) with information in the observed data. It provides fully probabilistic results describing all information on the parameters conditioned on the data—the posterior pdf.

Markov chain Monte Carlo (McMC) is frequently used to solve Bayesian inference problems (Metropolis *et al.* 1953; Hastings 1970; Geyer & Thompson 1995; Neal 2011; Hoffman & Gelman 2014). This method generates an ensemble of correlated samples, which are distributed according to a desired pdf, usually the Bayesian posterior pdf, as the number of samples tends to infinity. We can use any finite set of such samples to approximate statistical properties such as mean, standard deviation and marginal distributions of the posterior pdf. In geophysics, McMC methods have long been applied to solve inverse problems (Press 1968; Anderssen & Seneta 1971; Mosegaard & Tarantola 1995; Sambridge 1999; Malinverno 2002). In the more sophisticated Reversible Jump McMC (rj-McMC—Green 1995, 2003; Green & Mira 2001), the parametrization including dimensionality of the parameter vector is also treated as unknown and is constrained by the data during inversion (Bodin & Sambridge 2009; Bodin *et al.* 2012; Galetti *et al.* 2015, 2017; Galetti & Curtis 2018; Zhang *et al.* 2018, 2020). This can lead to huge gains in efficiency by reducing dimensionality to only parameters that are justifiably necessary to explain the data. Since the 'curse of dimensionality' is the major source of computational cost in sampling pdfs (Curtis & Lomax 2001), rj-McMC has been treated as one of the favourite methods for tomography over the past decade. Nevertheless, one deficiency of McMC based methods is that they are slow to converge to the true posterior distribution. They also may not converge in finite time, and detecting the state of convergence is difficult in practice. Recently, Hamiltonian Monte Carlo (HMC) has been recognized as a potential approach to solve geophysical inversion problems (Muir & Tkalcic 2015; Sen & Biswas 2017; Fichtner & Simutė 2018; Fichtner *et al.* 2019; Gebraad *et al.* 2020). This method uses the derivatives of data with respect to model parameters to speed up the sampling process. The computational cost of HMC grows with the dimensionality n as $O(n^{5/4})$ (Neal 2011), while it is $O(n^2)$ for Metropolis–Hastings McMC (MH-McMC). Walker & Curtis (2014) proposed a recursive algorithm for exact posterior sampling (meaning that every sample is exactly a sample of the posterior pdf) to solve Bayesian inversion in spatial or gridded models with localized data, such that the convergence issue of McMC is avoided entirely. Even so, for high dimensional 2-D or 3-D problems, all of these sampling algorithms are expensive due to the curse of dimensionality.

An alternative class of methods use neural networks (NNs) to solve inverse problems. In principle, NNs can represent (learn) any complex function between input and output vectors (Bishop 2006). The non-linear map from data space to parameter space is learned using optimization methods to adjust the NN parameters such that the NN optimally emulates the inverse data-model mapping represented by a large training data set generated by forward simulation of model samples. Thereafter any data set can be mapped to corresponding parameter values using that learned mapping. NNs have been applied successfully to many geophysical inverse problems, either to find a single deterministic solution that fits the observed data (Röth & Tarantola 1994; Moya & Irikura 2010; Araya-Polo *et al.* 2018; Bianco & Gerstoft 2018; Kong *et al.* 2019), or to find a fully probabilistic result that represents the posterior pdf (Devilee *et al.* 1999; Meier *et al.* 2007a, b; Shahraeeni & Curtis 2011; Shahraeeni *et al.* 2012; de Wit *et al.* 2013; Käufl *et al.* 2014, 2015; Earp & Curtis 2020; Earp *et al.* 2020). The merit of these methods is their efficiency when inverting different data sets: once the NN has been properly trained, the inversion process can be accomplished rapidly (usually in seconds) by feeding each new observed data set into the NN. This contrasts with McMC methods which must execute the whole sampling process for every new data set. However, training a representative and robust NN to emulate complicated data-model relationships is difficult, and generating sufficient training data that spans the whole prior space can also be prohibitively expensive due to the curse of dimensionality (by contrast, Monte Carlo methods only sample the posterior pdf which is usually far more compact than the prior pdf).

Considering the aforementioned deficiencies of McMC and NN based methods, in this paper we focus on variational inference to solve tomographic problems. Variational methods have long been recognized as an alternative strategy to McMC for modelling the posterior pdf in the machine learning community due to their computational efficiency and scalability to large data sets (Bishop 2006; Blei *et al.* 2017). The basic idea is to approximate the posterior distribution by a simpler distribution q (called the variational distribution) that lies within a predefined variational family $\mathcal{Q}$. To this end, we try to find a member in this family that minimizes the difference between the posterior and the variational distributions, for example by minimizing the Kullback–Leibler (KL) divergence (Kullback & Leibler 1951) or kernelized stein discrepancy (Liu *et al.* 2016) between the two distributions. The resulting best-fitting distribution q^* is the solution of the variational problem. Thus, similarly to NN based inversions, variational inference converts the usual sampling problem into an optimization problem, while still providing a fully probabilistic result. However, in contrast to the NN solution, variational methods usually only approximate the posterior pdf for a specific data set rather than over the entire prior pdf, which for a low number of data sets should be more accurate and efficient. What is more, this approach circumvents computation of the evidence term (the normalization constant) in Bayes' rule, which is often intractable in high-dimensional inference problems. For an introduction to variational methods, see Zhang *et al.* (2021).

Variational inference has previously been studied in many different fields such as computational biology (Carbonetto *et al.* 2012), computational neuroscience (Roberts & Penny 2002) and computer vision (Likas & Galatsanos 2004). In geophysics, Nawaz & Curtis (2018, 2019) first used variational methods to make inference on the spatial distribution of geological facies from attributes of seismic data, respectively using the expectation maximization algorithm and the mean field approximation. Recent extensions in Nawaz *et al.* (2020) inverted seismic attributes jointly for petrophysical rock properties and geological facies. Zhang & Curtis (2020a) introduced two variational inference algorithms to solve traveltime tomographic problems, namely automatic differential variational inference (ADVI—Kucukelbir *et al.* 2017) and Stein variational gradient descent (SVGD—Liu & Wang 2016), and the latter algorithm was also used to solve fully probabilistic full waveform inversion (FWI, Zhang & Curtis 2020b). ADVI restricts the variational distribution to lie within a Gaussian family; this is efficient for problems with Gaussian-like posterior pdfs, but provides poor approximations for complicated multimodal distributions (Zhang & Curtis 2020a). SVDG is a sample based method that iteratively perturbs a set of samples from the prior pdf to represent samples of the posterior distribution using optimization. The latter method avoids the problem of detecting statistical convergence that pervades McMC methods, but still suffers from the curse of dimensionality in the number of samples required to represent the posterior pdf.

In this paper, we introduce another variational inference method that is new to geophysics: normalizing flows (Rezende & Mohamed 2015). Normalizing flows are a set of invertible, differentiable and parametrized transforms that convert a simple and analytically known distribution (the initial distribution), for example a standard normal or Uniform distribution, into an approximation of any complex pdf (Dinh *et al.* 2015; Rezende & Mohamed 2015; Kobyzev *et al.* 2019; Papamakarios *et al.* 2021). Since both the initial distribution and the flows are analytically known, the resulting posterior distribution is also analytic. We show that flows have the potential to provide a step-change reduction in computation for probabilistic non-linear tomographic problems.

The rest of this paper is organized as follows. We start by introducing the variational method for Bayesian inversion, followed by a brief review of normalizing flows which includes their basic principles, two commonly used applications in the literature, and ways to construct normalizing flows. In the third section, two examples are implemented to prove the effectiveness and efficiency of normalizing flows for Bayesian inference. The first one is a synthetic traveltime tomography test and the second one is a field data test for Love wave tomography of the British Isles using traveltimes derived from ambient noise interferometry. Finally, we provide a brief discussion about the implications of this work and draw conclusions.

2 METHODOLOGY

2.1 Variational Bayesian inference

In a Bayesian framework, we solve inverse problems using probabilities to represent our state of knowledge about the unknown model parameters by invoking Bayes' rule:

$$p(\mathbf{m}|\mathbf{d}_{obs}) = \frac{p(\mathbf{m}, \mathbf{d}_{obs})}{p(\mathbf{d}_{obs})} = \frac{p(\mathbf{d}_{obs}|\mathbf{m})p(\mathbf{m})}{p(\mathbf{d}_{obs})}, \tag{1}$$

where for traveltime tomography problems, $\mathbf{m}$ is the vector of seismic velocities across the subsurface model, and $\mathbf{d}_{obs}$ is the vector of observed traveltime data. Distribution $p(\mathbf{m})$ is the *prior* pdf of the model parameters $\mathbf{m}$, which describes the information about $\mathbf{m}$ known before inversion of the current data $\mathbf{d}_{obs}$. The *likelihood* term $p(\mathbf{d}_{obs}|\mathbf{m})$ is the conditional probability of observing the data $\mathbf{d}_{obs}$ given a particular model $\mathbf{m}$, which is used to quantify how likely it is that model $\mathbf{m}$ could generate the observed data using a given forward function. The prior $p(\mathbf{m})$ and likelihood $p(\mathbf{d}_{obs}|\mathbf{m})$ together specify the *joint* probability distribution over parameters and data $p(\mathbf{m}, \mathbf{d}_{obs})$. The denominator $p(\mathbf{d}_{obs}) = \int_{\mathbf{m}} p(\mathbf{d}_{obs}|\mathbf{m})p(\mathbf{m})d\mathbf{m}$ is called *evidence* and acts as a normalization constant in Bayesian inference. Combining the three terms on the right-hand side of eq. (1) gives the *posterior* pdf $p(\mathbf{m}|\mathbf{d}_{obs})$ which represents how probable is model $\mathbf{m}$ given all information in the prior pdf and the current data.

We often assume the likelihood function to be a Gaussian distribution that represents uncertainties on the observed data $\mathbf{d}_{obs}$:

$$p(\mathbf{d}_{obs}|\mathbf{m}) \propto \exp\left[-\frac{\left(\mathbf{d}_{syn} - \mathbf{d}_{obs}\right)^T \Sigma_{\mathbf{d}}^{-1} \left(\mathbf{d}_{syn} - \mathbf{d}_{obs}\right)}{2}\right], \tag{2}$$

where $\mathbf{d}_{syn}$ is the synthetic data predicted by the forward function given a seismic velocity model, and $\Sigma_{\mathbf{d}}$ is the data covariance matrix. We often use a diagonal covariance matrix with $\sigma_{\mathbf{d}}^2$ on diagonal entries, where $\sigma_{\mathbf{d}}$ is the uncertainty of the traveltime data. The numerator in the square bracket is the (negative of the) least-squares error between observed and synthetic data.

Directly calculating the right-hand side of eq. (1) is computationally intractable since the evidence term $p(\mathbf{d}_{obs})$ implicitly contains a high-dimensional integration over the whole prior space. Sampling based methods such as McMC are popular despite their computational cost due to the curse of dimensionality, because they provide an ensemble of samples of the posterior distribution without calculating $p(\mathbf{d}_{obs})$ explicitly.

In contrast to sampling based methods, variational inference approximates the posterior distribution by a simpler one $q(\mathbf{m})$ (the variational distribution) defined in the family of variational pdfs $\mathcal{Q}(\mathbf{m}) = \{q(\mathbf{m})\}$. Dividing eq. (1) by $q(\mathbf{m})$ and taking the logarithm on both sides, it

becomes

$$\log p(\mathbf{d}_{obs}) = \log p(\mathbf{m}, \mathbf{d}_{obs}) - \log q(\mathbf{m}) + \log \frac{q(\mathbf{m})}{p(\mathbf{m}|\mathbf{d}_{obs})}. \tag{3}$$

Calculating the expectation with respect to $q(\mathbf{m})$ on both sides of eq. (3) then gives

$$\begin{aligned} \log p(\mathbf{d}_{obs}) &= \mathbb{E}_{q(\mathbf{m})}[\log p(\mathbf{m}, \mathbf{d}_{obs})] - \mathbb{E}_{q(\mathbf{m})}[\log q(\mathbf{m})] + \mathrm{KL}[q(\mathbf{m})||p(\mathbf{m}|\mathbf{d}_{obs})] \\ &\geq \mathbb{E}_{q(\mathbf{m})}[\log p(\mathbf{m}, \mathbf{d}_{obs})] - \mathbb{E}_{q(\mathbf{m})}[\log q(\mathbf{m})] \triangleq \mathcal{L}[q(\mathbf{m})] \end{aligned}. \tag{4}$$

The reason we apply the above mathematical manipulations is to explicitly obtain $\mathrm{KL}[q(\mathbf{m})||p(\mathbf{m}|\mathbf{d}_{obs})]$ in eq. (4). This term is the Kullback–Leibler (KL) divergence (Kullback & Leibler 1951) defined as $\mathrm{KL}[q(\mathbf{m})||p(\mathbf{m}|\mathbf{d}_{obs})] = \mathbb{E}_{q(\mathbf{m})}[\log \frac{q(\mathbf{m})}{p(\mathbf{m}|\mathbf{d}_{obs})}]$, and measures the difference (distance) between the two distributions. It has the property $\mathrm{KL}[q(\mathbf{m})||p(\mathbf{m}|\mathbf{d}_{obs})] \geq 0$ and equality holds only when $q(\mathbf{m}) = p(\mathbf{m}|\mathbf{d}_{obs})$. Thus it gives the second line of eq. (4), where symbol $\triangleq$ acts as the definition of $\mathcal{L}[q(\mathbf{m})]$—the so-called evidence lower bound of $\log p(\mathbf{d}_{obs})$.

By minimizing KL divergence within the variational family, the resulting optimal distribution $q^*(\mathbf{m})$ is by definition the one closest to $p(\mathbf{m}|\mathbf{d}_{obs})$, and thus serves as the optimal approximation to the posterior distribution. In eq. (4), this is equivalent to maximizing $\mathcal{L}[q(\mathbf{m})]$ because $\log p(\mathbf{d}_{obs})$ stays fixed for different $q(\mathbf{m})$. Thus an intractable, high dimensional sampling problem is converted into a numerical optimization problem, while still providing fully probabilistic results.

There is a trade-off when choosing the variational family: it needs to be sufficiently expressive to provide an accurate approximation, yet simple enough for efficient optimization. The mean field approximation has been invoked in previous work to solve variational problems; this assumes a diagonal covariance matrix for the model vector $\mathbf{m}$ so that no correlations between different parameters are considered (Bishop 2006; Blei *et al.* 2017; Nawaz & Curtis 2018, 2019). However, this restriction has a significant impact on the accuracy of the approximated posterior. In the following sections, we introduce a recently proposed variational method: *normalizing flows* (Dinh *et al.* 2015; Rezende & Mohamed 2015) to solve tomographic problems without invoking the mean field.

2.2 Introduction to normalizing flows

2.2.1 Fundamentals

Normalizing flows (Dinh *et al.* 2015; Rezende & Mohamed 2015) provides a flexible way to construct a probability density by passing one distribution through a series of invertible and differentiable transforms, called the flows. By repeatedly applying the rule for change of variables, a simple distribution (often a normal or Uniform distribution) 'flows' through the sequence of invertible mappings. Since the flows are designed to be expressive, we could transform the initial distribution into an approximation of any desired distribution. It can therefore be optimized to provide a better approximation to the posterior pdf than the initial distribution.

Let $\mathbf{m}_0 \in \mathbb{R}^D$ be a D-dimensional random variable whose probabilistic distribution $q_0(\mathbf{m}_0)$ is simple and analytically known (for example a Gaussian or Uniform distribution), and apply a differentiable and invertible function f_θ (parametrized by θ), such that $\mathbf{m}_1 = f_\theta(\mathbf{m}_0)$: $\mathbb{R}^D \to \mathbb{R}^D$. Based on the change of variable rule, the pdf of the transformed vector $\mathbf{m}_1$ can be calculated by

$$q_1(\mathbf{m}_1) = q_0(\mathbf{m}_0)\left|\det \frac{\partial f_\theta^{-1}}{\partial \mathbf{m}_1}\right| = q_0(\mathbf{m}_0)\left|\det \frac{\partial f_\theta}{\partial \mathbf{m}_0}\right|^{-1}, \tag{5}$$

where $|\cdot|$ calculates the absolute value, and $\det(\cdot)$ evaluates the determinant of a matrix. The absolute value of the Jacobian determinant denotes the volume change corresponding to this transform. Under this scenario, $q_0(\mathbf{m}_0)$ is called the initial (base) distribution and f_θ is a normalizing flow: it pushes the simpler and known initial distribution into the target distribution $q_1(\mathbf{m}_1)$ that we desire. Depending on the flow function f_θ, the initial distribution can be manipulated in different ways, for example it can be expanded, contracted, rotated or its location can be shifted, to approximate the target distribution.

The expressiveness of normalizing flows is predominantly controlled by the complexity and expressiveness of the flow function f_θ. Theoretically speaking, one could generate any form of target distribution from any known initial one using well-defined transforms (Papamakarios *et al.* 2021). We therefore need methods to design effective transforms for the target distribution of interest. Directly constructing the target distribution with one discrete transform (eq. 5) would be difficult since it is relatively hard to design a specific invertible transform that can transform a given distribution into any form of interest. Instead, it can be accomplished by combining multiple simple mappings and successively applying eq. (5), given that the composition of a series of invertible and differentiable functions is itself invertible and differentiable. Specifically, suppose we have K invertible and differentiable functions applied to $q_0(\mathbf{m}_0)$. Together they output:

$$\begin{cases} \mathbf{m}_K = f_{\theta_K} \circ f_{\theta_{K-1}} \cdots \circ f_{\theta_2} \circ f_{\theta_1}(\mathbf{m}_0) \\ q_K(\mathbf{m}_K) = q_0(\mathbf{m}_0) \prod_{i=1}^K \left|\det \frac{\partial f_{\theta_i}^{-1}}{\partial \mathbf{m}_i}\right| = q_0(\mathbf{m}_0) \prod_{i=1}^K \left|\det \frac{\partial f_{\theta_i}}{\partial \mathbf{m}_{i-1}}\right|^{-1} \end{cases}, \tag{6}$$

where the volume change is controlled by the absolute value of the Jacobian determinant of each transform $\left|\det \frac{\partial f_{\theta_i}^{-1}}{\partial \mathbf{m}_i}\right|$ according to the chain rule. Hereafter in this paper we will use F_Θ to denote the chain of these transforms: $F_\Theta = f_{\theta_K} \circ f_{\theta_{K-1}} \cdots \circ f_{\theta_2} \circ f_{\theta_1}$ and use $\left|\det \frac{\partial F_\Theta}{\partial \mathbf{m}_0}\right| =$

$\prod_{i=1}^{K}\left|\det\frac{\partial f_{\theta_i}}{\partial \mathbf{m}_{i-1}}\right|$ for conciseness. Eq. (6) explains the intuition behind normalizing flows: the initial distribution flows through the trajectory of these transforms, changing the probability density and giving the final transformed distribution. For a normalizing flows model, the initial distribution $q_0(\mathbf{m}_0)$ is usually chosen to be known analytically, and since we have the explicit formula of the transform function F_Θ, the result of this method $q_K(\mathbf{m}_K)$ is also analytic.

