Title

Sequential Kalman tuning of the t-preconditioned Crank-Nicolson algorithm: efficient, adaptive and gradient-free inference for Bayesian inverse problems

Permalink

https://escholarship.org/uc/item/8ph0t1q7

Journal

Inverse Problems, 40(12)

ISSN

0266-5611

Authors

Grumitt, Richard DP

Karamanis, Minas

Seljak, Uroš

Publication Date

2024-12-01

DOI

10.1088/1361-6420/ad934b

Sequential Kalman Tuning of the t-preconditioned Crank-Nicolson algorithm: efficient, adaptive and gradient-free inference for Bayesian inverse problems

Richard D.P. Grumitt[1], **Minas Karamanis**[2,3], **Uroš Seljak**[2,3]

[1]Department of Astronomy, Tsinghua University, Beijing 100084, China

[2]Berkeley Center for Cosmological Physics and Department of Physics, University of California, Berkeley, CA 94720

[3]Physics Department, Lawrence Berkeley National Laboratory, Cyclotron Rd, Berkeley, CA 94720

E-mail: `rgrumitt@mail.tsinghua.edu.cn`

June 2024

Abstract. Ensemble Kalman Inversion (EKI) has been proposed as an efficient method for the approximate solution of Bayesian inverse problems with expensive forward models. However, when applied to the Bayesian inverse problem EKI is only exact in the regime of Gaussian target measures and linear forward models. In this work we propose embedding EKI and Flow Annealed Kalman Inversion (FAKI), its normalizing flow (NF) preconditioned variant, within a Bayesian annealing scheme as part of an adaptive implementation of the t-preconditioned Crank-Nicolson (tpCN) sampler. The tpCN sampler differs from standard pCN in that its proposal is reversible with respect to the multivariate t-distribution. The more flexible tail behaviour allows for better adaptation to sampling from non-Gaussian targets. Within our Sequential Kalman Tuning (SKT) adaptation scheme, EKI is used to initialize and precondition the tpCN sampler for each annealed target. The subsequent tpCN iterations ensure particles are correctly distributed according to each annealed target, avoiding the accumulation of errors that would otherwise impact EKI. We demonstrate the performance of SKT for tpCN on three challenging numerical benchmarks, showing significant improvements in the rate of convergence compared to adaptation within standard SMC with importance weighted resampling at each temperature level, and compared to similar adaptive implementations of standard pCN. The SKT scheme applied to tpCN offers an efficient, practical solution for solving the Bayesian inverse problem when gradients of the forward model are not available. Code implementing the SKT schemes for tpCN is available at `https://github.com/RichardGrumitt/KalmanMC`.

Keywords: Inverse Problems, Bayesian Inference, Ensemble Kalman Inversion, Sequential Monte Carlo, Normalizing Flows

Submitted to: *Inverse Problems*

1. Introduction

Many scientific inference tasks can be viewed within the Bayesian inverse problem framework. In the Gaussian inverse problem setting, we can write the forward problem as

$$\boldsymbol{y} = \mathcal{F}(\boldsymbol{x}) + \boldsymbol{\eta}, \tag{1}$$

where $\boldsymbol{y} \in \mathbb{R}^{n_y}$ is the data vector, $\mathcal{F}$ is a forward model that maps the parameters $\boldsymbol{x} \in \mathbb{R}^d$ to our observables, and $\boldsymbol{\eta} \sim \mathcal{N}(0, \Gamma)$ is additive Gaussian noise with fixed noise covariance $\Gamma \in \mathbb{R}^{n_y \times n_y}$. For the Bayesian inverse problem we assign some prior over the parameters $\boldsymbol{x} \sim \pi_0(\boldsymbol{x})$, with the goal then being to recover the posterior distribution

$$\pi(\boldsymbol{x}|\boldsymbol{y}) = \frac{\pi(\boldsymbol{y}|\boldsymbol{x})\pi_0(\boldsymbol{x})}{\mathcal{Z}}, \tag{2}$$

where $\mathcal{Z}$ is some generally unknown normalizing constant and $\pi(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{y}|\mathcal{F}(\boldsymbol{x}), \Gamma)$ [1, 2].

The particular regime we are concerned with for this work is where we do not have access to gradients of a typically expensive forward model. This is a common setting for scientific inverse problems, where evaluating the forward model often involves running some black-box solver for which gradients cannot be easily and/or accurately obtained e.g., cosmological Boltzmann solvers [3, 4], computational fluid dynamics simulators [5], etc. Non-differentiable forward models can also be a result of inherently discontinuous physics, e.g., in cloud modelling [6]. Given the forward problem definition in Equation 1, we are restricted to Bayesian inference tasks with Gaussian likelihoods. However, this still encompasses a large number of scientific inverse problems, and is the regime for which the Ensemble Kalman methods we exploit in this work have been developed [7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18]. In addition to this, we are concerned with developing methods that can reliably obtain low bias estimates of posterior moments. This is critical for many scientific inference tasks where we require accurate uncertainty quantification on any model parameter constraints.

A typical approach to solving Bayesian inverse problems involves exploiting some form of sampling algorithm. This covers a wide range of methods, e.g., Markov Chain Monte Carlo (MCMC) algorithms [19, 20, 21, 22], simulating interacting particle systems [23, 24, 25, 26, 27, 28] etc. In MCMC algorithms, we seek to construct some transition, $T(\boldsymbol{x}', \boldsymbol{x})$ that preserves the target, $\pi(\boldsymbol{x}|\boldsymbol{y})$ as an invariant distribution, i.e.,

$$\pi(\boldsymbol{x}|\boldsymbol{y}) = \int T(\boldsymbol{x}', \boldsymbol{x})\pi(\boldsymbol{x}'|\boldsymbol{y})\,\mathrm{d}\boldsymbol{x}'. \tag{3}$$

Appropriately constructed, such methods enjoy target invariance and ergodicity properties. However, especially in the gradient-free regime we consider in this work, this often comes at the cost of requiring $\gtrsim \mathcal{O}(10^4)$ serial model evaluations [16], quickly rendering such algorithms intractable for expensive and high dimensional models.