2.2.2 *Two types of applications*

In this section, we discuss two common applications of normalizing flows in the literature: variational inference and density estimation.

2.2.2.1 *Variational inference*

To solve Bayesian inference problems we treat the target distribution $q_K(\mathbf{m}_K)$ as an approximation to the (unknown) posterior pdf. In order to realize an accurate approximation, we need to maximize $\mathcal{L}[q_K(\mathbf{m}_K)]$ (the goal of variational inference as described in Section 2.1) by iteratively optimizing the flows F_Θ with respect to parameter Θ. This can be accomplished by using a gradient-based optimization method:

$$\Theta^{t+1}=\Theta^{t}+\epsilon\nabla_\Theta\mathcal{L}, \tag{7}$$

where the superscripts t and $t+1$ denote two successive iterations, and real number ϵ is the step size and is chosen to be small and positive. Term $\nabla_\Theta\mathcal{L}$ is the gradient of $\mathcal{L}[q_K(\mathbf{m}_K)]$ with respect to the normalizing flows parameter Θ, and can be calculated by (see Appendix A for derivation):

$$\nabla_\Theta\mathcal{L}=\mathbb{E}_{q_0(\mathbf{m}_0)}\left[\nabla_{\mathbf{m}_K}(\log p(\mathbf{m}_K,\mathbf{d}_{obs}))\nabla_\Theta\mathbf{m}_K+\nabla_\Theta\log\left|\det\frac{\partial F_\Theta}{\partial\mathbf{m}_0}\right|\right]. \tag{8}$$

Note that the expectation term is taken with respect to the initial distribution $q_0(\mathbf{m}_0)$ which is known analytically, so that we can easily draw samples from it and use these samples to obtain unbiased estimates of this expectation term with a Monte Carlo approximation (Kingma & Welling 2014). We would normally perform many iterations in each of which we update the flows parameter Θ by a small amount by calculating $\nabla_\Theta\mathcal{L}$. We can therefore use a relatively small number of samples per iteration (perhaps even only a single sample—Kucukelbir *et al.* 2017) for the Monte Carlo approximation, even though the gradient estimates in each iteration may then be inaccurate. This is because the mean gradient over many iterations should point in approximately the correct direction, since the overall number of random samples used to approximate this mean remains large. Inside the expectation, $\nabla_{\mathbf{m}_K}(\log p(\mathbf{m}_K,\mathbf{d}_{obs}))$ stands for the logarithmic data-model gradient calculated in linearized inversion. By careful design of the flows structure, $\log\left|\det\frac{\partial F_\Theta}{\partial\mathbf{m}_0}\right|$ can be calculated analytically (see Appendix B). Although the flows should be mathematically invertible to ensure we have valid Jacobian determinants, we do not necessarily need the explicit form of their inverse maps when calculating eq. (8).

Fig. 1 shows a simple 1-D example of variational inference using normalizing flows. The blue line is the posterior distribution that we wish to infer, whereas the red line in Fig. 1(a) is the prior distribution (a standard normal distribution), which is also set to be the initial distribution $q_0(\mathbf{m}_0)$ for normalizing flows. We design a normalizing flows model by combining 10 successive planar flows (Rezende & Mohamed 2015, referring to Appendix B herein for details) to solve this inference problem. During optimization, we iterate the following algorithm to maximize $\mathcal{L}_\Theta[q_K(\mathbf{m}_K)]$ with respective to the flows parameter Θ:

1.Draw random samples $\{\mathbf{m}_0^i\}_{i=1}^N$ from the initial distribution $q_0(\mathbf{m}_0)$.
2.Transform these samples through the flows model to obtain $\{\mathbf{m}_K^i\}_{i=1}^N$ by: $\mathbf{m}_K^i=F_\Theta(\mathbf{m}_0^i)$.
3.Calculate $\nabla_\Theta\mathcal{L}$ in eq. (8) by using the transformed samples $\{\mathbf{m}_K^i\}_{i=1}^N$.
4.Update flows parameter Θ using eq. (7).

The red lines in Figs 1(b), (c) and (d) show the transformed distribution after 1000, 5000 and 15 000 iterations. The target distribution is gradually reshaped towards the true solution during the training process, and finally the model converges towards an accurate approximation of the posterior distribution (Fig. 1d).

In order to use normalizing flows for variational inference, we need to evaluate the likelihood of the target distribution but do not necessarily need to sample from it. Examples of this application in the machine learning community include Rezende & Mohamed (2015), Kingma *et al.* (2016), Tomczak & Welling (2016), Berg *et al.* (2018) and Durkan *et al.* (2019b), all of which try to define effective flows functions to model posterior pdfs. A review of these methods can be found in Appendix B.

In seismic tomography, we use normalizing flows to transform an initial distribution towards an optimal approximation to the unknown posterior distribution, and use the transformed variable $\mathbf{m}_K$ to approximate the velocity vector $\mathbf{m}$ in Bayes' rule. Since the velocity field is usually constrained to lie within a specific range, but ideally the flows are updated without any hard constraint, we first apply an invertible function to the initial distribution to convert it into the unconstrained (real) space, then pass this distribution through the trajectory of the normalizing flows, so that the flows can be updated without any hard constraint. Finally, we transform the output of normalizing flows back to the original constrained space using another invertible function. In this work, we use the following two functions to transform random

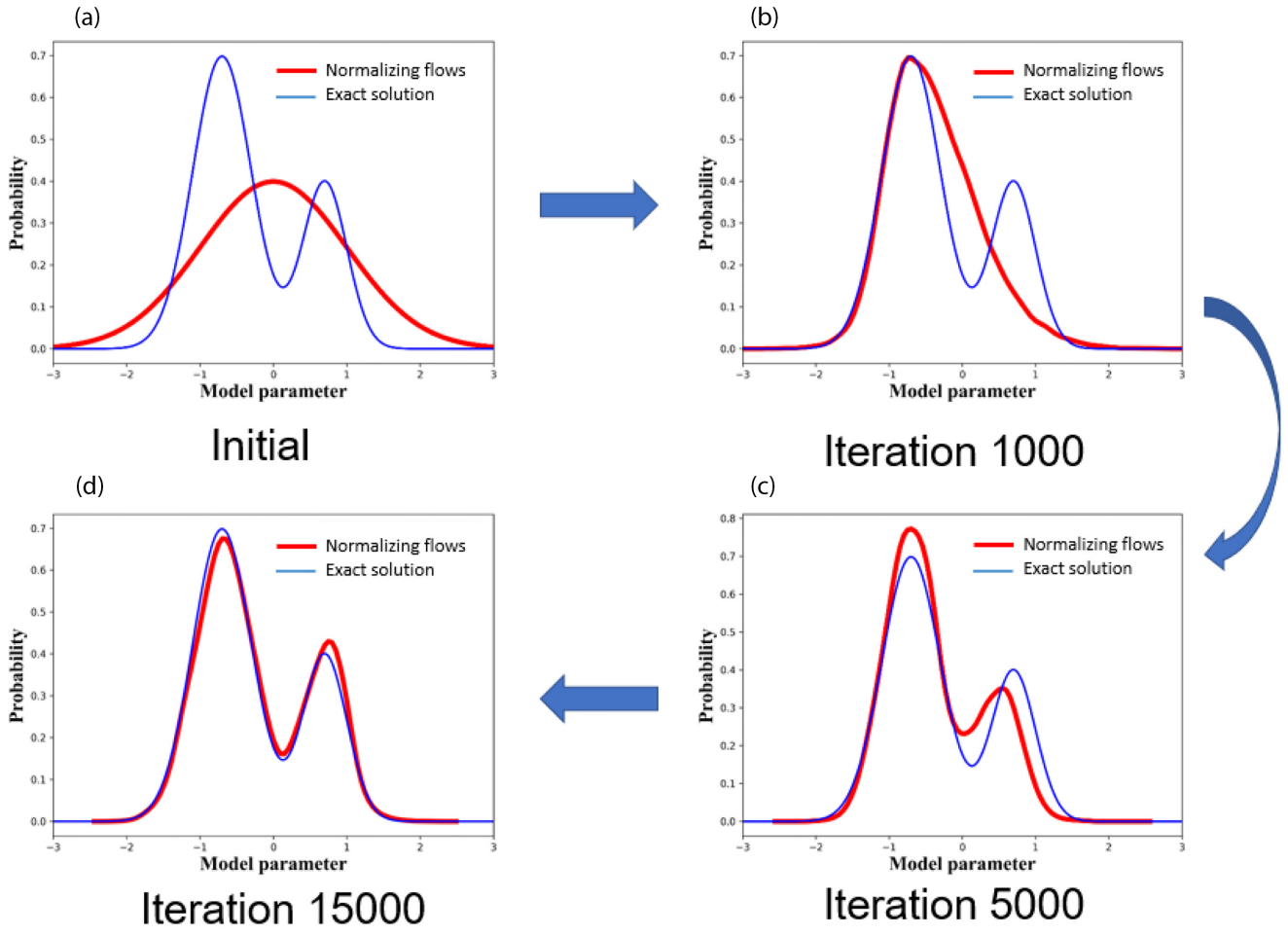

Figure 1. 1-D illustration of normalizing flows for variational inference. The blue line in each figure is the posterior distribution, and the red line in (a) shows a standard normal prior distribution (also used as the initial distribution for normalizing flows). (b), (c) and (d) show the results of normalizing flows after 1000, 5000 and 15 000 iterations, respectively.

variables between the constrained and real spaces (Zhang & Curtis 2020a):

$$\eta_i = T(m_i) = \log(m_i - a_i) - \log(b_i - m_i) \tag{9}$$

and

$$m_i = T^{-1}(\eta_i) = a_i + \frac{b_i - a_i}{1 + \exp(-\eta_i)}, \tag{10}$$

where m_i represents each element of the model vector under the constrained space, η_i is the corresponding unconstrained element, and a_i and b_i are the lower and upper bounds on m_i, respectively. Eq. (9) converts a constrained random variable into an unconstrained one, whereas eq. (10) has the opposite role, to transform an unconstrained variable back to a constrained one. Since these two functions are invertible and have easily evaluated (diagonal) Jacobian matrices, they can be treated as two additional normalizing flows whose parameters (a_i and b_i) are fixed during training. Then the normalizing flows model becomes

$$F_\Theta = T^{-1} \circ f_{\theta_K} \circ f_{\theta_{K-1}} \cdots \circ f_{\theta_2} \circ f_{\theta_1} \circ T. \tag{11}$$

In Fig. 2, we illustrate the effect of these two functions. The blue histogram is constructed from 5000 samples generated from a Uniform distribution, which lies in a constrained space between 0.5 and 3.0, and the orange histogram shows the corresponding samples converted into unconstrained space using eq. (9). If we further apply eq. (10) to those transformed samples represented in the orange histogram, we would obtain the original samples in the blue histogram.

2.2.2.2 Density estimation

Say we have samples generated from an unknown distribution (or we have the ability to generate samples from it), but we cannot necessarily or easily evaluate their underlying probability density. Normalizing flows is well-suited to estimate and parametrize the density of these samples.

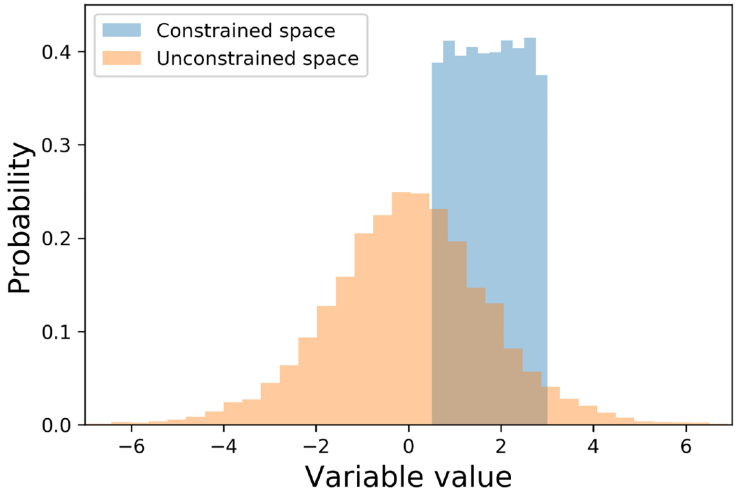

Figure 2. An illustration of the two functions in eqs (9) and (10). The blue histogram is constructed from random samples from a Uniform distribution constrained between 0.5 and 3.0, which is also used as the prior pdf in the synthetic example below. The orange histogram represents the distribution of those samples converted into unconstrained space by eq. (9).

Given a data set of N samples (observations) $\mathcal{D} = \{\mathbf{m}_K^i\}_{i=1}^N$ from a complicated distribution, we can fit the target distribution of normalizing flows to this unknown distribution. Then the log-likelihood of this data set can be estimated by:

$$\log p(\mathcal{D}) = \sum_{i=1}^{N} \log q_K(\mathbf{m}_K^i) = \sum_{i=1}^{N} \left[\log q_0(F_\Theta^{-1}(\mathbf{m}_K^i)) + \log \left| \det \frac{\partial F_\Theta^{-1}}{\partial \mathbf{m}_K^i} \right| \right], \tag{12}$$

where the first term in the summation on the right is the log-likelihood of the observed samples measured by the initial distribution $q_0(\mathbf{m}_0)$ (here sample $\mathbf{m}_0^i = F_\Theta^{-1}(\mathbf{m}_K^i)$ is the sample transformed (inversely) through the flows model), which can be evaluated analytically, and the second term is the volume correction required to transform from $q_K(\mathbf{m}_K)$ to $q_0(\mathbf{m}_0)$. The fitting process maximizes the log-likelihood with respect to the flows parameter Θ using gradient-based methods. Note that estimating the density of a distribution using normalizing flows only requires computation through the inverse direction F_Θ^{-1}, the effect of which is to simplify or 'normalize' a complicated and unknown sampling distribution towards a simpler and known one (pdf q_0). This gives rise to the name 'normalizing flows', which derives from the case where the initial distribution is chosen to be Gaussian.

Once the training is complete, the flows based model can be treated as a so-called *generative model* to generate samples that satisfy the target pdf: this can be accomplished by sampling from the base distribution and transforming the samples through the flows model F_Θ. The effect is similar to variational auto-encoders (Kingma & Welling 2014) and Generative Adversarial Networks (Goodfellow *et al.* 2014) for image and video generation. This application makes normalizing flows very attractive in machine learning applications (Dinh *et al.* 2015, 2017; Papamakarios *et al.* 2017; Kingma & Dhariwal 2018).

2.2.2.3 Comparison

To conclude, these two applications have very different requirements. Flows-based variational inference pushes the initial distribution towards an approximation to the posterior distribution, which requires efficient sampling from the initial distribution $q_0(\mathbf{m}_0)$, evaluating the forward transform of the flows ($\mathbf{m}_K = F_\Theta(\mathbf{m}_0)$) and the Jacobian determinant. By contrast, flows-based density estimation normalizes a complicated distribution towards a predefined base distribution, requiring the calculation of the inverse transform F_Θ^{-1} and its Jacobian information.

2.2.3 Constructing normalizing flows

A normalizing flow should satisfy several conditions in order to be practical for application. It should generally be: (1) invertible; (2) expressive enough to model any desired distribution and (3) computational efficient for the calculation of both forward and inverse transforms and associated Jacobian determinants. In this section, we introduce one specific normalizing flow structure that we use in the rest of this paper. In Appendix B, we review various other ways to construct normalizing flows.

Dinh *et al.* (2015) proposed a non-linear structure called a *coupling flow* for high-dimensional density estimation problems. Fig. 3 shows the main idea of a coupling flow: the D-dimensional input vector $\mathbf{m}_i$ is divided into two partitions $\mathbf{m}_i^A \in \mathbb{R}^d$ and $\mathbf{m}_i^B \in \mathbb{R}^{D-d}$, and for simplicity we set $d = D/2$. Partition $\mathbf{m}_i^A$ remains unchanged and is copied to output $\mathbf{m}_{i+1}^A$. For partition $\mathbf{m}_i^B$, we apply an invertible and element-wise bijection function f to transform each element in $\mathbf{m}_i^B$ into the corresponding element in $\mathbf{m}_{i+1}^B$. Dinh *et al.* (2015) suggested to use a neural network to define the bijection function f to improve the flexibility of the coupling flow. By inputting $\mathbf{m}_i^A$ to the neural network, its output $NN(\mathbf{m}_i^A)$ serves as a vector of hyperparameters used to fully parametrize the bijection f, such that $\mathbf{m}_{i+1}^B$ can be transformed in a

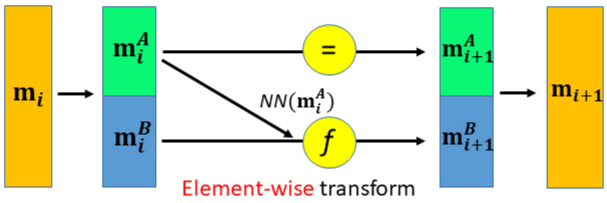

Figure 3. Structure of a coupling flow. The input vector $\mathbf{m}_i$ is divided into two sub-vectors $\mathbf{m}_i^A$ and $\mathbf{m}_i^B$. The former one is directly copied to the output of the flow. It is also input to a neural network $NN(\mathbf{m}_i^A)$ which outputs hyperparameters for an element-wise bijection f; this is used to transform each element of $\mathbf{m}_i^B$ into the same element of $\mathbf{m}_{i+1}^B$. Finally we concatenate the two partitions to obtain the output of the coupling flow $\mathbf{m}_{i+1}$.

desired way by optimizing (training) the neural network. Finally we combine the two partitions $\mathbf{m}_{i+1}^A$ and $\mathbf{m}_{i+1}^B$ to obtain the output vector $\mathbf{m}_{i+1}$. The transform formula for a coupling flow can therefore be summarized as:

$$\begin{aligned}\mathbf{m}_{i+1}^A &= \mathbf{m}_i^A \\ \mathbf{m}_{i+1}^B &= f\left(\mathbf{m}_i^B; NN\left(\mathbf{m}_i^A\right)\right).\end{aligned} \tag{13}$$

The bijection f is called a *coupling layer* and the resulting normalizing flow is the *coupling flow*. This flow is of particular interest because the Jacobian determinant and the inverse transform can be calculated efficiently by (see Appendix C for derivation)

$$\det \frac{\partial \mathbf{m}_{i+1}}{\partial \mathbf{m}_i} = \prod_{j=1}^{D-d} \frac{\partial m_{i+1,j}^B}{\partial m_{i,j}^B} \tag{14}$$

and

$$\begin{aligned}\mathbf{m}_i^A &= \mathbf{m}_{i+1}^A \\ \mathbf{m}_i^B &= f^{-1}\left(\mathbf{m}_{i+1}^B; NN\left(\mathbf{m}_{i+1}^A\right)\right),\end{aligned} \tag{15}$$

respectively, where $m_{i,j}^B$ means the jth element in partition $\mathbf{m}_i^B$.

In practice, we can compose several coupling flows to obtain a more complex layered transform for complicated inference problems. Since one coupling flow leaves one part of its input unchanged (e.g. the partition $\mathbf{m}_i^A$ in Fig. 3), Dinh *et al.* (2015) suggested to exchange the role of copied and transformed partitions within two successive coupling flows, so that the composition of the two flows can modify every element of the input vector $\mathbf{m}_i$.

The element-wise function f should be invertible and differentiable so that the constructed coupling flow is valid, and the expressiveness of a coupling flow is largely dependent on this bijection. Considering that the initial distribution of normalizing flows is typically simple, the final distribution may not approximate complicated posterior distributions well if a simple f is used. In order to improve the expressiveness of coupling flows, many approaches have been proposed to build an effective bijection function f (Dinh *et al.* 2015, 2017; Kingma & Dhariwal 2018; Müller *et al.* 2018; Huang *et al.* 2018; Durkan *et al.* 2019a, b; De Cao *et al.* 2019; Ziegler & Rush 2019). In this work, we use the *rational quadratic splines* proposed by Durkan *et al.* (2019b), and this, together with other kinds of element-wise functions can be found in Appendix B.

3 EXAMPLES

3.1 2-D synthetic test

We first test the effectiveness and efficiency of normalizing flows for traveltime tomography using a simple 2-D synthetic example of a medium which contains a circular low velocity anomaly, shown in Fig. 4. The low velocity anomaly of 1 km s^{-1} is surrounded by the background velocity of 2 km s^{-1}. 16 receivers (white triangles) are located around the low velocity area in a circular shape with a radius of 4 km, each of which will be further treated as a virtual source, thus emulating standard inter-receiver interferometry (Campillo & Paul 2003; Curtis *et al.* 2006; Wapenaar & Fokkema 2006). Under this scenario, we collect 120 observed inter-receiver traveltime data by solving the Eikonal equation using the fast marching method (FMM—Rawlinson & Sambridge 2005) using a 101 $\times$ 101 gridded discretization of the velocity model. Based on these data we wish to infer the velocity structure.

Within the inversion, we parametrize the velocity vector $\mathbf{m}$ into 21 $\times$ 21 regular grid cells of size 0.5 km in both directions, leading to an inference problem of 441 parameters (dimensions). We use a Uniform prior distribution for velocity in each cell, with lower and upper bounds of 0.5 and 3.0 km s^{-1} which encompass the true velocity model. The likelihood function is set using a Gaussian data error distribution with noise level assumed to be $\sigma_{\mathbf{d}} = 0.05s$ for all data points; this defines the data uncertainty information (noise is not actually added to the observed data).

The synthetic data of each model sample (a gridded velocity model) is predicted using the same FMM algorithm as that for the observed data, but under a lower discretization of 41 $\times$ 41 to decrease the computational cost of the forward evaluations. The data-model gradient

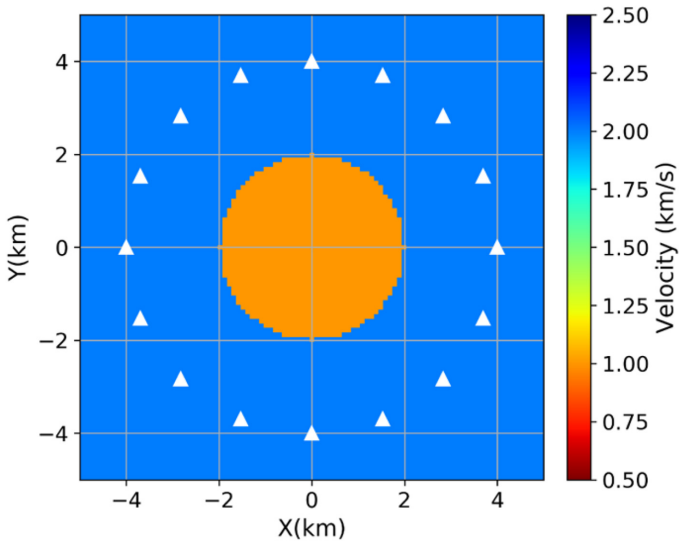

Figure 4. True velocity model of the 2-D synthetic test. Background region (blue area) has a velocity of 2 km s^{-1} while the orange anomaly has a lower velocity of 1 km s^{-1}. 16 receivers (white triangles) are shown in the figure, and each is used as a virtual source thus emulating a typical seismic ambient noise interferometry geometry. Traveltimes between each receiver pair form the data for this tomographic experiment.

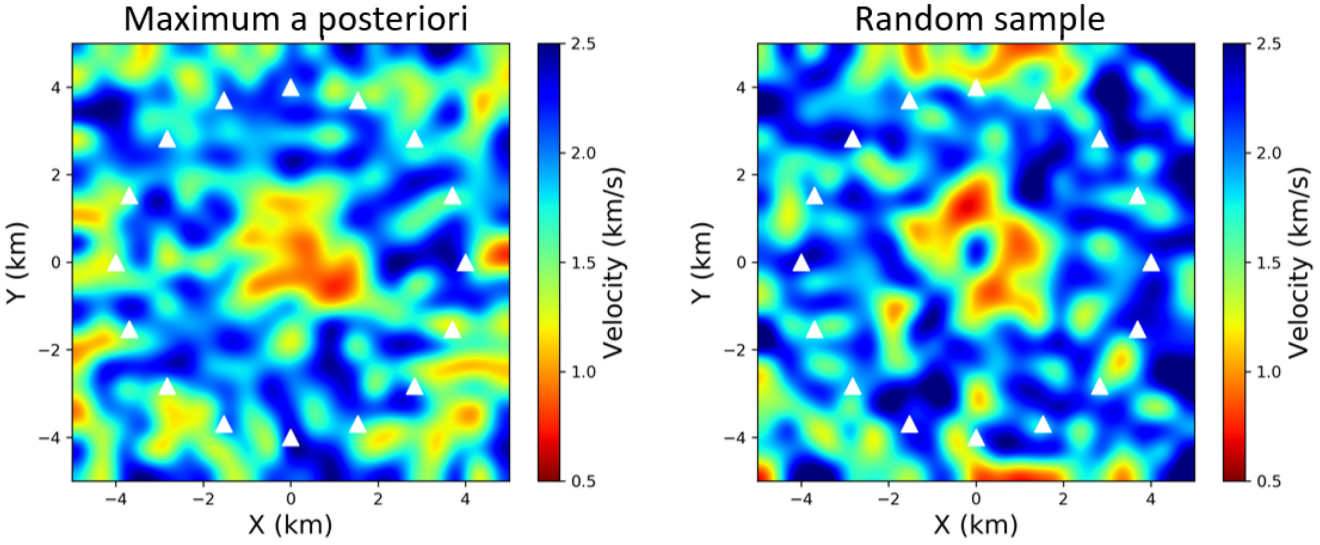

Figure 5. The maximum *a posteriori* (MAP) model (left-hand panel) and another random sample (right-hand panel) drawn from the approximated posterior distribution using normalizing flows. White triangles show the 16 receiver (and virtual source) locations.

is obtained by ray tracing through the traveltime field (the output of the FMM), which will be used to calculate the gradient of $\mathcal{L}[q(\mathbf{m})]$ in variational inference. The overall experimental setting in this example is exactly the same as that used in Zhang & Curtis (2020a), so that we can directly compare the result of normalizing flows with the results using the various methods used in that paper. The normalizing flows configuration used 6 coupling flows associated with rational quadratic splines for the bijection function, and all neural networks used to parametrize the bijection function in coupling flows are fully connected networks. Each network contains 2 hidden layers, and each hidden layer is constructed by 100 hidden units with Rectified Linear Units activation functions. The prior pdf is used as the initial distribution.

To train this model, we update the flows with 3000 iterations. In each iteration, we draw 10 samples from the Uniform initial distribution and convert these samples from this constrained space into unconstrained space. In Fig. 2 we show 5000 samples under constrained and unconstrained spaces to illustrate the effect of eq. (9). Each of the 10 samples is then transformed through the flows model to obtain $\mathbf{m}_K$, which is further used to calculate $\nabla_\Theta \mathcal{L}$ in eq. (8). After the training process, we draw 5000 samples from the initial (prior) distribution, transform each sample through the trained normalizing flows, and use them to calculate statistics of our approximation to the true posterior distribution. Fig. 5 shows the maximum *a posteriori* (MAP) model and another random sample drawn from the above 5000 approximated posterior samples. We find that these two models roughly recover the low velocity anomaly region in the centre of the model and high velocity values around this anomaly, whereas the MAP model provides a more similar structure to the true velocity model, for example in the centre of the low velocity anomaly the MAP model nearly recovers the whole low velocity region while the right panel fails to do so.

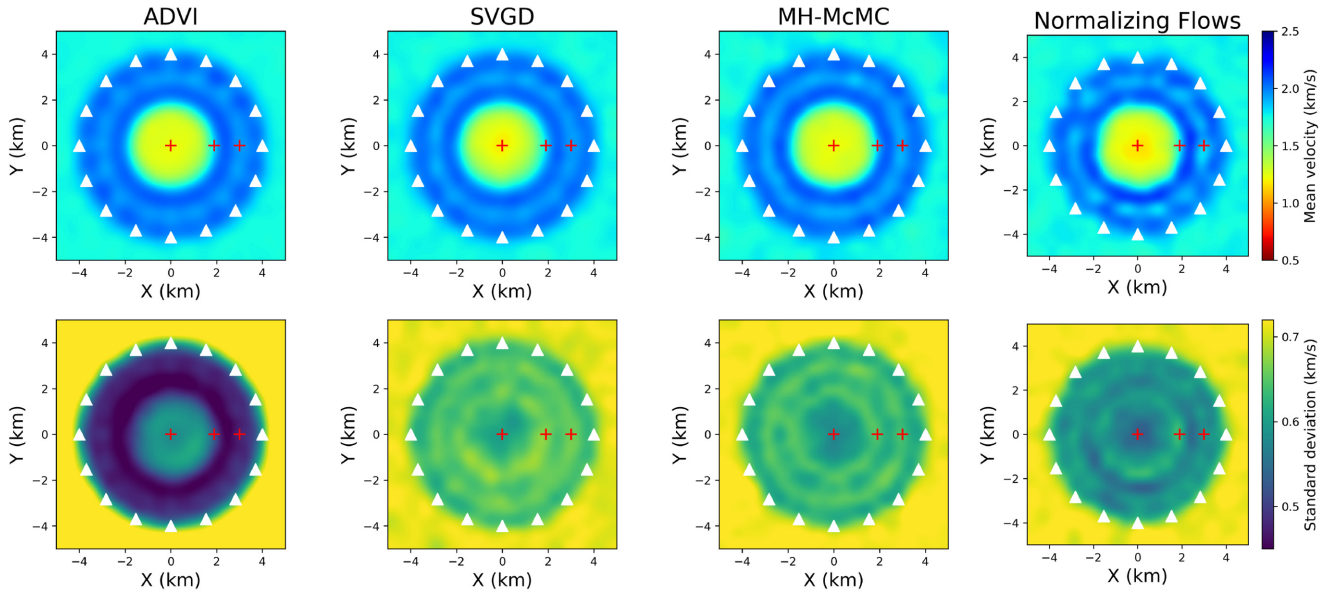

Figure 6. The mean (top row) and standard deviation (bottom row) of the posterior distributions using different methods, respectively ADVI, SVGD, MH-McMC and normalizing flows from left to right. All four results are plotted under the same velocity range for better comparison, as displayed by the colourbar in the right two figures. White triangles show the 16 receiver (virtual source) locations and red crosses denote three specific locations whose marginal distributions are analysed later in the text.

In Fig. 6, we compare the result of normalizing flows to results from ADVI, SVGD and MH-McMC, which are applied to the same problem by Zhang & Curtis (2020a), where ADVI and SVGD are two other variational methods (for details about these three tests, we refer readers to Zhang & Curtis 2020a). From the left-hand column of Fig. 6 to the right, the first row shows the mean models from ADVI, SVGD, MH-McMC and normalizing flows, while the second row shows the corresponding standard deviations. The posterior mean and standard deviation from normalizing flows are very similar to those of the prior information in areas outside of the receiver circle, since most ray paths do not pass through this region. Inside the receiver circle the mean model exhibits the low velocity anomaly well with a slightly higher mean velocity compared to the true model value. There is also a clear lower velocity loop between the receivers and the velocity anomaly. The standard deviation approximately exhibits two higher uncertainty loops within the receiver array. The interior one is due to differences in anomaly shapes and velocity values across the possible models that fit the observed traveltime data. This was also observed previously and assumed to be a robust feature in fully probabilistic tomographic studies (Galetti *et al.* 2015). The other higher uncertainty loop corresponds to the lower average velocity structure between the receivers and the low velocity region; this may be caused by insufficient data being available to constrain this area due to relatively few crossing paths, so that its mean value is closer to the prior value and uncertainty is higher.

In Fig. 6 we observe that the four mean models show nearly the same results and provide a reasonable velocity structure compared to the true model, all of which recover a low velocity region in the centre of the model and a lower velocity loop between the receiver array and the anomaly. Note that the mean model is a statistic of the posterior distribution, so there is no reason why it should equal the true structure. For the uncertainty maps, the right three results are fairly similar with two loop-like higher uncertainty structures, both located between the anomaly and the receivers. Since we often treat the result of MH-McMC as the true solution to a Bayesian inference problem (even though here it is essentially computationally intractable), and since we obtain nearly the same results using three entirely different methods, it is reasonable to assume that the result of normalizing flows is approximately correct. On the other hand, the standard deviation of ADVI fails to recover the two higher uncertainty loops, exhibiting high uncertainty inside the anomaly, and low uncertainty between receivers and the anomaly. This is because ADVI theoretically assumes an underlying (transformed) Gaussian approximation to the posterior distribution which is usually not realistic for tomographic problems (even in this simple example) due to non-linearity in the forward relation between velocity models and traveltimes, so the uncertainty result is not correct.

We also compare the marginal distribution of the three points denoted by the red crosses in Fig. 6 using the four different methods. From left to the right, each column denotes the marginal distribution obtained using ADVI, SVGD, MH-McMC and normalizing flows, respectively. Each row shows the marginal distribution of one specific location: (0, 0) km, (1.8, 0) km and (3.0, 0) km. The first point is located at the centre of the model within the low velocity anomaly, the second is at the edge of the anomaly, and the last point is in the lower velocity loop in the mean model (also the outer higher uncertainty loop in the standard deviation map) of Fig. 6. The dashed red line in each figure shows the non-zero section of the prior pdf at each point. Comparing the results of different methods, all marginal distributions from normalizing flows are similar to but a little less smooth than those from MH-McMC, which is assumed to be the reference solution, and are also similar to those of SVGD. Again, this shows reasonable accuracy from normalizing flows in this tomographic problem. The marginal pdfs of the four methods at point (0, 0) km are nearly the same, all concentrating around the true velocity value (1 km s^{-1}). At the other points, the results

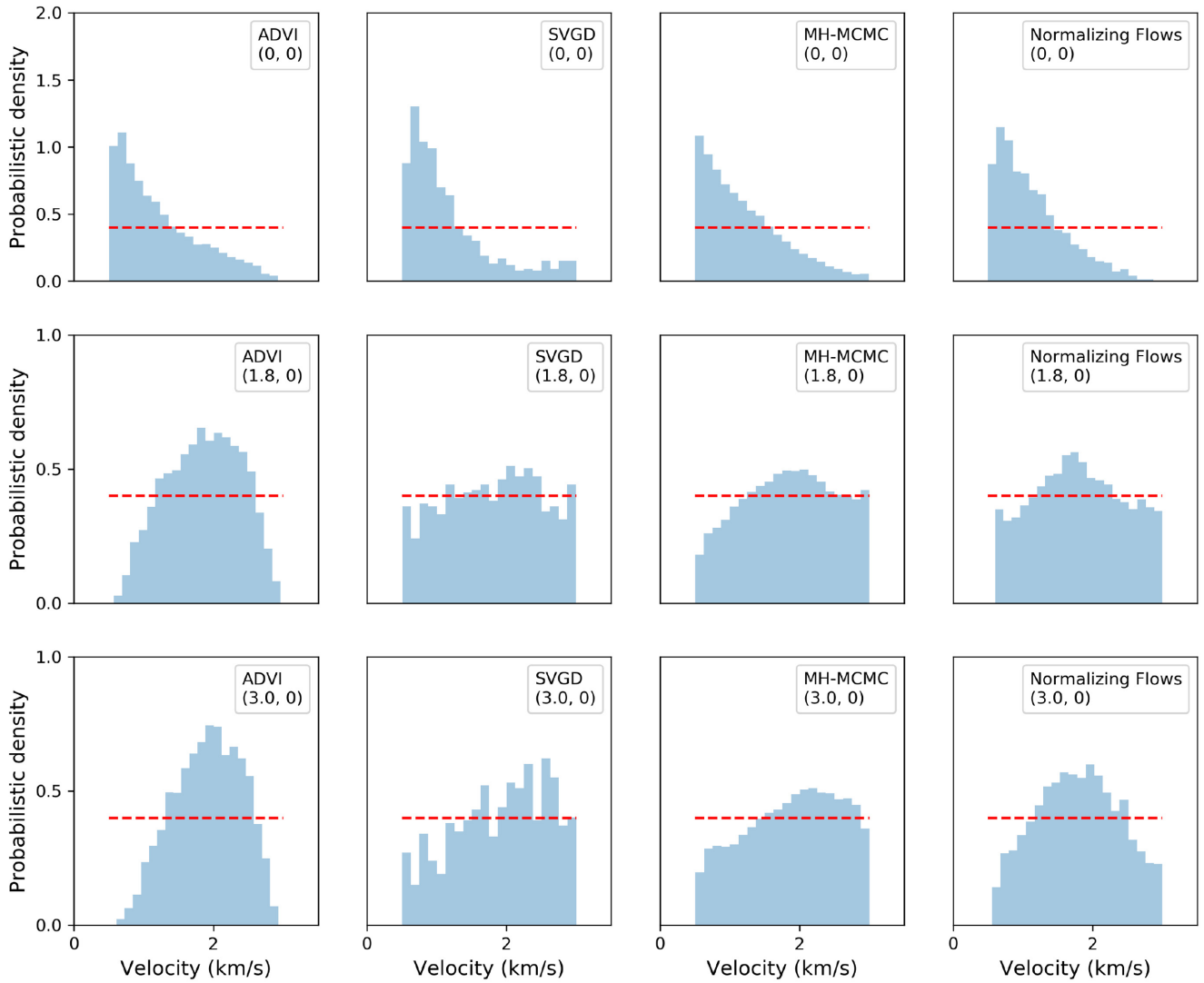

Figure 7. Marginal posterior distributions of three points located at (0, 0) km (the first row), (1.8, 0) km (the second row) and (3.0, 0) km (the third row), corresponding to the three red crosses in Fig. 6. From left to right, each panel shows the marginal pdf using ADVI, SVGD, MH-McMC and normalizing flows, corresponding to the columns in Fig. 6. The location and method in each panel is summarized in the top-right-hand corner. Dashed red lines display the non-zero section of each marginal prior distribution.

of SVGD, MH-McMC and normalizing flows are similar, giving marginal pdfs that are close to the priors, implying that little information is offered by the data. By contrast, ADVI produces a Gaussian-like shape which fails to describe the true uncertainty.