An alternative class of method involves constructing some coupling scheme, where we have a transition $C(\boldsymbol{x}', \boldsymbol{x})$, that moves us from the prior, $\pi_0(\boldsymbol{x})$ to the target $\pi(\boldsymbol{x}|\boldsymbol{y})$, i.e.,

$$\pi(\boldsymbol{x}|\boldsymbol{y}) = \int C(\boldsymbol{x}', \boldsymbol{x})\pi_0(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}. \tag{4}$$

Examples of coupling methods include the Ensemble Kalman Filter (EKF) [29, 30, 31], Ensemble Kalman Inversion (EKI) [7, 10, 11, 12, 13, 17], and Sequential Monte Carlo (SMC) [32, 33, 34, 35].

In addition to developing methods for solving Bayesian inverse problems, it is crucial to consider adaptation strategies that allow for efficient, tuning-free implementations of these methods that can used by practitioners. Extensive work has been done on the development of tuning-free implementations of gradient-based algorithms such as Hamiltonian Monte Carlo (HMC) [36, 37, 38, 39, 40], including in the context of SMC [41]. For gradient-free algorithms a notable adaptive method is Preconditioned Monte Carlo (PMC), implemented in the PocoMC library [35, 42], which uses normalizing flow (NF) [43, 44, 45, 46] preconditioning within sequential Monte Carlo (SMC) to accelerate gradient-free sampling.

1.1. Our Contributions

- We develop an adaptive, tuning-free implementation of the t-preconditioned Crank-Nicolson (tpCN) algorithm, designed for performing efficient gradient-free inference in Bayesian inverse problems. The tpCN algorithm preserves the exact target distribution as its invariant measure, allowing for accurate posterior moment estimation when faced with non-Gaussian targets and nonlinear forward models, which is critical for scientific inference tasks. Compared to the standard pCN algorithm, the tpCN algorithm is found to have significantly improved performance on non-Gaussian targets.
- Our adaptive scheme exploits the natural connection between EKI and Bayesian annealing approaches, by using EKI within an SMC sampling scheme. Controlling the transition between temperature levels as we move from the prior to the posterior in SMC allows us to apply EKI updates treating the target at the previous temperature level as an effective prior. EKI then provides a highly effective initialization and preconditioner for the tpCN updates. The tpCN updates help to ensure we correctly converge on the target at each temperature level, preventing the accumulation of errors that would result from applying EKI alone within an annealing scheme to the Bayesian inverse problem.
- We demonstrate the empirical performance of our adaptive sampling scheme on three challenging inverse problem benchmarks. We show that tpCN significantly outperforms standard pCN. We also show that the use of EKI as an initialization and preconditioner within a Bayesian annealing scheme for the exact tpCN updates yields significant performance improvements compared to using the standard

importance resampling step in SMC.

The structure of the paper is as follows: In Section 2 we describe the tpCN sampling algorithm we propose developing an adaptive implementation of in this work. In Section 3 we describe essential background regarding the methods used for adapting the tpCN sampler. In Section 4 we describe the Sequential Kalman Tuning (SKT) adaptation scheme for the tpCN algorithm, and its NF preconditioned variant NF-SKT, proposed in this work for rapid gradient-free Bayesian inference. In Section 5 we present numerical results comparing the performance of the adaptive SKT samplers against adaptation in standard SMC using importance weighted resampling, and we conclude in Section 6. Code implementing the adaptive SKT samplers presented in this work is available at `https://github.com/RichardGrumitt/KalmanMC`.

2. t-preconditioned Crank-Nicolson Algorithm

In this work we consider the adaptation of the tpCN sampling algorithm within SMC, which has been implemented in the context of NF preconditioned SMC in the pocoMC sampling package‡. At its core, tpCN modifies the standard pCN proposal such that it is reversible with respect to the multivariate t-distribution, as opposed to the multivariate Gaussian distribution for the pCN proposal. In [47] the mixed preconditioned Crank-Nicolson (MpCN) algorithm was proposed, which uses a proposal that is reversible with respect to the σ-finite measure $\bar{p}(\mathrm{d}\boldsymbol{x}) = \|\boldsymbol{x}\|_2^{-d}\,\mathrm{d}\boldsymbol{x}$. Detailed theoretical studies of the MpCN algorithm were performed in [48, 47], which showed improved convergence results for MpCN on heavy tailed targets compared to pCN. However, in our own numerical studies we found that the MpCN algorithm could not be easily adapted for sampling on the non-Gaussian targets we consider in this work. Whilst the base t-distribution in tpCN can be adapted for each target, adjusting the corresponding tail behaviour of the proposal, the base distribution of MpCN is not so readily adaptable. Even after pre-whitening of the target, we found the MpCN acceptance rate was typically close to zero. We therefore do not consider it further as a numerical benchmark in this work. A similarly detailed theoretical study of the tpCN algorithm as in [48, 47] is beyond the scope of this work, where we focus on its practicable adaptive implementation. However, we do show that the tpCN algorithm has superior empirical performance compared to standard pCN on a range of challenging benchmarks, when allowing for similar adaptation in their sampling hyper-parameters. In the remainder of this section we describe the pCN and tpCN algorithms.

2.1. pCN algorithm

Consider some target measure with probability density function (PDF) $p(\boldsymbol{x})$. The standard pCN algorithm generates samples from the target by iterating over the procedure described in Algorithm 1. We denote the PDF of the multivariate Gaussian

‡ `https://github.com/minaskar/pocomc/`

distribution at some location $\boldsymbol{x}$ as $\varphi_{\mathcal{N}}(\boldsymbol{x};\boldsymbol{\mu},\mathcal{C})$, where $\boldsymbol{\mu}$ is the Gaussian mean and $\mathcal{C}$ is the Gaussian covariance.