From Figs 6 and 7, we observe that the result of normalizing flows is not as smooth as those from the other methods. For example the two circular shapes in both the mean and standard deviation from normalizing flows are less regular and symmetric compared to SVGD and MH-McMC. In this example, SVGD perturbs 800 samples from prior to posterior distribution and MH-McMC randomly draws an ensemble of samples from the posterior distribution. Both methods make inference using samples only, so the posterior pdf is not parametrized. Provided a sufficient number of samples is used, the natural symmetries of the problem will emerge in the solution, producing partial smoothness. ADVI optimizes a Gaussian distribution so as to best fit the posterior, similarly to normalizing flows, however it applies a linear transform within the Gaussian family, such that the result is essentially defined to be smooth. We deduce that the reason for irregularity in the solution from normalizing flows is the use of a chain of non-linear transforms to manipulate the entire high-dimensional model space in an attempt to directly reshape the initial distribution towards posterior pdf. It is likely that this occurs due to the high number of parameters in the normalizing flows model defined above, which clearly have a non-unique solution, and which limit the method to a relatively low number of flows due to memory requirements during training.

In Table 1 we analyse the computational cost (number of forward evaluations) of the different methods. We also list the number of evaluations required by the Reversible Jump-McMC (RJ-McMC) in Zhang & Curtis (2020a), a method that varies the cell structure of the tomographic model during the inversion (Bodin & Sambridge 2009; Galetti *et al.* 2015). We did not compare the result of RJ-McMC with the four methods above due to the entirely different parametrization used by RJ-McMC which results in a different solution. Nevertheless, RJ-McMC is a commonly used method which often appears to converge more quickly than pure MH-McMC, so it is useful to show its

Table 1. Number of forward evaluations required to reach the solutions in Fig. 6, as well as the number required for Reversible Jump-McMC (RJ-McMC) in fig. 8 of Zhang & Curtis (2020a).

Method	Forward evaluations
ADVI	10 000
Normalizing Flows	30 000
SVGD	400 000
RJ-McMC	3 000 000
MH-McMC	12 000 000

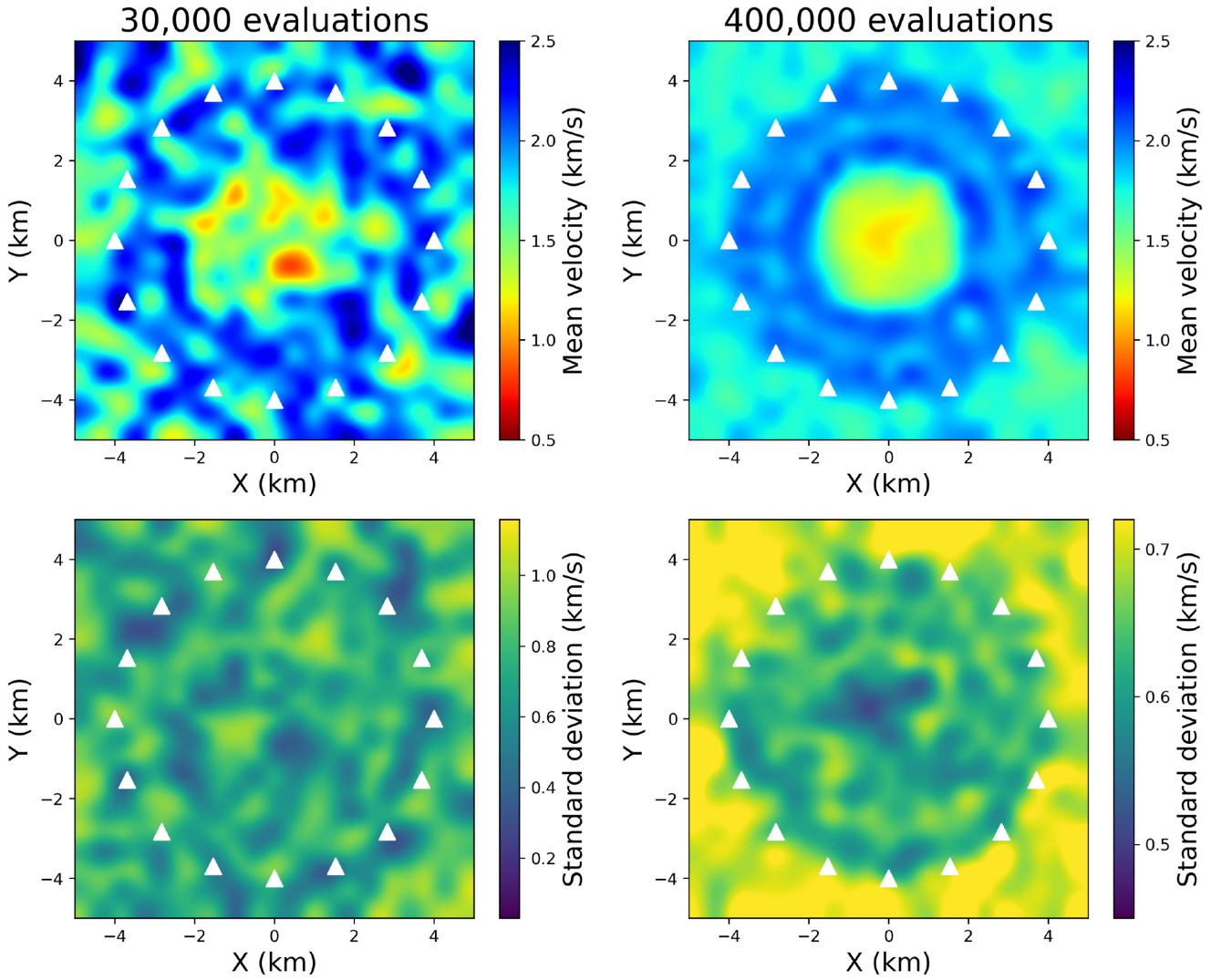

Figure 8. The mean (top row) and standard deviation (bottom row) of the posterior distributions using MH-(McMC) with 30 000 (left-hand column) and 400 000 (right-hand column) forward evaluations, respectively. White triangles show the 16 receiver (and virtual source) locations.

cost. From Table 1, we find ADVI is the cheapest method, but above we observe that it fails to provide the correct shape of posterior pdf due to its implicit Gaussian assumption. Normalizing flows is the most efficient method that gives a reasonably accurate inference result, while requiring the same order of magnitude of computation as that for ADVI. All three variational methods decrease the computational cost compared to both Monte Carlo based methods.

It is difficult to compare the computational cost of Monte Carlo methods to optimization based methods because detecting statistical convergence of the former is often a rather subjective process, and in this case the Monte Carlo runs were stopped only once they had a fairly stable mean and standard deviation. To make a fairer comparison of the computational performance of variational and Monte Carlo methods, in Fig. 8 we show two other MH-McMC tests using 30 000 and 400 000 forward evaluations, the same number of evaluations as were used by normalizing flows and SVGD, respectively, in the above tests. For the result in the left-hand column of Fig. 8 with 30 000 forward evaluations,

we run 3 Markov chains in parallel for 10 000 iterations each. The first 5000 samples are discarded as the burn-in period, and the remaining 5000 samples are used to calculate statistics of the posterior pdf. For the right column with 400 000 forward evaluations, MH-McMC is implemented by running five chains, each of which draws 80 000 samples. For each chain, we discard the first 40 000 samples as the burn-in period, after which we retain every 50th sample to reduce the correlation between samples. The retained samples are used to represent an ensemble of posterior samples which are used to calculate statistics. It is obvious that the three chains in the first test did not converge and provide very little information about the true posterior pdf. The mean model of the second test provides a very rough approximation to the true velocity structure, and is similar to the four mean models shown in Fig. 6, yet it is not as smooth as those from Fig. 6. However, this test fails to approximate the true uncertainty map to anything like the detail given by the solutions in Fig. 6. We think this is because the 10 chains did not fully converge in this test, meaning that 400 000 samples (forward evaluations) are not sufficient to approximate the true posterior pdf. Considering Figs 6 and 8 together, the Monte Carlo methods converged to smoother and (probably) more reliable solutions than variational methods given a very large number of forward evaluations, but provide worse results if they are restricted to use the same number of forward evaluations. Although it may be difficult to make a direct comparison between the methods' costs, in this test the 2–3 orders of magnitude reduction in cost of normalizing flows compared to the McMC methods seems significant, as are the definite and standard convergence criteria that can be applied in optimization based variational methods, and their ability to be fully parallelized (Zhang & Curtis 2020a). In the considerably more complex real-data example below, the difference in number of samples required by the various methods is far more apparent. This makes variational methods more attractive for large scale data sets with higher dimensionality in real applications.

3.2 Love wave tomography of the British Isles

In the second example, we conduct Love wave tomography using ambient noise data from the British Isles. The British Isles are a group of islands in the North Atlantic off the northwestern coast of continental Europe. In past decades, active earthquakes around this area tend to be infrequent and most have a small magnitude (the largest ever observed earthquake had a magnitude of 5.9 M_W). On the other hand, the British Isles are surrounded by seismic ambient noise sources from the Atlantic Ocean, the North Sea and the Norwegian Sea. Due to the limited data available from active earthquakes and to the natural geographic situation of the British Isles, it is therefore common to compute surface wave velocity maps from ambient noise tomography using seismic interferometry (Nicolson *et al.* 2012, 2014; Galetti *et al.* 2015, 2017).

Fig. 9 displays 61 seismometer locations around the British Isles used in this test and the terrane boundaries of the British Isles. Ambient noise data were recorded in 2001–2003, 2006–2007 and in 2010, respectively for three different subarrays, and all recordings contain one vertical (Z) and two horizontal (north and east) components of ground motion. The vertical component was previously used for Rayleigh wave tomography (Nicolson *et al.* 2014), whereas we perform Love wave tomography using traveltime estimates constructed from the two horizontal components by Galetti *et al.* (2017). During data processing, the noise data was firstly cross-correlated among all the possible interstation pairs, and the positive time (causal) part and time-reversed negative time (acausal) part of the correlation were stacked. This resulted in one-sided Green's functions estimates for all available interstation travel paths (some were removed due to quality control), which were used to estimate the traveltimes of Love waves of different periods. Detailed station network information and a description of the data processing procedures can be found in Galetti *et al.* (2017). In this test, we use the traveltime measurements of Love waves at 10 s period.

For tomography we fix the imaged area to lie within longitude 9°W–3°E and latitude 48°N–61°N, which fully encompasses the British Isles. The whole region is parametrized into a regular grid of 37 $\times$ 40 cells with spacing of 0.33° in both latitude and longitude, which leads to an inference problem with a parameter vector of 1480 dimensions. For Bayesian inversion, the prior pdf for the group velocity in each cell is chosen to be Uniform ranging from 1.56 to 4.81 km s^{-1}: the average value is obtained by measuring the average velocity across all valid ray paths, and the upper and lower bounds of the Uniform velocity is chosen to exceed the range of velocities observed on the dispersion curves (Galetti *et al.* 2017). The likelihood function is chosen to be a Gaussian distribution with a traveltime data uncertainty of each inter-receiver path estimated from the standard deviation of dispersion curves constructed by stacking randomly selected subsets of daily cross-correlations (Galetti *et al.* 2017). The predicted traveltime data in the inversion is calculated by solving the Eikonal equation using the FMM algorithm using 73 $\times$ 79 regularly gridded cells (four times as many as are inferred by tomography).

We apply the three variational methods, normalizing flows, ADVI and SVGD, as well as MH-McMC for comparison. For normalizing flows, we use the Uniform prior distribution as the initial distribution and choose 10 coupling flows with rational quadratic splines to construct the inference model as described in the previous section. During the training process, we update the flows parameters with 5000 iterations and draw 20 samples from the initial (prior) distribution per iteration to approximate $\nabla_\Theta \mathcal{L}$ in eq. (8). Finally, we generate 2000 samples from its posterior distribution to calculate the mean and standard deviation maps. For ADVI, the initial distribution is chosen to be a standard Gaussian in the unconstrained (real) space, and we perform 10 000 iterations using 1 sample per iteration to estimate $\nabla_\Theta \mathcal{L}$. Since the estimate of $\nabla_\Theta \mathcal{L}$ in each iteration is therefore inaccurate, we use a small step size ϵ (eq. 7) so that over many iterations the optimization converges. Similarly to normalizing flows, we draw 2000 samples to estimate posterior statistics. For SVGD, we approximate the initial distribution by 1000 samples generated from the prior distribution and perform 600 iterations to perturb those samples from prior to posterior space. These 1000 samples are used to calculate the mean and standard deviation of the posterior distribution. MH-McMC is implemented by running 10 parallel Markov chains for 1.5 million iterations each. The first 1 million samples are discarded as the burn-in period, and we retain every 100th sample after the burn-in period to reduce the correlation between samples. The retained samples are used to represent an ensemble of posterior samples which are used to calculate statistics.

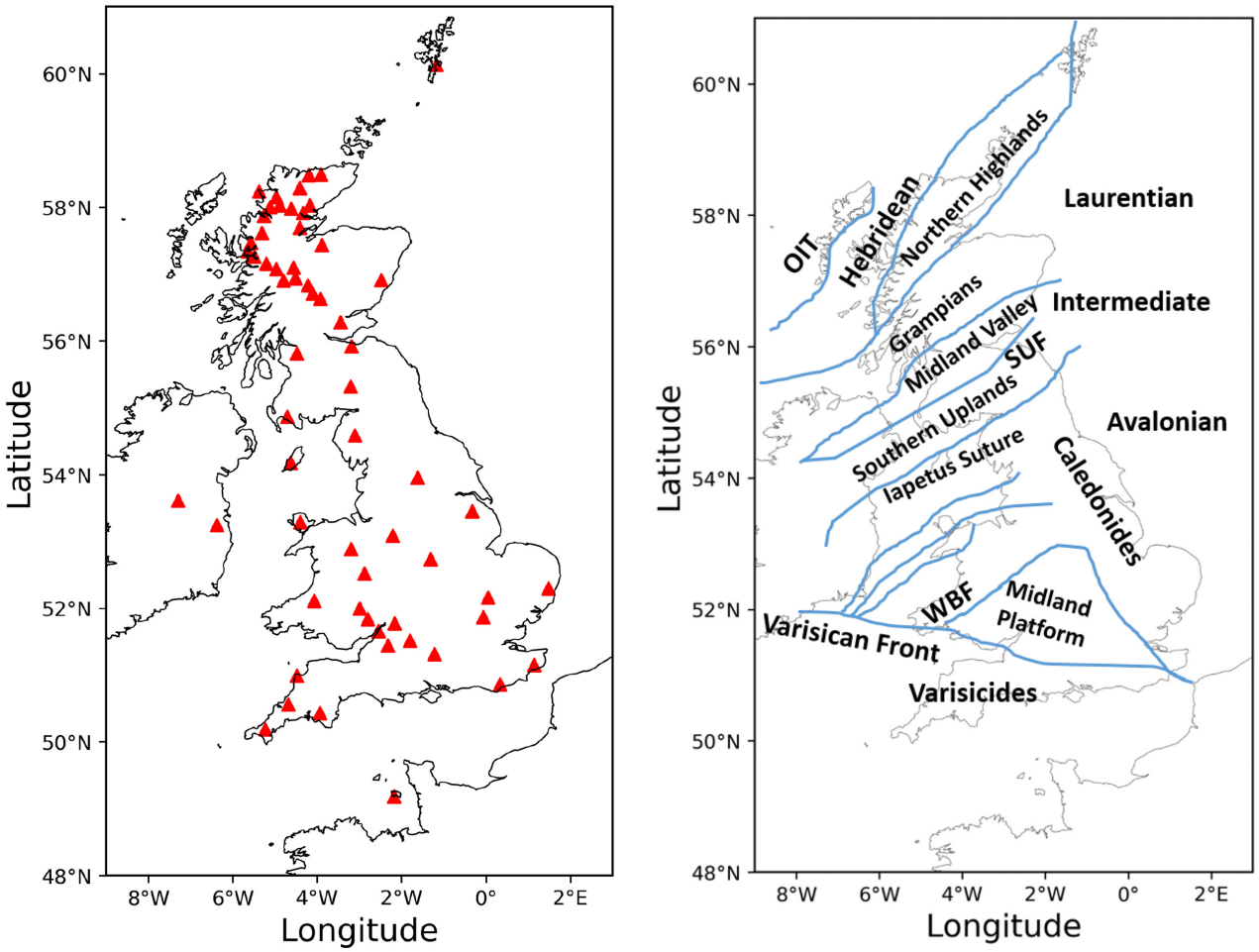

Figure 9. (a) The location of the 61 seismometers (red triangles) around the British Isles used by Galetti *et al.* (2017) and in this paper to perform Love wave tomography. The receivers are also treated as virtual sources for ambient noise interferometry. (b) Terrane boundaries in the British Isles used in the main text. Abbreviations in the figure are as follows: OIT, Outer Isles Thrust; SUF, Southern Uplands Fault; WBF, Welsh Borderland Fault System.

Figs 10(d) and (h) display the average and standard deviation maps of the Love wave group velocity across the British Isles using normalizing flows at 10 s period, and the annotations mark some representative locations discussed below. The structure of the mean model shows good consistency with the known geology (e.g. Fig. 9b) and previous tomographic studies of the British Isles (Nicolson *et al.* 2012, 2014; Galetti *et al.* 2015, 2017). For example, a clear high velocity area of metamorphic and igneous origin is observed in the Scottish Highlands (annotation 1 in Fig. 10d; hereafter we use the number of each annotation to denote its corresponding location for simplicity). Around 4°W, 55°N (2 in Fig. 10d), there is a SW–NE trending high velocity area in the Southern Uplands. Bounded between these two areas, the Midland Valley is a low velocity zone (around 3.5°W, 55.5°N – 3 in Fig. 10d). Another two low velocities can also be observed, one of which is located at the offshore sedimentary basins along the East coastline of mainland Britain from 3°W, 56°N to 1°E, 53°N (4 in Fig. 10d) and the other one is from East Ireland to Southwest Wales (5 in Fig. 10d). The East Irish Sea (4.5°W, 54°N – 6 in Fig. 10d) is also covered by a low velocity region, whereas Northwest Wales (around 4°W, 53°N – 7 in Fig. 10d) is characterized by high velocities. In northern England, two north–south trending high velocity regions are located at 3°W, 54.5°N (the Lake District in the Northwest England – 8 in Fig. 10d) and 2°W, 54°N (9 in Fig. 10d), and another high velocity zone can be found down to East Midlands at 1°W, 53°N (10 in Fig. 10d). A large low velocity area can be observed around the Midland Platform which spans several sedimentary basins like the Cheshire Basin (2.5°W, 52.5°N – 11 in Fig. 10d), the Anglian-London Basin (0°W, 52°N – 12 in Fig. 10d), the Weald Basin (0°W, 51°N – 13 in Fig. 10d) and the Wessex Basin (3°W, 50.5°N – 14 in Fig. 10d).

The standard deviation map in Fig. 10(h) shows a high uncertainty similar to prior values in offshore areas since few ray paths go through the open marine regions. Elsewhere the velocity uncertainty reflects how the velocity model is constrained by traveltime data. For example, uncertainty is relatively low in the Highlands (1 in Fig. 10h) and southern England (15 in Fig. 10h) since stations are densely distributed in those areas. Other small low-uncertainty areas associated with high or low average velocity anomalies can also be found around (4°W, 52°N – 16 in Fig. 10h), (2.5°W, 52.5°N – 15 in Fig. 10h) and elsewhere. There is a higher uncertainty loop around the low velocity anomaly in the East Irish Sea (6 in Fig. 10h), and this phenomenon is also observed in the synthetic test, since different anomaly shapes and velocity values

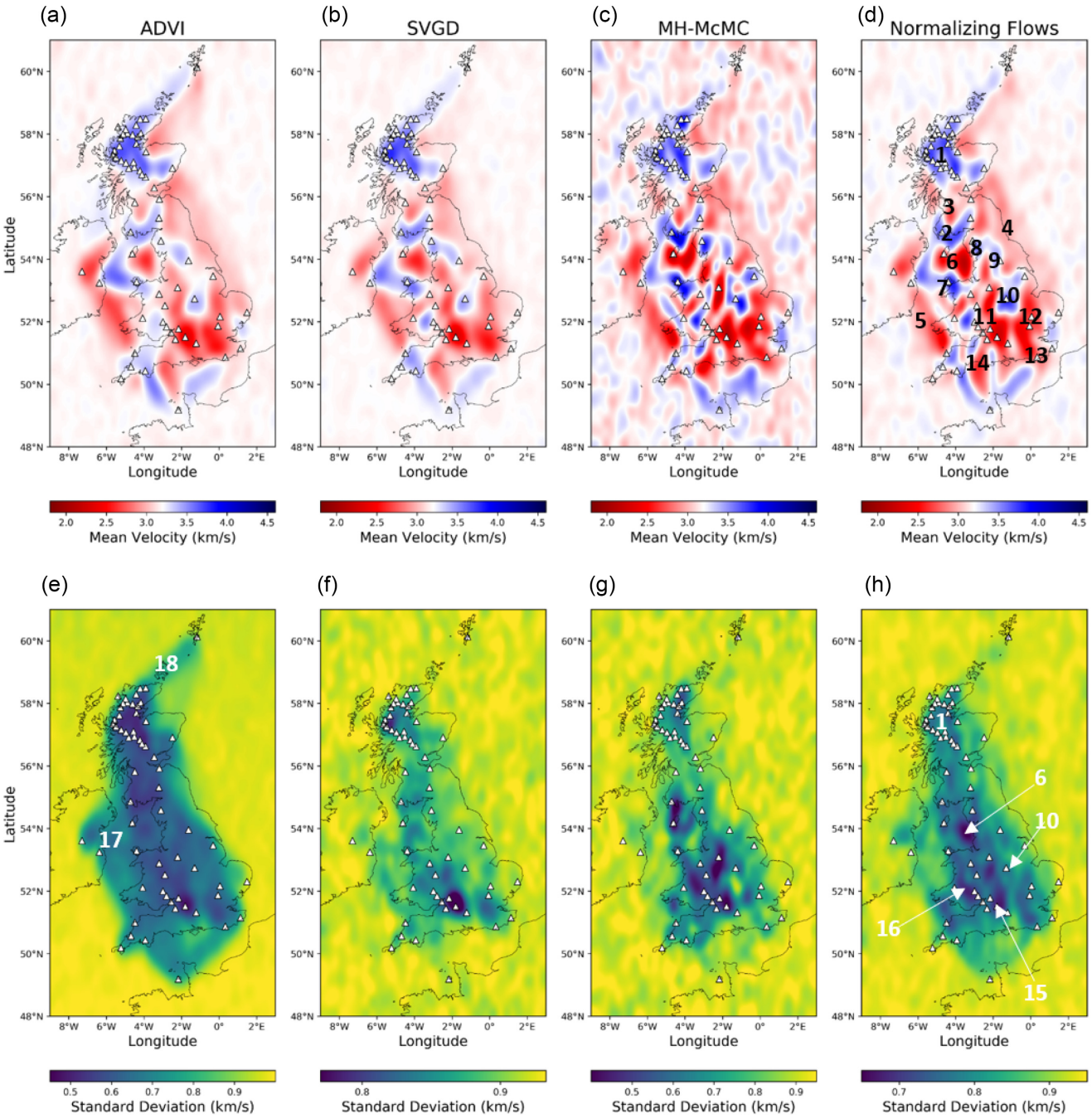

Figure 10. The results of Love wave group velocity maps of the British Isles at 10 s period: mean (top row) and standard deviation (bottom row) of the posterior distributions using different methods, respectively ADVI, SVGD, MH-McMC and normalizing flows from left to right. White triangles in all figures show the receiver (virtual source) locations. Annotation on each figure is used to denote specific locations discussed in the main text.

would fit the same traveltime data (Galetti *et al.* 2015, 2017). Another higher uncertainty structure around the East Midlands high velocity anomaly (1.5°W, 53°N – 10 in Fig. 10h), is observed probably for similar reasons.

Figs 10(a)–(c) show the average velocity models of ADVI, SVGD and MH-McMC respectively, and Figs 10(e)–(g) display the uncertainty maps from those three methods. Similarly to the synthetic test above, the velocity maps of both ADVI and SVGD are smoother than that from normalizing flows. The results of MH-McMC and normalizing flows show high consistency: both maps are less smooth and provide more detailed information compared to the other two maps. Around 2°W, 51°N and 4°W, 51.5°N, normalizing flows and MH-McMC produce some small structures comprising spatially rapid velocity transitions which are not observed in the other two results. Nevertheless, the three mean models exhibit similar structures compared to the mean map from normalizing flows: we observe high velocity regions in the Highlands, in the Southern Upland (at 4°W, 55°N); we obtain a similar SW–NE trend compared to normalizing flows and so on. Low velocity structures are

Table 2. Number of forward evaluations and the total elapsed time for ADVI, normalizing flows, SVGD, RJ-McMC and MH-McMC to obtain the tomographic results in Fig. 10. See main text for discussion of parallelization used in each method (this affects the right column only). Note that the result of RJ-McMC is from Galetti *et al.* (2017) and produces a quite different inference result due to the variable parametrization used in that work.

Method	Forward evaluations	Elapsed time (hr)
ADVI	10 000	6.95
Normalizing flows	100 000	7.83
SVGD	600 000	31.71
RJ-McMC	48 000 000	∼720
MH-McMC	15 000 000	660

also observed along the East coastline of the mainland Britain, to the east of Ireland down to Southwest Wales, at the East Irish Sea, around the Midland Platform. The high consistency between the four mean models also suggests that the obtained group velocity map is accurate.

Across the area inside the receiver array, the uncertainty estimates from ADVI are generally lower than those from the other methods. ADVI finds lowest uncertainty in the Highlands and in southern England (around 2°W, 51.5°N) where seismometers are densely spaced, but all other areas maintain nearly the same uncertainty level without much variation. Around the West Irish Sea (17 in Fig. 10e) and the North Sea area Northeast of Scotland (18 in Fig. 10e), the uncertainty value is low, whereas for SVGD, MH-McMC and normalizing flows these two areas present higher values. We observed a similar phenomenon in the synthetic test in Fig. 6: at the location of the outer uncertainty loop, ADVI provides an apparently biased result with a lower uncertainty value, while SVGD, MH-McMC and normalizing flows successfully recover the higher uncertainty loop. We therefore draw the conclusion that the uncertainty map in Fig. 10(e) may be a biased result due to ADVI's underlying assumption of a Gaussian-based posterior distribution. In the standard deviation map of SVGD and MH-McMC in Figs 10(f) and (g), we observe low uncertainty areas at Scotland, three separate lower uncertainty areas around southern England and so on, all of which correspond to similar results from normalizing flows in Fig. 10(h). On the other hand, despite those main uncertainty features that are quite similar to each other, we can still observe some inconsistent details from these three results—for example around the East Irish Sea. We think this may be caused by the fact that variational methods only seek an optimal approximation to the true posterior distribution whereas Monte Carlo methods directly sample from the posterior pdf itself, so there must be some differences among the results from the two variational methods and MH-McMC, especially for such a high-dimensional inference problem. Nevertheless, since the result of MH-McMC can be treated as an unbiased approximation of the true solution of a Bayesian inversion problem if it has converged to a reasonably stable equilibrium, and since we obtain similar mean and standard deviation results using entirely different methods, we infer that normalizing flows produces a reasonable estimate of the posterior means and standard deviation of the group velocity of the British Isles.

Table 2 lists the computational cost of the four methods in this example. ADVI required 10 000 iterations using 1 sample per iteration, which gives 10 000 forward evaluations in total, and an elapsed time of 6.95 hr. Since we only used 1 sample to update the variational parameters in each iteration, it is relatively hard to parallelize its training process. Normalizing flows performed 5000 iterations with 20 samples per iteration, so the total number of forward evaluations is 100 000. During the inversion, we used 10 cores to parallelize the forward simulation of the 20 samples and to train the neural networks in every iteration, which decreases the total elapsed time (including neural network training time) to 7.83 hr, only slightly longer than ADVI. This is easy to understand: the number of forward evaluations implemented on each core are the same for ADVI and normalizing flows (10 000 for both methods). The remaining time difference is mainly caused by the different internal complexity of the two methods themselves: the normalizing flows contain more learnable parameters than ADVI, and therefore need more calculations to train. For SVGD, 1000 samples were perturbed 600 times, so the total computational cost is 600 000 forward evaluations. For a fair comparison, we also parallelized across 10 cores to perform SVGD giving an elapsed time of 31.71 hr. SVGD required six times more evaluations than normalizing flows, but the total elapsed time was only about 4 times greater because normalizing flows required additional computation for neural network training, while SVGD required very few additional computations per iteration (to calculate the kernel functions), which are nearly negligible compared to the cost of the forward evaluations. MH-McMC drew 15 million samples in total with 10 cores for parallelization across 10 chains, and the elapsed time as 660 hr in all. We also list the cost of RJ-McMC conducted by Galetti *et al.* (2017). In their experiment, they used 16 chains and 3 million samples per chain, and took about one month of computation time. Again, we did not compare the result of RJ-McMC with those in Fig. 10 due to the variable parametrization used in that work which led to different results, and only list the computational cost for a rough comparison. Table 2 demonstrates the efficacy of performing variational inference in large scale problems, since McMC based methods become extremely expensive, if not intractable, for high-dimensional Bayesian inversion in real applications.

In order to compare the results more fairly between Monte Carlo and variational methods, in Fig. 11 we show the mean (left-hand panel) and uncertainty (right-hand panel) maps of one Markov chain with 2 million samples; this is still more than the numbers used for the variational methods, but it removes the possibility that our subjective assessment of when the Monte Carlo method had converged led to the large number of samples attributed to the method above. The mean model only provides a few of the main features that we observed previously in variational and full McMC results, while it fails to provide more detailed structures, and the standard deviation map hardly provides any useful information about the posterior pdf. Comparing the results from Figs 10 and 11 and the results from the synthetic test in Fig. 6, we

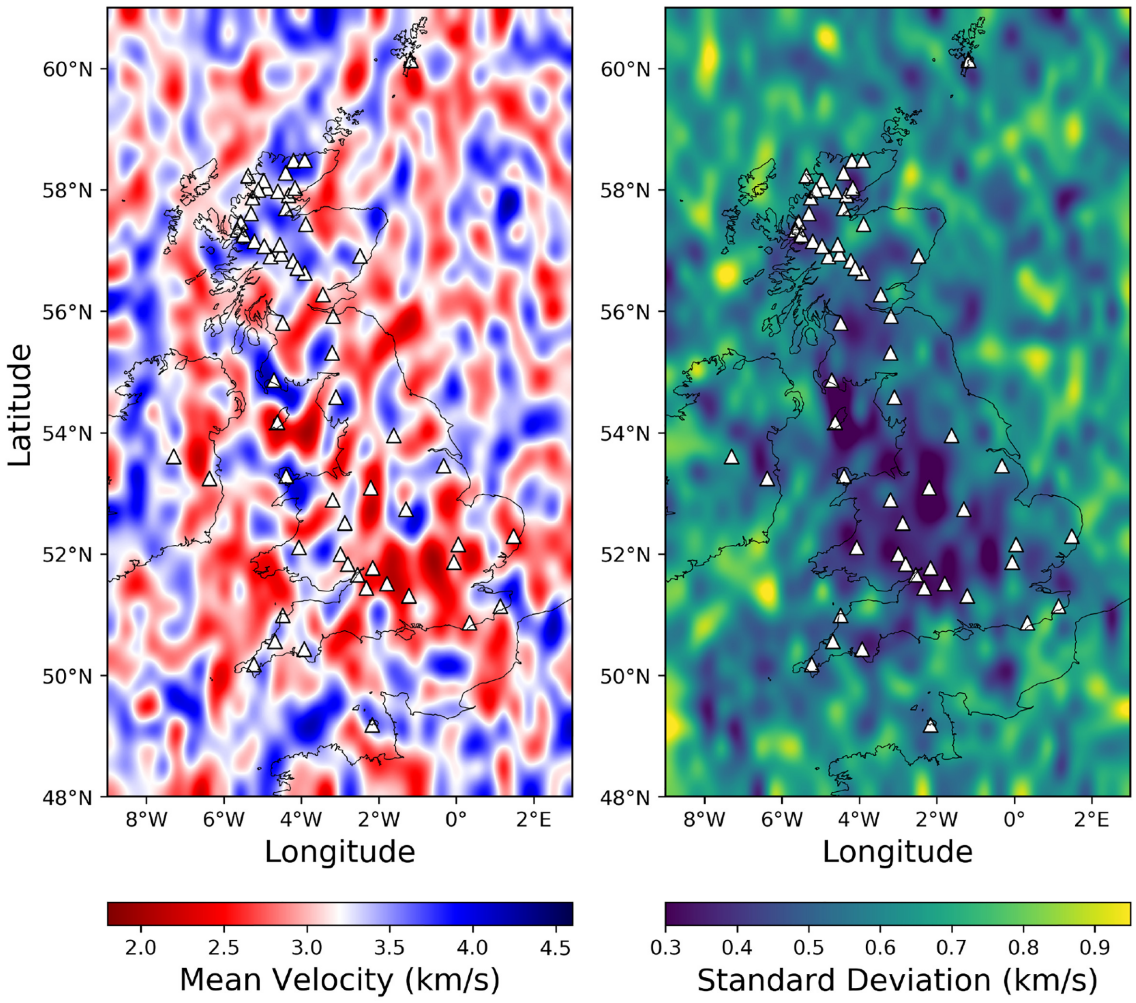

Figure 11. The mean (left-hand panel) and standard deviation (right-hand panel) maps of MH-McMC using one Markov chain with 2 million samples.

conclude that in synthetic tests, it is possible that we could use fewer Monte Carlo samples to obtain similar quality result, while in this more complex problem with such a high dimensionality, this is not generally possible and we need far more samples to obtain a respectable result which is usually computational demanding.

We thus draw similar conclusions to those in the synthetic test: variational methods provide an efficient approach for Bayesian tomography. In this example they provide a significant improvement over Monte Carlo based sampling methods which generally require millions of forward evaluations for such a high-dimensional imaging problem. All three variational methods provide a convincing average velocity map, while ADVI provides a biased uncertainty result. Normalizing flows and SVGD produce more convincing uncertainty estimates, but the former requires far less elapsed time than the latter, just slightly greater than ADVI.

4 DISCUSSION

Using normalizing flows to perform Bayesian inversion within an optimization framework, we seek the closest approximation to the posterior distribution. This contrasts with taking random samples from the posterior pdf in Monte Carlo methods, so the efficiency is improved. The method is based on several invertible and differentiable transforms (the flows) which are sequentially applied to an analytically known and simple initial distribution, such that the transformed distribution is an approximation to the posterior distribution. ADVI can be treated as a special case in which we use a single invertible transform (see Appendix B: mean field ADVI corresponds to a diagonal linear flow and full-rank ADVI to a triangular linear flow). This converts a standard Gaussian distribution into another Gaussian that is best-fit to the true posterior distribution after a simple transform (eq. 9) has been applied. The variational distribution is limited to the Gaussian family, thus ADVI can only solve Gaussian-like problems with unimodal posterior pdf. This explains results in our two tests: ADVI provides an accurate mean model but incorrect uncertainty. Nevertheless, considering that the method is very efficient, the result of ADVI could be used as the initial distribution for normalizing flows since the result of ADVI is analytic. In this scenario, both the required number of flows and their complexity can hopefully be decreased; for instance, we may only need to use flows that are able to model multimodal distributions such as planar flows (Rezende & Mohamed 2015).

SVGD is also based on invertible transforms which iteratively perturb prior samples towards samples of the posterior distribution, and uses those samples to estimate the posterior distribution itself. The perturbation direction is optimized based on the kernelized Stein discrepancy (Liu *et al.* 2016) within the reproducing kernel Hilbert space (Liu & Wang 2016). The main difference between SVGD and normalizing flows is in the invertible transforms used: the transforms in normalizing flows are explicitly known with some fixed formula and we optimize the hyperparameters of flows to model the posterior, while SVGD uses implicit transforms which push the samples through their trajectory. The final analytic form of the posterior is never estimated or approximated in SVGD. The result of SVGD is an ensemble of posterior samples and the number of samples (usually fewer than 1000) is a compromise between efficiency and accuracy: for very high-dimensional problems it might be impossible to use hundreds of samples to represent target statistical properties of the posterior distribution. On the other hand, SVGD should theoretically be more effective than normalizing flows owing to the implicit transform used in the former: the process of constructing fixed-formula normalizing flows can be understood as a way to mimic the effect of SVGD in order to select the optimal perturbation direction applied to the initial distribution. Our current results for traveltime tomography did not provide sufficient evidence on this point, so it should be investigated further by implementing more complicated and non-linear geophysical inference problems, for example full waveform inversion (FWI, Zhang & Curtis 2020b).

Both of our numerical tests show that normalizing flows are the most efficient method to approximate the correct uncertainty result, whereas ADVI, while cheaper, usually provides provides biased uncertainty information. Normalizing flows are easy to parallelize at the sample level within each iteration (calculation of eq. 8) to further improve the efficiency, whereas McMC methods are hard to parallelize on the sample level due to the detailed-balance property required of Markov chains (O'Hagan & Forster 2004). Although for some large scale tomography and FWI cases, we can parallelize the forward and gradient evaluation on the source (shot) level for McMC, this is less efficient since the lower level of parallelization often means more time overhead for synchronization. Nevertheless, there are many ways to make McMC more efficient including using RJ-McMC as noted above, but also the No-U-Turn sampler (Hoffman & Gelman 2014), Hamiltonian Monte Carlo (Fichtner *et al.* 2019; Gebraad *et al.* 2020) and informed proposal Monte Carlo (Khoshkholgh *et al.* 2020). Also, the No Free Lunch theorem states that no method is better than any other when averaged across all possible problems (Wolpert & Macready 1997), so we note that our conclusions extend only to the class of traveltime tomographic problems studied here. Similar tests should be performed for other important classes of problems such as FWI or inference using other (e.g. non-wave based) physics.

In this paper, we use coupling flows for both tests, and one potential deficiency is in the training of the neural networks (which is used to model the bijection function f in Fig. 3). When coping with high-dimensional problems such as in 3-D tomography, we end up with very large networks, so the computational cost of optimizing the networks in each iteration cannot be neglected and may even dominate the whole calculation. Under such circumstances, directly comparing the forward evaluation numbers as in Tables 1 and 2 would be less relevant, as it is even possible that the elapsed time required for normalizing flows would be longer than that for SVGD. Future improvements should consider how to decrease this overhead, for example by constructing more effective flow structures to reduce either the size of the neural networks or the required number of flows, or both.