Algorithm 1 pCN update

```
**Input:** Current particle location $\boldsymbol{x}_{m-1}$, pCN proposal reference mean $\boldsymbol{\mu}$, pCN proposal reference covariance $\mathcal{C}$, pCN step size $\rho$, target density $p(\boldsymbol{x})$.
Draw $\boldsymbol{W}_m \sim \mathcal{N}(0,\mathcal{C})$.
$\boldsymbol{x}'_m = \boldsymbol{\mu} + \sqrt{1-\rho^2}(\boldsymbol{x}_{m-1} - \boldsymbol{\mu}) + \rho\boldsymbol{W}_m$.
Update particle location,
    $\boldsymbol{x}_m = \begin{cases} \boldsymbol{x}'_m, & \text{with probability}\, \alpha(\boldsymbol{x}'_m, \boldsymbol{x}_{m-1}), \\ \boldsymbol{x}_{m-1}, & \text{with probability}\, 1-\alpha(\boldsymbol{x}'_m, \boldsymbol{x}_{m-1}), \end{cases}$   (5)
    where
    $\alpha(\boldsymbol{x}'_m, \boldsymbol{x}_{m-1}) = \min\left\{1, \frac{p(\boldsymbol{x}'_m)\varphi_{\mathcal{N}}(\boldsymbol{x}_{m-1};\boldsymbol{\mu},\mathcal{C})}{p(\boldsymbol{x}_{m-1})\varphi_{\mathcal{N}}(\boldsymbol{x}'_m;\boldsymbol{\mu},\mathcal{C})}\right\}.$   (6)
**Output**: New particle location $\boldsymbol{x}_m$.
```

The pCN step size parameter, ρ controls the extent to which a proposal sample is correlated with the previous sample. In the limit where $\rho \to 1$, the pCN proposal reduces to an independent proposal drawn from $\mathcal{N}(\boldsymbol{\mu},\mathcal{C})$. The proposal kernel for pCN as defined in Algorithm 1 is given by

$$\mathcal{K}(\boldsymbol{x},\mathrm{d}\boldsymbol{x}') = \mathcal{N}(\boldsymbol{\mu} + \sqrt{1-\rho^2}(\boldsymbol{x}-\boldsymbol{\mu}), \rho^2\mathcal{C}), \tag{7}$$

which is reversible with respect to the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu},\mathcal{C})$ i.e.,

$$\varphi_{\mathcal{N}}(\boldsymbol{x};\boldsymbol{\mu},\mathcal{C})\mathrm{d}\boldsymbol{x}\mathcal{K}(\boldsymbol{x},\mathrm{d}\boldsymbol{x}') = \varphi_{\mathcal{N}}(\boldsymbol{x}';\boldsymbol{\mu},\mathcal{C})\mathrm{d}\boldsymbol{x}'\mathcal{K}(\boldsymbol{x}',\mathrm{d}\boldsymbol{x}). \tag{8}$$

The pCN algorithm has been shown to exhibit a dimension independent spectral gap for large class of target measures which are the finite dimensional approximations of densities defined with respect to some Gaussian reference measure i.e., for some target posterior measure π we have the Radon-Nikodym derivative

$$\frac{\mathrm{d}\pi}{\mathrm{d}\pi_0}(\boldsymbol{x}) \propto \exp(-\Phi(\boldsymbol{x})), \tag{9}$$

where the reference prior measure π_0 is taken to be the Gaussian $\mathcal{N}(\boldsymbol{\mu},\mathcal{C})$ and $\Phi(\boldsymbol{x})$ is the likelihood potential [22, 49]. The pCN algorithm performs well when the target measure is close to Gaussian. However, for non-Gaussian targets and targets with heavy tails the performance of the algorithm can be severely degraded. Indeed, in [47] it was shown that pCN performs worse than Random Walk Metropolis-Hastings (RWHM) on a family of heavy tailed targets. This presents a problem for many scientific inference tasks where the target distribution can be expected to show some degree of non-Gaussianity.

2.2. tpCN algorithm

To develop an adaptive sampling scheme that will perform well against non-Gaussian targets we consider the tpCN algorithm. Instead of using a Gaussian base distribution to generate a proposal, as with standard pCN, the tpCN algorithm uses a multivariate t-distribution $t_{\nu_s}(\boldsymbol{\mu}_s, \mathcal{C}_s)$, where $\nu_s > 0$ denotes the degrees of freedom, $\boldsymbol{\mu}_s$ is the mean and $\mathcal{C}_s$ is the scale matrix. A simple non-adaptive variant of tpCN has previously been used in the estimation of drift and diffusion parameters for stochastic differential equations in [50]. The tpCN algorithm generates samples by iterating over the procedure in Algorithm 2. For brevity in the discussion below we use the inner product notation $\langle \boldsymbol{x}_1, \boldsymbol{x}_2 \rangle_s = (\boldsymbol{x}_1 - \boldsymbol{\mu}_s)^{\intercal} \mathcal{C}_s^{-1} (\boldsymbol{x}_2 - \boldsymbol{\mu}_s)$.

Algorithm 2 tpCN update

```
Input: Current particle location $\boldsymbol{x}_{m-1}$, $t$-distribution mean $\boldsymbol{\mu}_s$, $t$-distribution scale matrix $\mathcal{C}_s$, $t$-distribution degrees of freedom $\nu_s$, pCN step size $\rho$, target density $p(\boldsymbol{x})$.

Draw $Z_m^{-1} \sim \text{Gamma}(k = \frac{1}{2}(d + \nu_s), \theta = 2/(\nu_s + \langle \boldsymbol{x}_{m-1}, \boldsymbol{x}_{m-1} \rangle_s))$ and $\boldsymbol{W}_m \sim \mathcal{N}(0, \mathcal{C}_s)$.
$\boldsymbol{x}'_m = \boldsymbol{\mu}_s + \sqrt{1-\rho^2}(\boldsymbol{x}_{m-1} - \boldsymbol{\mu}_s) + \rho\sqrt{Z_m}\boldsymbol{W}_m$.
Update particle location,
    $\boldsymbol{x}_m = \begin{cases} \boldsymbol{x}'_m, & \text{with probability}\, \alpha(\boldsymbol{x}'_m, \boldsymbol{x}_{m-1}), \\ \boldsymbol{x}_{m-1}, & \text{with probability}\, 1 - \alpha(\boldsymbol{x}'_m, \boldsymbol{x}_{m-1}), \end{cases} \qquad (10)$
where
    $\alpha(\boldsymbol{x}'_m, \boldsymbol{x}_{m-1}) = \min\left\{1, \frac{p(\boldsymbol{x}'_m)(1 + \langle \boldsymbol{x}_{m-1}, \boldsymbol{x}_{m-1} \rangle_s / \nu_s)^{-(d+\nu_s)/2}}{p(\boldsymbol{x}_{m-1})(1 + \langle \boldsymbol{x}'_m, \boldsymbol{x}'_m \rangle_s / \nu_s)^{-(d+\nu_s)/2}}\right\}. \qquad (11)$
Output: New particle location $\boldsymbol{x}_m$.
```

It can be shown that the tpCN proposal is reversible with respect to the multivariate t-distribution $t_{\nu_s}(\boldsymbol{\mu}_s, \mathcal{C}_s)$. The reversibility and acceptance rate properties of the tpCN algorithm are stated in Lemma 2.1, with the corresponding proof given in Appendix A.

Lemma 2.1. *The proposal transition kernel of the tpCN algorithm is reversible with respect to the multivariate t-distribution $t_{\nu_s}(\boldsymbol{\mu}_s, \mathcal{C}_s)$ and the proposal acceptance probability is given by Equation 11.*

Similarly to the MpCN algorithm, the tpCN proposal is reversible with respect to a distribution that will generally have heavier tails than the standard pCN algorithm. One may therefore expect that it will show similarly improved performance on heavy tailed targets. A key difference between the tpCN and MpCN algorithms is the ability to tune the degrees of freedom ν_s, which controls the tail behaviour of the tpCN proposal.

However, it is worth emphasising that the benefits of using the t-distribution as a base distribution extend beyond heavy tailed targets to non-Gaussian targets more generally. The ability to tune the parameters of the more flexible t-distribution to the target allows for improved sampling of non-Gaussian targets, as observed in our numerical experiments in Section 5. The multivariate t-distribution has been exploited in the development of adaptive elliptical slice sampling implementations in [51], where the increased flexibility of the multivariate t-distribution in approximating the target distribution was found to yield significant performance gains compared to standard elliptical slice sampling.

For targets with strong non-Gaussianity, the performance of tpCN can be futher improved through NF preconditioning. In this case the tpCN updates are performed on the NF latent space particles, $\boldsymbol{z} = f(\boldsymbol{x})$ with the corresponding latent space acceptance probability being given by

$$\alpha(\boldsymbol{z}, \boldsymbol{z}') = \min\left\{1, \frac{\pi_0(f_n^{-1}(\boldsymbol{z}'))\pi(\boldsymbol{y}|f^{-1}(\boldsymbol{z}'))^{\beta_{n+1}}|\mathrm{det}Df^{-1}(\boldsymbol{z}')|(1+\langle \boldsymbol{z}, \boldsymbol{z}\rangle_s/\nu_s)^{-(d+\nu_s)/2}}{\pi_0(f^{-1}(\boldsymbol{z}))\pi(\boldsymbol{y}|f^{-1}(\boldsymbol{z}))^{\beta_{n+1}}|\mathrm{det}Df^{-1}(\boldsymbol{z})|(1+\langle \boldsymbol{z}', \boldsymbol{z}'\rangle_s/\nu_s)^{-(d+\nu_s)/2}}\right\}, \tag{12}$$

where $Df^{-1}(\boldsymbol{z}) = \partial f^{-1}(\boldsymbol{z})/\partial \boldsymbol{z}$ is the Jacobian of the inverse NF transformation. The use of NF preconditioning within our adaptation scheme is discussed in detail in Section 3.4.

3. Background Methods

In this section we introduce essential background regarding the methods we use for implementing an adaptive tpCN sampler, targeted at solving the Bayesian inverse problem. In Section 3.1 we give a brief description of EKI as applied to the Bayesian inverse problem and its connection to Bayesian annealing, in order to motivate its use within our adaptation scheme. In Section 3.2 we describe the SMC sampling scheme within which we embed our adaptation procedures. In Section 3.3 we describe the temperature schedule adaptation, and in Section 3.4 we describe the use of NFs for additional preconditioning.

3.1. Ensemble Kalman Inversion

EKI is a coupling-based algorithm that leverages ideas from EKF to construct iterative particle ensemble updates for the solution of inverse problems [7, 8, 9, 10, 11, 12, 13, 17]. In its standard setting, EKI seeks to solve the variational inverse problem i.e., finding parameter values that minimize the misfit functional

$$\Phi(\boldsymbol{x}) = \frac{1}{2}\left\|\Gamma^{-1/2}(\boldsymbol{y} - \mathcal{F}(\boldsymbol{x}))\right\|_2^2. \tag{13}$$

In this optimization setting, the EKI ensemble will collapse to a single solution [7], and can also be applied rigorously to problems with nonlinear forward models [17]. However, EKI has also been applied to the solution of the Bayesian inverse problem, where we seek to obtain an approximation to the full target posterior [13, 18]. In this setting, EKI is only exact when applied to linear forward models with Gaussian targets, providing an uncontrolled posterior approximation outside this regime.

In this work we consider leveraging the EKI target approximation within the context of Bayesian annealing to initialize and precondition tpCN sampling iterations. It is worth noting that EKI has previously been used in the context of preconditioning for sampling in [26], where the Ensemble Kalman Sampler (EKS) was developed based on the EKI ensemble structure. However, this sampling scheme is only exact for linear forward models, and can otherwise give highly biased posterior moment estimates. A key difference in our work is in the direct use of EKI updates to intialize and precondition the tpCN sampling scheme, which preserves the exact target as its invariant measure. For adaptive MCMC schemes based within Bayesian annealing this achieves significant convergence acceleration whilst allowing for accurate posterior moment estimation outside the regime of linear forward models and Gaussian targets. In Appendix G we demonstrate the performance of EKS on our numerical experiments, showing that it fails to recover accurate posterior moment estimates.