Zhang & Curtis (2020a) proved that SVGD significantly decreases the computational requirement compared standard Metropolis-Hastings McMC. In this work, we further decrease the total elapsed time required by SVGD to the order of hours for our real data application—to nearly the same time as that for ADVI, which converges far more rapidly than McMC. In the future, normalizing flows may bring fully non-linear uncertainty assessment of tomographic models into the new realm of running on standard desktop computers, which is hardly possible using other existing methods.

The result of normalizing flows would be affected by many hyperparameters, for example type of flows, the number of flows, and structure of the neural network if using coupling flows. Generally speaking, the choice of type of flow depends substantially on the problem at hand. For example, planar and Sylvester flows are suitable for multimodal problems (Rezende & Mohamed 2015; Berg *et al.* 2018). Radial flow works well to modify probability density only around a reference point (Rezende & Mohamed 2015). Linear flow has concise formulae and is simple to implement, yet it presents unsatisfactory performance for complicated or multimodal problems due to its limited expressiveness (Kucukelbir *et al.* 2017; Tomczak & Welling 2017). Coupling flow is one of the most widely used flows architectures since it contains neural networks, such that the target distribution can be modelled in a flexible way (Dinh *et al.* 2015, 2017; Kingma & Dhariwal 2018). More detailed information about these flows can be found in Appendix B.

To illustrate the effect of different numbers of flows, in Fig. 12 we show two other normalizing flows tests for the synthetic example, which use 1 and 3 coupling flows respectively while keeping all the other hyperparameters unchanged as used previously. If we compare the original result using 6 coupling flows in Fig. 6 with these two results, the three mean velocity maps are similar to each other, although the lower velocity loop from the two results (especially in the 1 coupling flow result) in Fig. 12 seems less symmetric. The standard deviation map in the left column of Fig. 12 fails to provide two higher uncertainty loops as we observed previously. We can vaguely see two (less smooth) higher uncertainty loops from the result using three coupling flows, yet the map is less accurate compared to the result in Fig. 6. From these two tomographic tests we conclude that by increasing the number of flows, we end up with a more accurate approximation to the posterior distribution (Rezende & Mohamed (2015) reached the same conclusion based on several 2-D examples). Thus when using normalizing flows to solve Bayesian inference problems (at least for tomographic problems), we suggest to use coupling flow to build a flows-based model. To improve accuracy, it would be wise to first increase the number of flows rather than increasing the neural network complexity, given that tuning a many-parameter neural network can itself be difficult.

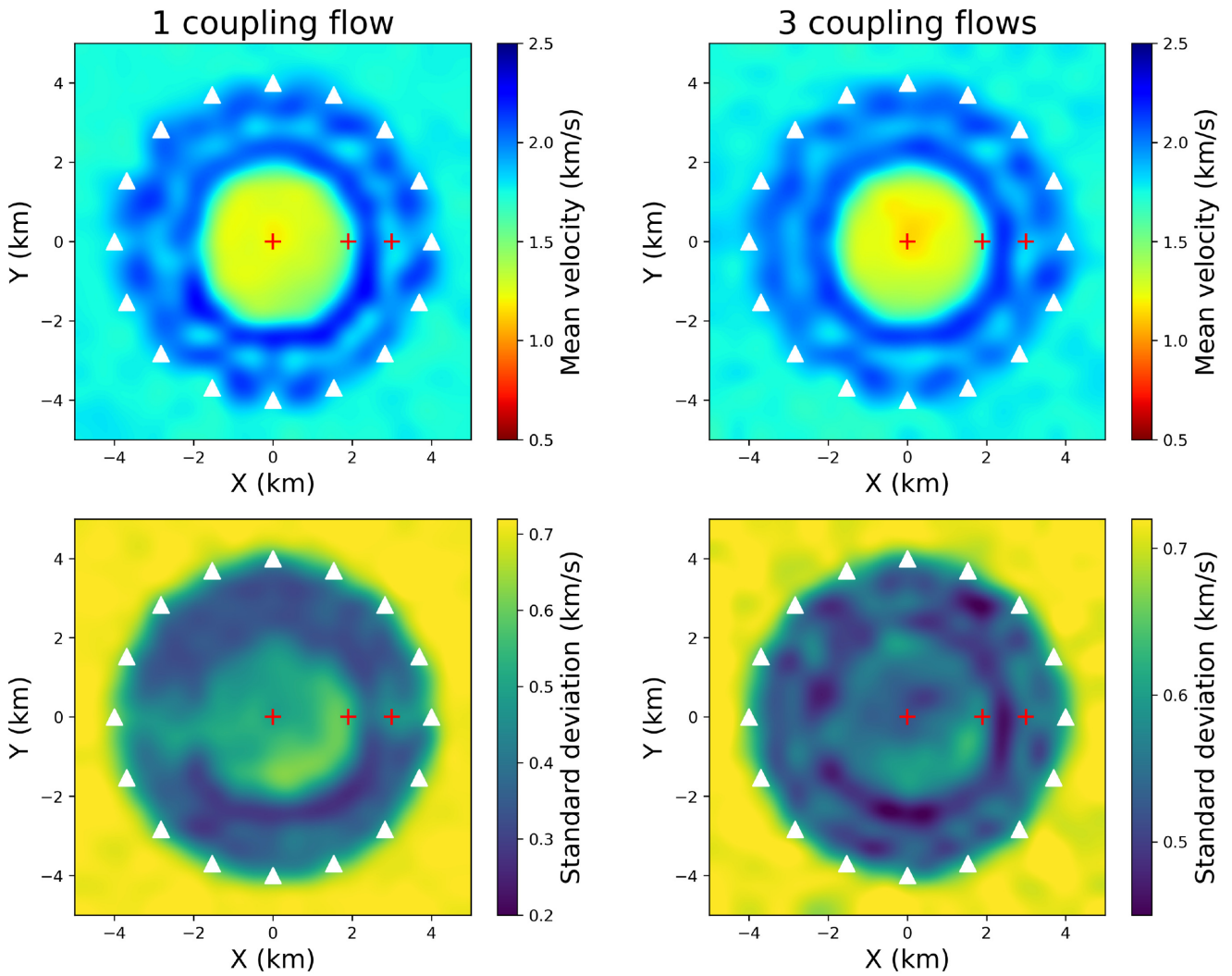

Figure 12. The mean (top row) and standard deviation (bottom row) of the posterior distributions using normalizing flows with 1 coupling flow (left-hand column) and 3 coupling flows (right-hand column), respectively. White triangles show the 16 receiver (virtual source) locations.

Although normalizing flows needs elaborate design for real applications, and the results seem to be less smooth compared to ADVI, SVGD and McMC, the method still provides an attractive approach to solve Bayesian inference problems due to the analyticity of its posterior solution. This is different from other sample based methods like McMC and SVGD: for a sample based method, the statistical properties (e.g. the mean and standard deviation) of the posterior distribution are calculated using an ensemble of posterior samples; we may fail to accurately evaluate a high-dimensional distribution with thousands of samples due to the curse of dimensionality. At present we still describe the posterior distribution of normalizing flows using statistics calculated as for SVGD and McMC, by drawing samples from the initial distribution and transforming them through normalizing flows to obtain posterior samples. We have not found an appropriate way to obtain analytical expressions of the posterior marginal pdfs, but it is intuitive that such a solution might be obtained by some kind of integration over the initial distribution through the trajectory of the normalizing flows. The analyticity of the solution is a promising research direction for future research to eliminate the sampling step to calculate marginals. We believe such a solution would also be useful for decision-making during seismic data interpretation by answering specific questions about the subsurface using interrogation theory since finding such answers usually reduces to calculating high-dimensional integrals over the posterior pdf (Arnold & Curtis 2018). At least for now, we can efficiently generate as many new posterior samples as we desire using normalizing flows once we have finished the training process and without evaluating the forward function, whereas the same thing is impossible for SVGD.

Normalizing flows provides a general mechanism to define expressive probability distributions (Papamakarios *et al.* 2021; Zhao *et al.* 2020; Siahkoohi *et al.* 2020), and has received much attention since the method was proposed. The main purpose of this paper is to introduce the method to readers in geophysics to solve geophysical problems. Future work might target other geophysical inversion problems to test the method's efficiency.

5 CONCLUSION

In this paper, we solve 2-D probabilistic traveltime tomography using normalizing flows under the framework of variational inference, which significantly improves the efficiency of Bayesian inversion by using optimization. The method transforms a simple and analytically known distribution into an approximation of the posterior distribution by applying a chain of invertible transforms. We first prove the accuracy and efficiency of normalizing flows for tomographic problems using a simple 2-D synthetic test, where normalizing flows is the most efficient method that approximates a correct uncertainty result compared with MH-McMC and two other variational methods: ADVI and SVGD. We also perform Love wave tomography to construct group velocity maps of the British Isles, in which normalizing flows give convincing average velocity and standard deviation maps that are consistent with the known geology and with previous research in this area. The flows provide nearly the same result compared to SVGD and MH-McMC, obviously outperforming ADVI for uncertainty estimation, while the computational cost is significantly reduced compared to other methods. This example shows the ability of normalizing flows to solve high-dimensional and complicated inference problems with real data. What is more, normalizing flows provides an analytic solution for the posterior distribution, which may provide a feasible and promising way to interrogate that solution in future.

ACKNOWLEDGEMENTS

We thank Edinburgh Imaging Project (EIP) sponsors (Schlumberger Foundation, BP and Total) for supporting this research.

6 DATA AVAILABILITY

Data associated with this work are available at British Geological Survey (http://www.earthquakes.bgs.ac.uk/data/data_archive.html).

REFERENCES

Aki, K., Christoffersson, A. & Husebye, E. S., 1977. Determination of the three-dimensional seismic structure of the lithosphere, *J. geophys. Res.,* **82**(2), 277–296.

Allmark, C., Curtis, A., Galetti, E. & de Ridder, S., 2018. Seismic attenuation from ambient noise across the north sea ekofisk permanent array, *J. geophys. Res.,* **123**(10), 8691–8710.

Anderssen, R. & Seneta, E., 1971. A simple statistical estimation procedure for Monte Carlo inversion in geophysics, *Pure appl. Geophys.,* **91**(1), 5–13.

Araya-Polo, M., Jennings, J., Adler, A. & Dahlke, T., 2018. Deep-learning tomography, *Leading Edge,* **37**(1), 58–66.

Arnold, R. & Curtis, A., 2018. Interrogation theory, *Geophys. J. Int.,* **214**(3), 1830–1846.

Bensen, G., Ritzwoller, M. & Shapiro, N. M., 2008. Broadband ambient noise surface wave tomography across the United States, *J. geophys. Res.,* **113**(B5), doi:10.1029/2007JB005248.

Berg, R. V. D., Hasenclever, L., Tomczak, J. M. & Welling, M., 2018. Sylvester normalizing flows for variational inference, in *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence 2018, (UAI 2018),* Association For Uncertainty in Artificial Intelligence (AUAI), pp. 393–402.

Bianco, M. J. & Gerstoft, P., 2018. Travel time tomography with adaptive dictionaries, *IEEE Trans. Comput. Imaging,* **4**(4), 499–511.

Bishop, C. M., 2006. *Pattern Recognition and Machine Learning,* Springer.

Blei, D. M., Kucukelbir, A. & McAuliffe, J. D., 2017. Variational inference: a review for statisticians, *J. Am. Stat. Assoc.,* **112**(518), 859–877.

Bodin, T. & Sambridge, M., 2009. Seismic tomography with the reversible jump algorithm, *Geophys. J. Int.,* **178**(3), 1411–1436.

Bodin, T., Sambridge, M., Tkalčić, H., Arroucau, P., Gallagher, K. & Rawlinson, N., 2012. Transdimensional inversion of receiver functions and surface wave dispersion, *J. geophys. Res.,* **117**(B2), doi:10.1029/2011JB008560.

Campillo, M. & Paul, A., 2003. Long-range correlations in the diffuse seismic coda, *Science,* **299**(5606), 547–549.

Carbonetto, P., Stephens, M., *et al.*, 2012. Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies, *Bayesian Anal.,* **7**(1), 73–108.

Chen, T. Q., Rubanova, Y., Bettencourt, J. & Duvenaud, D. K., 2018. Neural ordinary differential equations, in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018),* Montréal, Canada, pp. 6571–6583.

Curtis, A., Gerstoft, P., Sato, H., Snieder, R. & Wapenaar, K., 2006. Seismic interferometry – turning noise into signal, *Leading Edge,* **25**(9), 1082–1092.

Curtis, A. & Lomax, A., 2001. Prior information, sampling distributions, and the curse of dimensionality, *Geophysics,* **66**(2), 372–378.

Curtis, A., Nicolson, H., Halliday, D., Trampert, J. & Baptie, B., 2009. Virtual seismometers in the subsurface of the earth from seismic interferometry, *Nat. Geosci.,* **2**(10), 700–704.

Curtis, A. & Snieder, R., 2002. Probing the earth's interior with seismic tomography, *Int. Geophys. Ser.,* **81**(A), 861–874.

Curtis, A., Trampert, J., Snieder, R. & Dost, B., 1998. Eurasian fundamental mode surface wave phase velocities and their relationship with tectonic structures, *J. geophys. Res.,* **103**(B11), 26 919–26 947.

De Cao, N., Aziz, W. & Titov, I., 2019. Block neural autoregressive flow, in *Uncertainty in Artificial Intelligence,* pp.1263–-1273, PMLR.

de Ridder, S., Biondi, B. & Clapp, R., 2014. Time-lapse seismic noise correlation tomography at Valhall, *Geophys. Res. Lett.,* **41**(17), 6116–6122.

de Ridder, S. & Dellinger, J., 2011. Ambient seismic noise eikonal tomography for near-surface imaging at valhall, *Leading Edge,* **30**(5), 506–512.

de Wit, R. W., Valentine, A. P. & Trampert, J., 2013. Bayesian inference of earth's radial seismic structure from body-wave traveltimes using neural networks, *Geophys. J. Int.,* **195**(1), 408–422.

Devilee, R., Curtis, A. & Roy-Chowdhury, K., 1999. An efficient, probabilistic neural network approach to solving inverse problems: Inverting surface wave velocities for Eurasian crustal thickness, *J. geophys. Res.,* **104**(B12), 28 841–28 857.

Dinh, L., Krueger, D. & Bengio, Y., 2015. Nice: non-linear independent components estimation, preprint (arXiv:1410.8516).

Dinh, L., Sohl-Dickstein, J. & Bengio, S., 2017. Density estimation using real NVP, preprint (arXiv:1605.08803).

Durkan, C., Bekasov, A., Murray, I. & Papamakarios, G., 2019a. Cubic-spline flows, preprint (arXiv:1906.02145).

Durkan, C., Bekasov, A., Murray, I. & Papamakarios, G., 2019b. Neural spline flows, in *Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019),* Vancouver, Canada, pp. 7509–7520.

Dziewonski, A. M. & Woodhouse, J. H., 1987. Global images of the earth's interior, *Science,* **236**(4797), 37–48.

Earp, S. & Curtis, A., 2020. Probabilistic neural network-based 2D travel-time tomography, *Neural Comput. Appl.,* **32**(22), 17 077–17 095.

Earp, S., Curtis, A., Zhang, X. & Hansteen, F., 2020. Probabilistic neural network tomography across Grane field (North Sea) from surface wave dispersion data, *Geophys. J. Int.,* **223**(3), 1741–1757.

Fichtner, A. & Simutė, S., 2018. Hamiltonian monte carlo inversion of seismic sources in complex media, *J. geophys. Res.,* **123**(4), 2984–2999.

Fichtner, A., Zunino, A. & Gebraad, L., 2019. Hamiltonian Monte Carlo solution of tomographic inverse problems, *Geophys. J. Int.,* **216**(2), 1344–1363.

Galetti, E. & Curtis, A., 2012. Generalised receiver functions and seismic interferometry, *Tectonophysics,* **532,** 1–26.

Galetti, E. & Curtis, A., 2018. Transdimensional electrical resistivity tomography, *J. geophys. Res.,* **123**(8), 6347–6377.

Galetti, E., Curtis, A., Baptie, B., Jenkins, D. & Nicolson, H., 2017. Transdimensional love-wave tomography of the British Isles and shear-velocity structure of the east Irish Sea Basin from ambient-noise interferometry, *Geophys. J. Int.,* **208**(1), 36–58.

Galetti, E., Curtis, A., Meles, G. A. & Baptie, B., 2015. Uncertainty loops in travel-time tomography from nonlinear wave physics, *Phys. Rev. Lett.,* **114**(14), 148501,.

Gebraad, L., Boehm, C. & Fichtner, A., 2020. Bayesian elastic full-waveform inversion using Hamiltonian Monte Carlo, *J. geophys. Res.,* **125**(3), e2019JB018428.

Geyer, C. J. & Thompson, E. A., 1995. Annealing Markov Chain Monte Carlo with applications to ancestral inference, *J. Am. Stat. Assoc.,* **90**(431), 909–920.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y., 2014. Generative adversarial nets, in *NIPS'14: Proceedings of the 27th International Conference on Neural Information Processing Systems,* Vol. 2, pp. 2672–2680.

Gorbatov, A., Widiyantoro, S., Fukao, Y. & Gordeev, E., 2000. Signature of remnant slabs in the north pacific from P-wave tomography, *Geophys. J. Int.,* **142**(1), 27–36.

Green, P. J., 1995. Reversible jump Markov Chain Monte Carlo computation and Bayesian model determination, *Biometrika,* **82**(4), 711–732.

Green, P. J., 2003. Trans-dimensional Markov Chain Monte Carlo, in *Highly Structured Stochastic Systems,* Oxford Statistical Science Series, pp. 179–198, eds Green, P.J., Hjort, N.L. & Richardson, S., Oxford Univ. Press.

Green, P. J. & Mira, A., 2001. Delayed rejection in reversible jump Metropolis–Hastings, *Biometrika,* **88**(4), 1035–1053.

Gregory, J. & Delbourgo, R., 1982. Piecewise rational quadratic interpolation to monotonic data, *IMA J. Numer. Anal.,* **2**(2), 123–130.

Hastings, W. K., 1970. Monte Carlo sampling methods using Markov chains and their applications, *Biometrika,* **57**(1), 97–109.

Ho, J., Chen, X., Srinivas, A., Duan, Y. & Abbeel, P., 2019. Flow++: improving flow-based generative models with variational dequantization and architecture design, in *International Conference on Machine Learning,* pp. 2722–-2730, PMLR.

Hoffman, M. D. & Gelman, A., 2014. The no-u-turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo, *J. Mach. Learn. Res.,* **15**(1), 1593–1623.

Hoogeboom, E., Berg, R. V. D. & Welling, M., 2019. Emerging convolutions for generative normalizing flows, in *International Conference on Machine Learning,* pp. 2771–-2780, PMLR.

Huang, C.-W., Krueger, D., Lacoste, A. & Courville, A., 2018. Neural autoregressive flows, in *International Conference on Machine Learning,* pp.2078–-2087, PMLR.