Following [13], we can motivate EKI applied to the solution of Bayesian inverse problems within the context of a Bayesian annealing scheme. Given a prior measure $\pi_0(\boldsymbol{x})$, EKI proceeds by constructing a sequence of Gaussian ensemble approximations to the intermediate measures

$$\pi_n(\mathrm{d}\boldsymbol{x}) \propto \pi_0(\mathrm{d}\boldsymbol{x})\mathcal{N}(\boldsymbol{y}|\mathcal{F}(\boldsymbol{x}), \beta_n^{-1}\Gamma), \tag{14}$$

where the inverse temperatures satisfy $0 \equiv \beta_0 < \beta_1 < \ldots < \beta_N < \beta_{N+1} \equiv 1$. From Equation 14 we can obtain the recursion

$$\frac{\pi_{n+1}(\mathrm{d}\boldsymbol{x})}{\pi_n(\mathrm{d}\boldsymbol{x})} \propto \mathcal{N}(\boldsymbol{y}|\mathcal{F}(\boldsymbol{x}), \alpha_n\Gamma), \tag{15}$$

where the annealing step size $\alpha_n = (\beta_{n+1} - \beta_n)^{-1}$. The step size can be viewed as a regularization parameter [8, 12, 13], which can be selected such that we make a gradual transition from the prior to the posterior.

The ensemble updates for EKI can be derived by assuming we have some Gaussian approximation to the prior measure $\tilde{\pi}_0 = \mathcal{N}(\boldsymbol{m}_0, C_0)$, proceeding to move through a sequence of Gaussian approximations, $\tilde{\pi}_n = \mathcal{N}(\boldsymbol{m}_n, C_n)$ using the recursion

$$\frac{\tilde{\pi}_{n+1}(\mathrm{d}\boldsymbol{x})}{\tilde{\pi}_n(\mathrm{d}\boldsymbol{x})} \propto \mathcal{N}\left(\boldsymbol{y}|\mathcal{F}(\boldsymbol{m}_n) + \frac{\partial\mathcal{F}_n}{\partial\boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{m}_n), \alpha_n\Gamma\right). \tag{16}$$

The forward model has been linearized around the approximation mean, $\mathcal{F}(\boldsymbol{x}) \approx \mathcal{F}(\boldsymbol{m}_n) - \partial\mathcal{F}_n/\partial\boldsymbol{x}(\boldsymbol{x} - \boldsymbol{m}_n)$, where $\partial\mathcal{F}_n/\partial\boldsymbol{x} = \partial\mathcal{F}/\partial\boldsymbol{x}|_{\boldsymbol{x}=\boldsymbol{m}_n}$. From Equation 16 we can obtain recursions for the approximation means and covariances,

$$\boldsymbol{m}_{n+1} = \boldsymbol{m}_n + C_n\frac{\partial\mathcal{F}_n^*}{\partial\boldsymbol{x}}\left(\frac{\partial\mathcal{F}_n}{\partial\boldsymbol{x}}C_n\frac{\partial\mathcal{F}_n^*}{\partial\boldsymbol{x}} + \alpha_n\Gamma\right)^{-1}(\boldsymbol{y} - \mathcal{F}(\boldsymbol{m}_n)), \tag{17}$$

$$C_{n+1} = C_n - C_n\frac{\partial\mathcal{F}_n^*}{\partial\boldsymbol{x}}\left(\frac{\partial\mathcal{F}_n}{\partial\boldsymbol{x}}C_n\frac{\partial\mathcal{F}_n^*}{\partial\boldsymbol{x}} + \alpha_n\Gamma\right)^{-1}\frac{\partial\mathcal{F}_n}{\partial\boldsymbol{x}}C_n, \tag{18}$$

where $\partial \mathcal{F}_n^*/\partial \boldsymbol{x}$ is the adjoint of $\partial \mathcal{F}_n/\partial \boldsymbol{x}$. Using the linearized forward model, the terms involving derivatives can be approximated as

$$C_n \frac{\partial \mathcal{F}_n^*}{\partial \boldsymbol{x}} \approx \mathbb{E}_n[(\boldsymbol{x}_n - \boldsymbol{m}_n) \otimes (\mathcal{F}(\boldsymbol{x}_n) - \mathbb{E}_n[\mathcal{F}(\boldsymbol{x}_n)])], \tag{19}$$

$$\frac{\partial \mathcal{F}_n}{\partial \boldsymbol{x}} C_n \frac{\partial \mathcal{F}_n^*}{\partial \boldsymbol{x}} \approx \mathbb{E}_n[(\mathcal{F}(\boldsymbol{x}_n) - \mathbb{E}_n[\mathcal{F}(\boldsymbol{x}_n)]) \otimes (\mathcal{F}(\boldsymbol{x}_n) - \mathbb{E}_n[\mathcal{F}(\boldsymbol{x}_n)])], \tag{20}$$

where $\mathbb{E}_n$ denotes the expectation with respect to $\tilde{\pi}_n$. These expectations cannot be computed in closed form. To overcome this, EKI exploits an ensemble approximation. Given an initial particle ensemble $\{\boldsymbol{x}_0^i \sim \pi_0(\boldsymbol{x})\}_{i=1}^J$, EKI applies embarrassingly parallel recursive updates using the expression

$$\boldsymbol{x}_{n+1}^i = \boldsymbol{x}_n^i + C_n^{\boldsymbol{x}\mathcal{F}} \left(C_n^{\mathcal{F}\mathcal{F}} + \alpha_n \Gamma \right)^{-1} \left(\boldsymbol{y} - \mathcal{F}(\boldsymbol{x}_n^i) + \sqrt{\alpha_n} \boldsymbol{\xi}_n^i \right), \tag{21}$$

where $\boldsymbol{\xi}_n^i \sim \mathcal{N}(0, \Gamma)$ is a Gaussian noise vector [13]. The empirical covariances are given by

$$C_n^{\boldsymbol{x}\mathcal{F}} = \frac{1}{J-1} \sum_{i=1}^{J} (\boldsymbol{x}_n^i - \langle \boldsymbol{x}_n \rangle) \otimes (\mathcal{F}(\boldsymbol{x}_n^i) - \langle \mathcal{F}_n \rangle), \tag{22}$$

$$C_n^{\mathcal{F}\mathcal{F}} = \frac{1}{J-1} \sum_{i=1}^{J} (\mathcal{F}(\boldsymbol{x}_n^i) - \langle \mathcal{F}_n \rangle) \otimes (\mathcal{F}(\boldsymbol{x}_n^i) - \langle \mathcal{F}_n \rangle), \tag{23}$$

where $\langle \boldsymbol{x}_n \rangle = \frac{1}{J} \sum_{i=1}^{J} \boldsymbol{x}_n^i$ and $\langle \mathcal{F}_n \rangle = \frac{1}{J} \sum_{i=1}^{J} \mathcal{F}(\boldsymbol{x}_n^i)$. It can be shown that the ensemble means and covariances obtained through the EKI updates approximate those in Equations 17 and 18 as $J \to \infty$ [52, 13].

Applied to the Bayesian inverse problem, EKI enjoys rapid convergence properties, typically converging in $\mathcal{O}(10)$ iterations [16]. For the case of Gaussian targets with linear forward models the particle ensemble will be distributed according to the target posterior as the ensemble size $J \to \infty$ [13], otherwise giving an uncontrolled approximation. In [18], NF maps were learned at each temperature level in the EKI iterations. By learning an NF map, one can map the particle distribution at a given temperature level to a Gaussian latent space and perform the EKI update in this latent space. Whilst this can improve the stability of EKI when faced with non-Gaussian targets, it does not address the linearity assumptions used in deriving EKI. Further, the NF map can introduce additional nonlinearity in the forward model evaluation due to the need to apply the inverse transformation when evaluating the forward model at each latent space location. These problems can result in the converged particle ensemble being a poor approximation to the true posterior, which poses a major drawback for scientific inference tasks where we desire accurate estimation of the first and second moments of the target posterior.

Despite the limitations of EKI when applied to the solution of the Bayesian inverse problem alone, we can exploit its natural connection with Bayesian annealing to form part of an adaptive SMC sampling scheme. The core idea here is that an EKI update can be used at each temperature level to target the next temperature level. The updated ensemble provides both an initialization and preconditioner for the subsequent sampling iterations.

3.2. Sequential Monte Carlo

SMC encompasses a class of sampling methods that move through a sequence of probability measures $\{\pi_n(\boldsymbol{x})\}_{n=1}^{N+1}$ in order to sample from the final target measure $\pi_{N+1}(\boldsymbol{x})$ [32]. The method has seen extensive applications in sequential Bayesian inference where one has a set of sequential observations $\{\boldsymbol{y}_t\}_{t=1}^{T}$ e.g., time series data [53]. In this case SMC moves through targets $\pi_n(\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{y}_1, ..., \boldsymbol{y}_n)$, where at each iteration an additional observation is added. Ensemble Kalman methods have previously been exploited for SMC in sequential Bayesian inference in [33], where the EKF update was used to construct an efficient importance sampling proposal for SMC.

An alternative setting involves moving from some tractable density $\pi_0(\boldsymbol{x})$, through a sequence of intermediate measures towards the final target. In this work, we consider the situation where $\pi_0(\boldsymbol{x})$ is the prior and we move through a sequence of temperature annealed targets $\pi_n(\boldsymbol{x}) \propto \pi_0(\boldsymbol{x})\pi(\boldsymbol{y}|\boldsymbol{x})^{\beta_n}$ where $\pi(\boldsymbol{y}|\boldsymbol{x})$ is the likelihood. As with EKI, the inverse temperatures satisfy $0 \equiv \beta_0 < \beta_1 < \ldots < \beta_N < \beta_{N+1} \equiv 1$, with $\beta_0 = 0$ corresponding to the prior and $\beta_{N+1} = 1$ corresponding to the full posterior.

Consider a particle ensemble at the inverse temperature β_n. Assuming the ensemble is distributed according to the annealed target $\pi_n(\boldsymbol{x}) \propto \pi_0(\boldsymbol{x})\pi(\boldsymbol{y}|\boldsymbol{x})^{\beta_n}$, we can calculate the unnormalized importance weights corresponding to the next temperature level,

$$w_n(\boldsymbol{x}_n^i) = \frac{\pi(\boldsymbol{y}|\boldsymbol{x}_n^i)^{\beta_{n+1}}}{\pi(\boldsymbol{y}|\boldsymbol{x}_n^i)^{\beta_n}}. \tag{24}$$

From this we obtain an estimator for expectations of test functions $f(\boldsymbol{x})$ with respect to the subsequent annealed target given by

$$\mathbb{E}_{\pi_{n+1}}[f(\boldsymbol{x})] = \frac{\sum_{i=1}^{J} f(\boldsymbol{x}_n^i) w_n(\boldsymbol{x}_n^i)}{\sum_{i=1}^{J} w_n(\boldsymbol{x}_n^i)}. \tag{25}$$

If the importance sampling proposal distribution, in this case $\pi_n(\boldsymbol{x})$, is not close to the target, the importance sampling estimator can have very high variance, scaling approximately with the variance of the importance weights [32].

Direct application of importance weighting through the annealed targets in SMC can quickly result in weight collapse, where all the importance weight is assigned to a single particle in the ensemble. This issue can be partially addressed through resampling, where the particle ensemble is resampled according to their importance weights, duplicating particles with high weight and removing particles with low weight [54]. This also gives an equal weight particle ensemble that is approximately distributed according to the annealed target. We discuss the exact resampling scheme used in this work in Appendix B.

In order to further improve the quality of the MC approximation given by the particle ensemble, one can perform sampling updates at each temperature level. If resampling has been performed, this also helps to disperse particles and remove duplicates in the ensemble. Typically, this will involve the application of some $\pi_{n+1}(\boldsymbol{x})$ invariant MCMC kernel, $K_n(\boldsymbol{x}'|\boldsymbol{x})$ for several iterations such that the particle ensemble is distributed according to $\pi_{n+1}(\boldsymbol{x})$. Pseudocode for the SMC algorithms we use as benchmarks in this work is given in Appendix C.

In principle, SMC can produce a particle ensemble that provides asymptotically unbiased approximations to posterior marginal moments, without the limitations of EKI in only being exact for Gaussian targets with linear forward models. However, in order to attain low bias on these moment estimates in practice we must run multiple iterations of the MCMC updates at each temperature level [35]. For scientific inverse problems with expensive forward models we would like to minimize the number of MCMC iterations required to achieve low bias. Previous works have leveraged ideas from EKF within MCMC, with examples including [55], where a proposal kernel was developed based on the analysis step in the EKF update, and [56] which used EKF to accelerate pseudo-marginal MCMC in state space models. In this work we propose using EKI as part of an adaptation scheme for the tpCN sampler, replacing the resampling step in SMC by instead using the EKI update as an initialization and preconditioner for each intermediate target.