Inoue, H., Fukao, Y., Tanabe, K. & Ogata, Y., 1990. Whole mantle P-wave travel time tomography, *Phys. Earth planet. Inter.,* **59**(4), 294–328.

Iyer, H. & Hirahara, K., 1993. *Seismic Tomography: Theory And Practice,* Springer Science & Business Media.

Jaini, P., Selby, K. A. & Yu, Y., 2019. Sum-of-squares polynomial flow, in *International Conference on Machine Learning,* pp. 3009–-3018, PMLR.

Käufl, P., Valentine, A. P., O'Toole, T. B. & Trampert, J., 2014. A framework for fast probabilistic centroid-moment-tensor determination – inversion of regional static displacement measurements, *Geophys. J. Int.,* **196**(3), 1676–1693.

Käufl, P., Valentine, A., de Wit, R. & Trampert, J., 2015. Robust and fast probabilistic source parameter estimation from near-field displacement waveforms using pattern recognition, *Bull. seism. Soc. Am.,* **105**(4), 2299–2312.

Khoshkholgh, S., Zunino, A. & Mosegaard, K., 2020. Informed proposal Monte Carlo, *Geophys. J. Int.,* **226,** 1239–-1248.

Kingma, D. P. & Dhariwal, P., 2018. Glow: generative flow with invertible 1x1 convolutions, in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018),* Montréal, Canada, pp. 10 215–10 224.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I. & Welling, M., 2016. Improved variational inference with inverse autoregressive flow, in *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016),* Barcelona, Spain., pp. 4743–4751.

Kingma, D. P. & Welling, M., 2014. Auto-encoding variational Bayes, *stat,* **1050,** 10.

Kobyzev, I., Prince, S. & Brubaker, M., 2019. Normalizing flows: an introduction and review of current methods, in *IEEE Transactions on Pattern Analysis and Machine Intelligence,* IEEE.

Kong, Q., Trugman, D. T., Ross, Z. E., Bianco, M. J., Meade, B. J. & Gerstoft, P., 2019. Machine learning in seismology: turning data into insights, *Seismol. Res. Lett.,* **90**(1), 3–14.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A. & Blei, D. M., 2017. Automatic differentiation variational inference, *J. Mach. Learn. Res.,* **18**(1), 430–474.

Kullback, S. & Leibler, R. A., 1951. On information and sufficiency, *Ann. Math. Stat.,* **22**(1), 79–86.

Likas, A. C. & Galatsanos, N. P., 2004. A variational approach for Bayesian blind image deconvolution, *IEEE Trans. Signal Process.,* **52**(8), 2222–2233.

Liu, Q., Lee, J. & Jordan, M., 2016. A kernelized stein discrepancy for goodness-of-fit tests, *Proc. Mach. Learn. Res.,* **48,** 276–284.

Liu, Q. & Wang, D., 2016. Stein variational gradient descent: a general purpose Bayesian inference algorithm, in *Proceedings of the 30th Conference on Neural Information Processing Systems (NIPS 2016),* Barcelona, Spain., pp. 2378–2386.

Loris, I., Nolet, G., Daubechies, I. & Dahlen, F., 2007. Tomographic inversion using l1-norm regularization of wavelet coefficients, *Geophys. J. Int.,* **170**(1), 359–370.

Malinverno, A., 2002. Parsimonious Bayesian Markov Chain Monte Carlo inversion in a nonlinear geophysical problem, *Geophys. J. Int.,* **151**(3), 675–688.

Meier, U., Curtis, A. & Trampert, J., 2007a. Fully nonlinear inversion of fundamental mode surface waves for a global crustal model, *Geophys. Res. Lett.,* **34**(16), doi:10.1029/2007GL030989.

Meier, U., Curtis, A. & Trampert, J., 2007b. Global crustal thickness from neural network inversion of surface wave data, *Geophys. J. Int.,* **169**(2), 706–722.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E., 1953. Equation of state calculations by fast computing machines, *J. Chem. Phys.,* **21**(6), 1087–1092.

Mordret, A., Landès, M., Shapiro, N. M., Singh, S. & Roux, P., 2014. Ambient noise surface wave tomography to determine the shallow shear velocity structure at Valhall: depth inversion with a neighbourhood algorithm, *Geophys. J. Int.,* **198**(3), 1514–1525.

Mordret, A., Landès, M., Shapiro, N. M., Singh, S., Roux, P. & Barkved, O., 2013. Near-surface study at the Valhall oil field from ambient noise surface wave tomography, *Geophys. J. Int.,* **193**(3), 1627–1643.

Mosegaard, K. & Tarantola, A., 1995. Monte Carlo sampling of solutions to inverse problems, *J. geophys. Res.,* **100**(B7), 12 431–12 447.

Moya, A. & Irikura, K., 2010. Inversion of a velocity model using artificial neural networks, *Comput. Geosci.,* **36**(12), 1474–1483.

Muir, J. B. & Tkalcic, H., 2015. Probabilistic joint inversion of lowermost mantle P-wave velocities and core mantle boundary topography using differential travel times and hierarchical Hamiltonian Monte-Carlo sampling, *AGUFM,* **2015,** S14A–03.

Müller, T., Mcwilliams, B., Rousselle, F., Gross, M. & Novák, J., 2018. Neural importance sampling, *ACM Trans. Graphics (TOG),* **38**(5), 1–19.

Nawaz, A. & Curtis, A., 2018. Variational Bayesian inversion (VBI) of quasi-localized seismic attributes for the spatial distribution of geological facies, *Geophys. J. Int.,* **214**(2), 845–875.

Nawaz, A. & Curtis, A., 2019. Rapid discriminative variational Bayesian inversion of geophysical data for the spatial distribution of geological properties, *J. geophys. Res.,* **124**(6), 5867–5887.

Nawaz, A., Curtis, A., Shahraeeni, M. S. & Gerea, C., 2020. Variational Bayesian inversion of seismic attributes jointly for geological facies and petrophysical rock properties, *Geophysics,* **85**(4), 1–78.

Neal, R. M. 2011. MCMC using Hamiltonian dynamics, in *Handbook of Markov Chain Monte Carlo,* eds Brooks, S., Gelman, A., Jones, G.L., Meng, X.-L., CRC Press.

Nicolson, H., Curtis, A. & Baptie, B., 2014. Rayleigh wave tomography of the British Isles from ambient seismic noise, *Geophys. J. Int.,* **198**(2), 637–655.

Nicolson, H., Curtis, A., Baptie, B. & Galetti, E., 2012. Seismic interferometry and ambient noise tomography in the British Isles, *Proc. Geol. Assoc.,* **123**(1), 74–86.

O'Hagan, A. & Forster, J. J., 2004. *Kendall's Advanced Theory of Statistics,* 2nd edn, Vol. 2B: Bayesian Inference, Arnold.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S. & Lakshminarayanan, B., 2021. Normalizing flows for probabilistic modeling and inference, *J. Mach. Learn. Res.,* **22,** 1–-64.

Papamakarios, G., Pavlakou, T. & Murray, I., 2017. Masked autoregressive flow for density estimation, in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems,* pp. 2338–2347.

Press, F., 1968. Earth models obtained by monte carlo inversion, *J. geophys. Res.,* **73**(16), 5223–5234.

Rawlinson, N., Pozgay, S. & Fishwick, S., 2010. Seismic tomography: a window into deep earth, *Phys. Earth planet. Inter.,* **178**(3–4), 101–135.

Rawlinson, N. & Sambridge, M., 2005. The fast marching method: an effective tool for tomographic imaging and tracking multiple phases in complex layered media, *Exploration Geophysics,* **36**(4), 341–350.

Rawlinson, N., Sambridge, M. & Saygin, E., 2008. A dynamic objective function technique for generating multiple solution models in seismic tomography, *Geophys. J. Int.,* **174**(1), 295–308.

Rezende, D. & Mohamed, S., 2015. Variational inference with normalizing flows, in *International conference on machine learning,* pp. 1530–-1538, PMLR.

Roberts, S. J. & Penny, W. D., 2002. Variational Bayes for generalized autoregressive models, *IEEE Trans. Signal Process.,* **50**(9), 2245–2257.

Röth, G. & Tarantola, A., 1994. Neural networks and inversion of seismic data, *J. geophys. Res.,* **99**(B4), 6753–6768.

Sabra, K. G., Gerstoft, P., Roux, P., Kuperman, W. & Fehler, M. C., 2005. Surface wave tomography from microseisms in southern California, *Geophys. Res. Lett.,* **32**(14), doi:10.1029/2005GL023155.

Sambridge, M., 1999. Geophysical inversion with a neighbourhood algorithm—I. Searching a parameter space, *Geophys. J. Int.,* **138**(2), 479–494.

Sen, M. K. & Biswas, R., 2017. Transdimensional seismic inversion using the reversible jump Hamiltonian Monte Carlo algorithm, *Geophysics,* **82**(3), R119–R134.

Shahraeeni, M. S. & Curtis, A., 2011. Fast probabilistic nonlinear petrophysical inversion, *Geophysics,* **76**(2), E45–E58.

Shahraeeni, M. S., Curtis, A. & Chao, G., 2012. Fast probabilistic petrophysical mapping of reservoirs from 3D seismic data, *Geophysics,* **77**(3), O1–O19.

Shapiro, N. M., Campillo, M., Stehly, L. & Ritzwoller, M. H., 2005. High-resolution surface-wave tomography from ambient seismic noise, *Science,* **307**(5715), 1615–1618.

Shapiro, N. M. & Ritzwoller, M., 2002. Monte-Carlo inversion for a global shear-velocity model of the crust and upper mantle, *Geophys. J. Int.,* **151**(1), 88–105.

Siahkoohi, A., Rizzuti, G., Witte, P. A. & Herrmann, F. J., 2020. Faster uncertainty quantification for inverse problems with conditional normalizing flows, preprint (arXiv:2007.07985).

Simons, F. J., Van Der Hilst, R. D., Montagner, J.-P. & Zielhuis, A., 2002. Multimode Rayleigh wave inversion for heterogeneity and azimuthal anisotropy of the Australian upper mantle, *Geophys. J. Int.,* **151**(3), 738–754.

Spakman, W., 1991. Delay-time tomography of the upper mantle below Europe, the Mediterranean, and Asia minor, *Geophys. J. Int.,* **107**(2), 309–332.

Tarantola, A., 2005. *Inverse Problem Theory and Methods for Model Parameter Estimation,* Vol. **89,** SIAM.

Thurber, C. H., 1983. Earthquake locations and three-dimensional crustal structure in the coyote lake area, central California, *J. geophys. Res.,* **88**(B10), 8226–8236.

Tomczak, J. M. & Welling, M., 2016. Improving variational auto-encoders using householder flow, preprint (arXiv:1611.09630).

Tomczak, J. M. & Welling, M., 2017. Improving variational auto-encoders using convex combination linear inverse autoregressive flow, in *Benelearn 2017: Proceedings of the 26th Benelux Conference on Machine Learning, Technische Universiteit Eindhoven,* pp. 162.

Trampert, J. & Woodhouse, J. H., 1995. Global phase velocity maps of love and Rayleigh waves between 40 and 150 seconds, *Geophys. J. Int.,* **122**(2), 675–690.

Villasenor, A., Yang, Y., Ritzwoller, M. H. & Gallart, J., 2007. Ambient noise surface wave tomography of the Iberian Peninsula: implications for shallow seismic structure, *Geophys. Res. Lett.,* **34**(11), doi:10.1029/2007GL030164.

Walker, M. & Curtis, A., 2014. Spatial Bayesian inversion with localized likelihoods: an exact sampling alternative to MCMc, *J. geophys. Res.,* **119**(7), 5741–5761.

Wapenaar, K., Draganov, D., Snieder, R., Campman, X. & Verdel, A., 2010a. Tutorial on seismic interferometry: Part 1 – basic principles and applications, *Geophysics,* **75**(5), 75A195–75A209.

Wapenaar, K. & Fokkema, J., 2006. Green's function representations for seismic interferometry, *Geophysics,* **71**(4), SI33–SI46.

Wapenaar, K., Slob, E., Snieder, R. & Curtis, A., 2010b. Tutorial on seismic interferometry: Part 2 – underlying theory and new advances, *Geophysics,* **75**(5), 75A211–75A227.

Wolpert, D. H. & Macready, W. G., 1997. No free lunch theorems for optimization, *IEEE Trans. Evolut. Comput.,* **1**(1), 67–82.

Zhang, X. & Curtis, A., 2020a. Seismic tomography using variational inference methods, *J. geophys. Res.,* **125**(4), e2019JB018589.

Zhang, X. & Curtis, A., 2020b. Variational full-waveform inversion, *Geophys. J. Int.,* **222**(1), 406–411.

Zhang, X., Curtis, A., Galetti, E. & De Ridder, S., 2018. 3-D Monte Carlo surface wave tomography, *Geophys. J. Int.,* **215**(3), 1644–1658.

Zhang, X., Nawaz, M. A.,Zhao, X.& Curtis, A.,2021. An introduction to variational inference in geophysical inverse problems, in *Advances in Geophysics,* Elsevier, available at: https://www.sciencedirect.com/science/article/pii/S0065268721000030.

Zhang, X., Roy, C., Curtis, A., Nowacki, A. & Baptie, B., 2020. Imaging the subsurface using induced seismicity and ambient noise: 3D tomographic Monte Carlo joint inversion of earthquake body wave travel times and surface wave dispersion, *Geophys. J. Int.,* **222**(3), 1639–1655.

Zhao, X., Curtis, A. & Zhang, X., 2020. Bayesian seismic tomography using normalizing flows, *Earth,* preprint (arXiv:10.31223/X53K6G).

Zhdanov, M. S., 2002. *Geophysical Inverse Theory and Regularization Problems,* Vol. **36,** Elsevier.

Zheng, X., Jiao, W., Zhang, C. & Wang, L., 2010. Short-period Rayleigh-wave group velocity tomography through ambient noise cross-correlation in Xinjiang, northwest China, *Bull. seism. Soc. Am.,* **100**(3), 1350–1355.

Ziegler, Z. & Rush, A., 2019. Latent normalizing flows for discrete sequences, in *International Conference on Machine Learning,* pp. 7673–-7682, PMLR.

APPENDIX A: DERIVATION OF $\mathcal{L}[q_K(\mathbf{m}_K)]$

This appendix provides a derivation of eq. (8). We start with a general function $h(\mathbf{m}_K)$ and take its expectation with respect to $q_K(\mathbf{m}_K)$. Using the flows formula in eq. (6), we obtain

$$\begin{aligned}\mathbb{E}_{q_K(\mathbf{m}_K)}[h(\mathbf{m}_K)] &= \int q_K(\mathbf{m}_K)h(\mathbf{m}_K)d\mathbf{m}_K \\ &= \int q_0(\mathbf{m}_0)h(\mathbf{m}_K)d\mathbf{m}_0 \\ &= \mathbb{E}_{q_0(\mathbf{m}_0)}[h(\mathbf{m}_K)],\end{aligned} \tag{A1}$$

where the step to the second line invokes the implicit relationship between $\mathbf{m}_0$ and $\mathbf{m}_K$ in eq. (6). Eq. (A1) implies that the expectation of $h(\mathbf{m}_K)$ with respect to the transformed pdf $q_K(\mathbf{m}_K)$ can be computed without explicitly knowing $q_K(\mathbf{m}_K)$ itself when $h(\mathbf{m}_K)$ does not depend on $q_K(\mathbf{m}_K)$ (Rezende & Mohamed 2015). Thus $\mathcal{L}[q_K(\mathbf{m}_K)]$ can be rewritten as

$$\begin{aligned}\mathcal{L}_\Theta[q_K(\mathbf{m}_K)] &= \mathbb{E}_{q_K(\mathbf{m}_K)}[\log p(\mathbf{m}_K, \mathbf{d}_{obs})] - \mathbb{E}_{q_K(\mathbf{m}_K)}[\log q_K(\mathbf{m}_K)] \\ &= \mathbb{E}_{q_0(\mathbf{m}_0)}[\log p(\mathbf{m}_K, \mathbf{d}_{obs})] - \mathbb{E}_{q_0(\mathbf{m}_0)}[\log q_0(\mathbf{m}_0)] + \mathbb{E}_{q_0(\mathbf{m}_0)}\left[\log\left|\det\frac{\partial F_\Theta}{\partial \mathbf{m}_0}\right|\right]\end{aligned} \tag{A2}$$

which can be iteratively maximized using gradient-based optimization methods by calculating the gradient of $\mathcal{L}_\Theta[q_K(\mathbf{m}_K)]$ with respect to the normalizing flows parameter Θ:

$$\nabla_\Theta\mathcal{L} = \mathbb{E}_{q_0(\mathbf{m}_0)}\left[\nabla_{\mathbf{m}_K}(\log p(\mathbf{m}_K, \mathbf{d}_{obs}))\nabla_\Theta\mathbf{m}_K + \nabla_\Theta\log\left|\det\frac{\partial F_\Theta}{\partial \mathbf{m}_0}\right|\right]. \tag{A3}$$

Here the term $\nabla_{\mathbf{m}_K}(\log p(\mathbf{m}_K, \mathbf{d}_{obs}))$ stands for the conventional logarithmic data-model gradient calculated in linearized inversion. Compared to linearized inversion, the optimization process of normalizing flows model only needs additional gradient information about the flows parameters, which can be analytically calculated by elaborate design of the flows structure (see Appendix B).

APPENDIX B: WAYS TO CONSTRUCT NORMALIZING FLOWS

B1 Rational quadratic splines

In this section, we introduce one specific bijection function used throughout the examples in this paper—rational quadratic splines (Durkan *et al.* 2019b). Fig. B1 shows an illustration of an element-wise monotonic increasing rational quadratic spline that transforms the input element

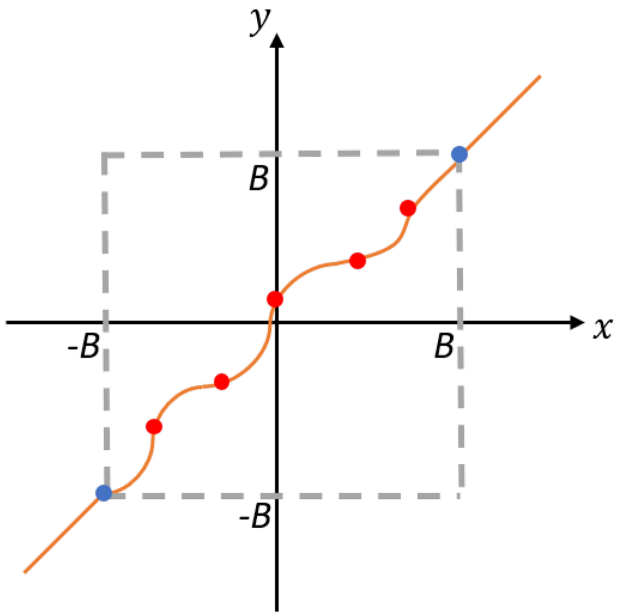

Figure B1. An illustration of monotonic increasing rational quadratic spline that transforms x to y across the real domain. The spline is divided into 8 pieces by 7 knots (2 blue and 5 red knots). The 2 pieces outside the interval $[-B, B]$ are identity functions, and the inner 6 bins are non-linear rational quadratic functions obtained by interpolation (Gregory & Delbourgo 1982). The 2 blue knots have coordinates of $[\pm B, \pm B]$ and derivative values of 1, and the coordinates and the derivatives of the 5 red knots are to be learned during optimization. Therefore, we need to learn 15 parameters (10 of which for width and height of each bin (implicitly determined by the coordinates of the 5 red knots) and the rest 5 for the derivatives at the inner red knots) to fully parametrize the spline throughout the real domain.

x to the output element y. The spline maps a predefined interval [-B, B] to [-B, B] non-linearly, and is defined to be the identity function outside this region, resulting in linear tails in Fig. B1 such that the overall spline is monotonic across the real domain. Inside [-B, B], the spline is parametrized as L different rational quadratic functions using rational quadratic interpolation (Gregory & Delbourgo 1982), and is separated by $L+1$ boundary knots $\{(x^{(l)}, y^{(l)})\}_{l=0}^{L}$ (2 blue and 5 red knots in Fig. B1, in which we set $L=6$). Those $L+1$ knots monotonically increase between points $(x^{(0)}, y^{(0)}) = (-B, -B)$ and $(x^{(L)}, y^{(L)}) = (B, B)$, and the coordinates of the inner $L-1$ (red) knots—which determine the width and height of each bin—are parameters to be learned during optimization.