3.3. Temperature Adaptation

The choice of temperature schedule is crucial to both EKI and SMC. We wish to take steps in inverse temperature that are neither too small, which would unnecessarily increase the number of model evaluations, nor too large, which would render the particle ensemble obtained at the previous temperature level of limited use in adapting the MCMC kernel used for the next target temperature. This is particularly relevant when we learn NF maps for preconditioning, where we rely on the particle distribution from the previous temperature level to inform our preconditioning.

In this work we select temperature levels by estimating the effective sample size (ESS) of the particle ensemble and choosing a value of β such that we attain some fractional ESS target. The ESS in targeting some β_{n+1} from β_n can be estimated by calculating the importance weights given by

$$w_n^i = \exp\left(-\frac{1}{2}\left(\beta_{n+1}-\beta_n\right)\left\|\Gamma^{-1/2}\left(\boldsymbol{y}-\mathcal{F}(\boldsymbol{x}_n^i)\right)\right\|_2^2\right). \tag{26}$$

The value of the target inverse temperature can then be obtained by solving for β_{n+1} in,

$$\left(\sum_{i=1}^{J} w_n^i(\beta_{n+1})^2\right)^{-1}\left(\sum_{i=1}^{J} w_n^i(\beta_{n+1})\right)^2 = \tau J, \tag{27}$$

where $0 < \tau < 1$ is the fractional ESS threshold. The value of τ controls the size of the steps in β, with larger values of τ resulting in smaller steps. This method has seen extensive application in adaptive SMC and EKI implementations due to the ability to control the ensemble ESS, which is crucial for effective resampling [57, 9].

Alternative temperature adaption schemes can be used, for example if one was seeking to use a more aggressive temperature schedule [13]. However, more agressive temperature schedules can be unstable for standard SMC, which uses importance resampling at each temperature level, if the ESS becomes very low. In order to make more direct comparisons between SKT algorithms and standard SMC algorithms in this work, we only consider the ESS-based criterion expressed in Equation 27. It is worth noting that such temperature adaptation renders SMC a biased but consistent method. However, this bias is typically negligible, and the ability to adapt the temperature schedule to each problem offers significant advantages in avoiding the need to manually select an appropriate schedule, hence the widespread use of adaptive temperature schedules in SMC [58, 59].

3.4. Normalizing Flow Preconditioning

In this paper we leverage NFs in two contexts; learning a map to a Gaussian latent space at each temperature level to improve the fidelity of the EKI target approximation [18], and to act as a preconditioner for the subsequent tpCN sampling iterations.

NFs are generative models where one learns a bijective map between some original data space, $\boldsymbol{x} \in \mathbb{R}^d$ and a simple latent space, $\boldsymbol{z} \in \mathbb{R}^d$. They can be used for highly expressive density estimation and allow efficient sampling from the learned generative model [43, 44, 45, 46]. The full bijective map, $\boldsymbol{z} = f(\boldsymbol{x})$ proceeds through a sequence of invertible transformations $f = f_1 \circ \ldots \circ f_{n_L}$, with the latent space base distribution typically chosen to be the standard Gaussian such that $\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z}) = \mathcal{N}(0, I_d)$, where I_d denotes the $d \times d$ identity matrix. Data space samples can be obtained from the NF distribution by drawing samples from the latent space base distribution and evaluating the inverse transformation $\boldsymbol{x} = f^{-1}(\boldsymbol{z})$.

The learned NF density, $q(\boldsymbol{x})$ can be evaluated using the standard change of variables formula,

$$q(\boldsymbol{x}) = p_{\boldsymbol{z}}(f(\boldsymbol{x}))\left|\det Df(\boldsymbol{x})\right| = p_{\boldsymbol{z}}(f(\boldsymbol{x}))\prod_{l=1}^{n_L}\left|\det Df_l(\boldsymbol{x})\right|, \tag{28}$$

where $Df(\boldsymbol{x}) = \partial f(\boldsymbol{x})/\partial \boldsymbol{x}$ is the Jacobian for the NF transformation. In this work we use neural spline flows (NSF) [60] as implemented in the FlowMC package [61, 62], which have been found to be highly expressive flow architectures able to capture complex target geometries. In the numerical experiments performed in this work we were able use a single set of default configurations across the test models without the need for extensive NSF hyper-parameter searches.

The impact of the NF in the annealing schemes considered in this work can be seen by considering the recursive expression for the target with inverse temperature β_{n+1},

$$\pi_{n+1}(\boldsymbol{x}) \propto \pi_n(\boldsymbol{x})\mathcal{N}(\boldsymbol{y}|\mathcal{F}(\boldsymbol{x}), \alpha_n\Gamma). \tag{29}$$

We can view $\pi_n(\boldsymbol{x})$ as a pseudo-prior for $\pi_{n+1}(\boldsymbol{x})$, with the likelihood contribution being controlled by α_n. By fitting an NF to the particle ensemble obtained for β_n, and assuming the particle ensemble is correctly distributed as $\pi_n(\boldsymbol{x})$, we can map the pseudo-prior to an approximately Gaussian space. In the NF latent space, the β_{n+1} target is given by

$$\pi_{n+1}(\boldsymbol{z}) \propto \pi_n(f_n^{-1}(\boldsymbol{z}))|\text{det}Df_n^{-1}(\boldsymbol{z})|\mathcal{N}(\boldsymbol{y}|\mathcal{F}(f_n^{-1}(\boldsymbol{z})), \alpha_n\Gamma). \tag{30}$$

The latent space pseudo-prior is approximately the standard Gaussian. For EKI updates performed in the NF latent space, we can view this as single step EKI with prior $\pi_n(\boldsymbol{z}) = \pi_n(f_n^{-1}(\boldsymbol{z}))|\text{det}Df_n^{-1}(\boldsymbol{z})|$ and target posterior $\pi_{n+1}(\boldsymbol{z})$. Provided the particle ensemble for β_n was correctly distributed according $\pi_n(\boldsymbol{x})$, by performing the EKI update in the latent space we have a Gaussian prior ensemble. If the value of α_n is chosen to be sufficiently large (i.e., small step size in β) such that $\pi_n(\boldsymbol{z})$ is prior dominated, we are able to effectively relax the Gaussian ansatz of EKI.