Let $\{\delta^{(l)}\}_{l=0}^{L}$ be the derivatives $\frac{\partial y}{\partial x}$ at the $L+1$ knots, respectively. The derivatives at $\pm B$ are fixed to be $\delta^{(0)} = \delta^{(L)} = 1$ to match the linear tails outside [-B, B] (see Fig. B1). For the inside $L-1$ knots, we set their derivatives $\{\delta^{(l)}\}_{l=1}^{L-1}$ as learnable parameters with positive values to ensure the continuity of $\frac{\partial y}{\partial x}$ in the real domain. If the derivatives within L bins are not matched in this way, the transform is still continuous, but its derivative can have jump discontinuities at the spline boundary points. This in turn makes the log-likelihood training objective discontinuous, which would make the optimization fail (Durkan *et al.* 2019b).

Given $3(L-1)$ learnable parameters mentioned above, where $2(L-1)$ of them stand for the coordinates of the inner knots (which determine the widths and heights of the L bins) and $L-1$ of them for the derivative values of the inner $L-1$ knots, we could fully parametrize the rational quadratic spline in the real domain, so as to transform the input element x into y, evaluate its inverse map from y to x and the derivative $\frac{\partial y}{\partial x}$). Define

$$\begin{aligned} s^{(l)} &= \frac{y^{(l+1)} - y^{(l)}}{x^{(l+1)} - x^{(l)}} \\ \xi(x) &= \frac{x - x^{(l)}}{x^{(l+1)} - x^{(l)}} \end{aligned} \tag{B1}$$

such that the element-wise rational quadratic function within the lth bin can be interpolated by (Durkan *et al.* 2019b)

$$y = y^{(l)} + \frac{\left(y^{(l+1)} - y^{(l)}\right)\left[s^{(l)}\xi^2 + \delta^{(l)}\xi(1-\xi)\right]}{s^{(l)} + \left[\delta^{(l+1)} + \delta^{(l)} - 2s^{(l)}\right]\xi(1-\xi)} \tag{B2}$$

and the derivative within the lth bin by

$$\frac{\partial y}{\partial x} = \frac{\left(s^{(l)}\right)^2\left[\delta^{(l+1)}\xi^2 + 2s^{(l)}\xi(1-\xi) + \delta^{(l)}(1-\xi)^2\right]}{\left[s^{(l)} + \left(\delta^{(l+1)} + \delta^{(l)} - 2s^{(l)}\right)\xi(1-\xi)\right]^2}. \tag{B3}$$

The inverse of eq. (B2) can be obtained by calculating the root of a quadratic equation, which turns out to be $\xi(x) = 2c/\left(-b - \sqrt{b^2 - 4ac}\right)$ with

$$\begin{aligned} a &= \left(y^{(l+1)} - y^{(l)}\right)\left[s^{(l)} - \delta^{(l)}\right] + \left(y - y^{(l)}\right)\left[\delta^{(l+1)} + \delta^{(l)} - 2s^{(l)}\right] \\ b &= \left(y^{(l+1)} - y^{(l)}\right)\delta^{(l)} - \left(y - y^{(l)}\right)\left[\delta^{(l+1)} + \delta^{(l)} - 2s^{(l)}\right] \\ c &= -s^{(l)}\left(y - y^{(l)}\right) \end{aligned} \tag{B4}$$

which can be used to determine the inverse map of the lth spline—the value of x given y.

For rational quadratic splines based coupling flows, we transform each element in $\mathbf{m}_i^B$ to that in $\mathbf{m}_{i+1}^B$ using one specific rational quadratic spline, so the neural network has input vector of d parameters and output vector of $3(L-1)\times(D-d)$ parameters, which is used to fully parametrize $(D-d)$ rational quadratic splines, each corresponding to one element in $\mathbf{m}_i^B$. The output $3(L-1)$ parameters of each element are further divided into three subvectors, each containing $L-1$ parameters. The first two partitions are interpreted as the widths and heights of the L bins (denoted as Θ^w and Θ^h), and are passed through a softmax activation function (each parameter is defined as $sm(\Theta_l^{w,h}) = \frac{\exp(\Theta_l^{w,h})}{\sum_{l=1}^{L}\exp(\Theta_l^{w,h})}$, where $\Theta_l^{w,h,d}$ denotes lth element in subvector $\Theta^{w,h,d}$) and multiplied by $2B$, such that the parameters have positive values and summed to $2B$, and can be interpreted as the width and height of each bin. The last partition (denoted as Θ^d) is passed through a softplus function (each parameter is defined as $sp(\Theta_l^d) = \ln(1+\exp(\Theta_l^d))$), such that the parameters have positive values and can be interpreted as $L-1$ positive derivatives $\{\delta_i^{(l)}\}_{l=1}^{L-1}$ at the inner $L-1$ knots. Then the coupling flows can be parametrized using eqs (B2)–(B4).

B2 Linear flow

Linear flow has the general form

$$\mathbf{m}_{i+1} = \mathbf{A}\mathbf{m}_i + \mathbf{b}, \tag{B5}$$

where $\mathbf{A} \in \mathbb{R}^{D\times D}$ and $\mathbf{b} \in \mathbb{R}^D$ are flow parameters. If $\mathbf{A}$ is an invertible matrix, the transform is itself invertible. Its Jacobian determinant is simply $\det(\mathbf{A})$, and by making some restrictions on the structure of $\mathbf{A}$, it can be calculated efficiently.

B2.1 Diagonal flow

If $\mathbf{A}$ is a diagonal matrix with non-zero diagonal entries, the forward transform and the Jacobian determinant of the linear flow can be calculated within linear times ($\mathcal{O}(D)$). However, this results in an element-wise transform and expresses no correlation between different dimensions.

B2.2 Triangular flow

If $\mathbf{A}$ is a triangular matrix with non-zero diagonal entries, the correlation between dimensions are included while the Jacobian determinant remains easy to evaluate. Automatic differential variational inference (ADVI—Kucukelbir *et al.* 2017; Zhang & Curtis 2020a) can be viewed as a triangular flow that transforms a standard Gaussian distribution into any form within the Gaussian family. Tomczak & Welling (2017) constructed a linear flow by adding M triangular matrices $\mathbf{A}$ in weight, such that the flow function in eq. (B5) becomes $\mathbf{m}_{i+1} = \left(\sum_{j=1}^{M} w_j \mathbf{A}_j\right) \mathbf{m}_i$, where $\sum_{j=1}^{M} w_j = 1$. Each of the triangular matrices has 1 on the diagonal such that the composite flow is volume-preserving (the Jacobian determinant equals to 1).

B2.3 Matrix decomposition

Instead of limiting the specific form of $\mathbf{A}$, many normalizing flows are based on matrix decomposition to decompose $\mathbf{A}$ into a product of structured matrices, each of which has easily calculated Jacobian determinant. For example, Tomczak & Welling (2016) used a Householder transform to model an orthogonal matrix $\mathbf{A}$ which led to a volume-preserving flow. LU decomposition (Kingma & Dhariwal 2018) and QR decomposition (Hoogeboom *et al.* 2019) are used to model a general matrix $\mathbf{A}$ with easily calculated Jacobian determinant.

B3 Planar and radial flows

Rezende & Mohamed (2015) derived two invertible and differentiable normalizing flows: *Planar and Radial flows*. Planar flow is defined as

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \mathbf{u}h(\mathbf{w}^T \mathbf{m}_i + b) \tag{B6}$$

and is used to expand or contract a distribution along the specific hyperplane $\mathbf{w}^T \mathbf{m}_i + b = 0$. Vectors $\mathbf{u}, \mathbf{w} \in \mathbb{R}^D$ and $b \in \mathbb{R}$ are flow parameters. h is a smooth and differentiable function, and Rezende & Mohamed (2015) suggested to use $h(x) = \tanh(x)$ to ensure invertibility. Using the matrix determinant lemma, the Jacobian determinant can be calculated within $\mathcal{O}(D)$ time by (Rezende & Mohamed 2015):

$$\det \frac{\partial \mathbf{m}_{i+1}}{\partial \mathbf{m}_i} = 1 + \mathbf{u}^T h'(\mathbf{w}^T \mathbf{m}_i + b)\mathbf{w}. \tag{B7}$$

Planar flow can be interpreted as a neural network that contains one hidden layer and one hidden neural (Kingma *et al.* 2016), and expressiveness is obtained by stacking many planar flows in series. Berg *et al.* (2018) proposed *Sylvester flow* as an improvement of planar flow:

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \mathbf{U}h(\mathbf{W}^T \mathbf{m}_i + \mathbf{b}), \tag{B8}$$

where $\mathbf{U}$ and $\mathbf{W}$ are $D \times M$ matrices and $b \in \mathbb{R}^M$. h is an element-wise differentiable function. $1 \leq M \leq D$ is a predefined hyperparameter and can be interpreted as the dimensionality of a hidden layer in a neural network.

Radial flow can be written as

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \frac{\beta}{\alpha + \|\mathbf{m}_i - \mathbf{m}'\|}(\mathbf{m}_i - \mathbf{m}'), \tag{B9}$$

where $\alpha > 0$, $\beta \in \mathbb{R}$ and $\mathbf{m}' \in \mathbb{R}^D$ are flow parameters. Radial flow is used to reshape a distribution around a reference point $\mathbf{m}'$ (for example radial contraction and expansion around the reference point). The Jacobian determinant can be evaluated by

$$\det \frac{\partial \mathbf{m}_{i+1}}{\partial \mathbf{m}_i} = \left(1 + \frac{\beta}{\alpha + \|\mathbf{m}_i - \mathbf{m}'\|}\right)^{D-1} \left(1 + \frac{\alpha\beta}{(\alpha + \|\mathbf{m}_i - \mathbf{m}'\|)^2}\right). \tag{B10}$$

Note that the invertibility of these three flows can only be guaranteed under some specific conditions (Rezende & Mohamed 2015; Berg *et al.* 2018), and there is no explicit expression for the inverse transform.

B4 Coupling and autoregressive flows

In the main text, we have discussed coupling flow, and now we introduce *autoregressive flow*—another special flow structure that has easily calculated Jacobian determinant. Autoregressive flow was proposed by Kingma *et al.* (2016) for variational inference and by Papamakarios *et al.* (2017) for density estimation. As shown in Fig. B2, for each element $m_{i,j}$ in the input vector $\mathbf{m}_i$, the neural network inputs all the previous elements $\mathbf{m}_{i,1:j-1}$, and the output is used to construct an element-wise bijection f, such that

$$m_{i+1,j} = f(m_{i,j}; NN(\mathbf{m}_{i,1:j-1})). \tag{B11}$$

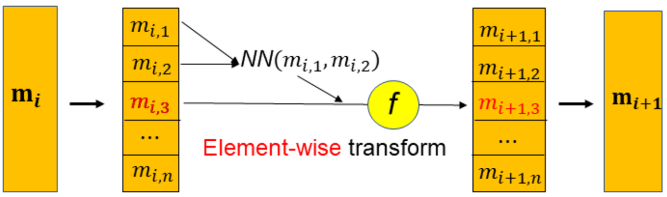

Figure B2. Structure of autoregressive flow. The input vector $\mathbf{m}_i$ is evaluated element-wise. For each of the elements, for example the third element $m_{i,3}$ in this figure, we input all its previous elements ($m_{i,1}$ and $m_{i,2}$ in this case) into a neural network, and its output is used to construct an element-wise function f. This function is further used to transform $m_{i,3}$ into $m_{i+1,3}$. The same procedure can be applied to all of the other elements in $\mathbf{m}_i$, such that we obtain the transformed vector $\mathbf{m}_{i+1}$.

Based on the structure of autoregressive flow, we can obtain a lower triangular Jacobian matrix with $\frac{\partial m_{i+1,j}}{\partial m_i}|_{j=1}^{D}$ on diagonal entries. Then the Jacobian determinant can be calculated by

$$\det \frac{\partial \mathbf{m}_{i+1}}{\partial \mathbf{m}_i} = \prod_{j=1}^{D} \frac{\partial m_{i+1,j}}{\partial m_{i,j}} \tag{B12}$$

and the inverse transform of autoregressive flow is

$$m_{i,j} = f^{-1}(m_{i+1,j}; NN(\mathbf{m}_{1:i-1})). \tag{B13}$$

Coupling and autoregressive flows are the two most popular structures for normalizing flows due to their efficiency for calculating the Jacobian determinant. On the other hand, compared to the matrix based flows which have concise formulae but unsatisfactory performance for high dimensional problems, the expressiveness of these two flows can be guaranteed by elaborate design of the element-wise function f. In addition to the rational quadratic splines used in the main text, we now introduce other kinds of the bijection functions.

B4.1 Affine

The original papers that proposed coupling flow (Dinh *et al.* 2015) and autoregressive flow (Kingma *et al.* 2016; Papamakarios *et al.* 2017) used the affine transform

$$m_{i+1,j} = \sigma m_{i,j} + \mu, \tag{B14}$$

where σ and μ are the output of the neural network in coupling and autoregressive flows. The inverse and Jacobian determinant of eq. (B14) is easy to calculate. Dinh *et al.* (2017) and Kingma & Dhariwal (2018) modified affine based coupling flow by introducing random permutation and 1×1 convolution to change the element order of the input vector, such that the flow performance is improved for image generation. Although the affine function is simple and efficient, it is hard to use to model complex distributions.

B4.2 Splines

Müller *et al.* (2018) first proposed to use several monotonic piecewise linear and quadratic splines to model the bijection. Durkan *et al.* (2019a) extended this work by using the cubic splines for the bijection, and permuting the elements of the input vector by LU-decomposition. Durkan *et al.* (2019b) further introduced the rational quadratic splines, and demonstrated that the proposed splines significantly enhance the flexibility of both coupling and autoregressive flows for variational inference and density estimation, and in some cases bring the performance of coupling flow on par with the best-known autoregressive flow.

B4.3 Neural autoregressive flow

Huang *et al.* (2018) introduced *neural autoregressive flow*. In this flow structure, another neural network is introduced to mimic the effect of the bijection f that inputs $m_{i,j}$ and outputs $m_{i+1,j}$. In this network, all the weights need to be positive and the activation functions to be strictly monotonic to ensure the invertibility of f. De Cao *et al.* (2019) further proposed *block neural autoregressive flow* to improve the efficiency of neural autoregressive flow. The deficiency of these flows is that though they are theoretically invertible, evaluating their inverses is quite difficult.

B4.4 Others

Ho *et al.* (2019) introduced *flow++* to use a cumulative density function to modify a linear transform; Ziegler & Rush (2019) used non-linear squared transform; and Jaini *et al.* (2019) modelled a monotonic increasing function f by using the sum of several squared polynomials so as to approximate any univariate continuous function. For details of these works, we suggest readers refer to the original papers.

B5 Continuous flows

In the previous discussion, normalizing flows are constructed by combining several discrete one-step transforms in series. In this section, we transform the initial distribution towards the target through a continuous trajectory. This kind of normalizing flow is called *continuous flow*. Assume $\mathbf{m}_t$ is the model vector state at time t, $\mathbf{m}_{t_0}$ is the model parameter under the initial distribution and $\mathbf{m}_{t_T}$ is that under the target distribution. Then the evolution of $\mathbf{m}_t$ through t can be determined by

$$\frac{d\mathbf{m}_t}{dt} = f(t, \mathbf{m}_t), \tag{B15}$$

where f is a function of both time t and model parameter $\mathbf{m}_t$, and denotes the change of $\mathbf{m}_t$ through time. Model vector $\mathbf{m}_{t_T}$ can be calculated by solving this ordinary differential equation as

$$\mathbf{m}_{t_T} = \mathbf{m}_{t_0} + \int_{t=t_0}^{t_T} f(t, \mathbf{m}_t)dt. \tag{B16}$$

The corresponding inverse transform is

$$\mathbf{m}_{t_0} = \mathbf{m}_{t_T} - \int_{t=t_0}^{t_T} f(t, \mathbf{m}_t)dt. \tag{B17}$$

So the forward and inverse transforms of continuous flow have the same computational cost and complexity. In addition, unlike the above discrete flows that use the Jacobian determinant to characterize the volume change of a transform, the change of variables formula for continuous flow is (Chen *et al.* 2018)

$$\frac{d \log p(\mathbf{m}_t)}{dt} = -\mathrm{Tr}\left(\frac{d f(t, \mathbf{m}_t)}{d\mathbf{m}_t}\right), \tag{B18}$$

where $\mathrm{Tr}(\cdot)$ is the trace operator of the Jacobian matrix, whose computation is far more efficient than the determinant operator.

APPENDIX C: DERIVATION OF JACOBIAN DETERMINANT FOR COUPLING FLOW

For coupling flow, based on eq. (13), the determinant of the Jacobian matrix can be evaluated in blocks:

$$\det \frac{\partial \mathbf{m}_{i+1}}{\partial \mathbf{m}_i} = \det \begin{pmatrix} \frac{\partial \mathbf{m}_{i+1}^A}{\partial \mathbf{m}_i^A} & \frac{\partial \mathbf{m}_{i+1}^A}{\partial \mathbf{m}_i^B} \\ \frac{\partial \mathbf{m}_{i+1}^B}{\partial \mathbf{m}_i^A} & \frac{\partial \mathbf{m}_{i+1}^B}{\partial \mathbf{m}_i^B} \end{pmatrix} = \det \frac{\partial \mathbf{m}_{i+1}^B}{\partial \mathbf{m}_i^B}. \tag{C1}$$

From Fig. 3, it is obvious that $\frac{\partial \mathbf{m}_{i+1}^A}{\partial \mathbf{m}_i^A} = \mathbf{I}$ and $\frac{\partial \mathbf{m}_{i+1}^A}{\partial \mathbf{m}_i^B} = \mathbf{0}$, so the right-hand side of eq. (C1) holds. What is more, since we use an element-wise function f to transform each element of $\mathbf{m}_i^B$ into $\mathbf{m}_{i+1}^B$, $\frac{\partial \mathbf{m}_{i+1}^B}{\partial \mathbf{m}_i^B}$ is actually a diagonal matrix, which leads to eq. (14) in the main text.