Whilst the use to NFs in EKI has been found to improve robustness against non-Gaussianity [18], the NF maps do not address the assumption that the forward model is linear. If the forward model is nonlinear the EKI update will not be exact, even when performed in the Gaussian latent space. This means the particle ensemble will not be correctly distributed according to the subsequent tempered target. When this is then treated as the pseudo-prior for the next temperature level, the NF will map the incorrect particle ensemble to a Gaussian latent space which does not correspond to the correct pseudo-prior distribution. These errors can accumulate as one progresses from the prior to the posterior, resulting in a low fidelity ensemble approximation to the final posterior.

Nonetheless, within our adaptive sampling scheme NF preconditioning has the benefit of helping to stabilize EKI/Flow Annealed Kalman Inversion (FAKI) updates, and acting as a nonlinear preconditioner for the tpCN updates. Considering again Equation 30, if the particle ensemble obtained for β_n can be mapped to a Gaussian latent space, and the value of α_n is chosen such that the β_{n+1} target is dominated by the pseudo-prior, the NF provides a highly effective preconditioner that is able to account for local variations in the target geometry. Mapping to a Gaussian latent space has the additional advantage of allowing us to use scaling relations derived for samplers with Gaussian targets when selecting sampling hyper-parameters [20, 63, 64]. The use of NFs as preconditioners has seen several applications for sampling, including in MCMC [65, 61], with interacting particle systems [28] and in SMC [35, 42].

4. Adaptation of tpCN with Ensemble Kalman Inversion in Sequential Monte Carlo

In this section we describe the SKT adaptation procedure we propose for tpCN. In essence, EKI is used within a Bayesian annealing scheme, with the core performance improvements arising from the ability of the EKI updates to provide an effective initialization and preconditioner for the tpCN sampling iterations at each temperature level. The tpCN iterations are then able to efficiently converge on the target at each temperature level, allowing for accurate posterior moment estimation. In Section 4.1 we describe the procedure for adapting the tpCN kernel parameters, and a method for selecting the number of tpCN iterations to perform at each temperature level. We then provide pseudocode outlining the full adaptive sampling algorithm in Section 4.2.

4.1. Adaptation of tpCN kernel parameters

At each temperature level we need to select values of the tpCN kernel parameters such that the base t-distribution approximates the target distribution well. In [51], a parallel adaptation scheme for elliptical slice sampling was used where the parameters of the proposal t-distribution were obtained by dividing a particle ensemble into two groups. The parameters for the proposal in one group were then obtained by fitting a t-distribution to the particles in the other group. For the adaptive SMC scheme we consider in this work we do not need to divide the ensemble into groups, instead using the particle ensemble prior to sampling at each temperature level to fit for the t-distribution parameters. This also means we do not need to alternate the tpCN updates between groups as in [51], instead relying on control of the transition between temperature levels and the quality of the EKI target approximation to provide an effective tpCN kernel.

To select the t-distribution parameters at each temperature level we use the expectation maximization (EM) algorithm [66, 67], described in Algorithm 4 of [51]. This is a stable choice provided the size of the particle ensemble $J \geq 2d$, where d is the target dimension. As noted in [51], more sophisticated procedures could be used in high dimensions, although the structure of our adaptation scheme would be largely the same. The t-distribution parameters are fitted to the ensemble after applying the EKI update, such that it approximates the target distribution. Similarly, for the standard SMC benchmarks we fit the t-distribution parameters to the resampled particle ensemble.

In addition to selecting appropriate parameters for the tpCN kernel, it is also important to run a sufficient number of tpCN iterations at each temperature level to ensure particles are distributed according to each intermediate target. In this work we study both the sampling performance using a fixed number of sampling iterations at each temperature level, in order to more directly assess the impact of the EKI adaptation step compared to resampling, and also present numerical results when selecting the number of tpCN iterations based on autocorrelation statistics. Such an approach has previously been used in the context of adaptive SMC, for example in [41] the number of MCMC iterations was selected by monitoring the component-wise first-order autocorrelation of the statistic $\boldsymbol{x}^i_{n,m}(j) + \boldsymbol{x}^i_{n,m}(j)^2$, where i denotes the ensemble member, n denotes the temperature level, m denotes the MCMC iteration number and j denotes the component of $\boldsymbol{x}^i_{n,m}$. This statistic monitors the correlation of the first and second moments of the ensemble. In this work we present numerical results where MCMC iterations are performed at each temperature level until the product of the first-order autocorrelations falls below some threshold τ_{corr} for all dimensions i.e.,

$$\prod_{m=1}^{M} \hat{\rho}_m(j) < \tau_{\mathrm{corr}} \, \forall j, \tag{31}$$

where $\hat{\rho}_m(j)$ is the autocorrelation statistic calculated over the ensemble for successive states $\{\boldsymbol{x}_{n,m-1}^i\}_{i=1}^J$ and $\{\boldsymbol{x}_{n,m}^i\}_{i=1}^J$.

An important difference to note between the adaptive SKT samplers we propose here and standard SMC is that, for a fixed number of MCMC iterations at each temperature level, the SMC particle ensemble will converge asymptotically to the target posterior as the ensemble size $J \to \infty$ [32]. In contrast, the SKT algorithm requires sufficient MCMC iterations to be performed at each temperature level such that we converge on each intermediate target. However, in the practical settings we consider in this work, where the ensemble size is some multiple of the target dimension, standard SMC still requires a large number of MCMC iterations at each temperature level to ensure we obtain low bias posterior moment estimates. This is required to ensure the particle ensemble does not collapse through the repeated resampling steps, and to remove duplicates from the resampled ensembles that can otherwise provide high variance posterior moment estimates. In Section 5 we provide numerical results demonstrating both the need for a large number of MCMC iterations at each temperature level in SMC, and the ability of the EKI adaptation step to accelerate the convergence of the MCMC iterations.

4.2. Sequential Kalman Tuning for tpCN

All of the procedures we have outlined thus far can be combined to produce the SKT adaptive sampling scheme for tpCN. The core of the SKT approach lies in replacing the importance resampling step of SMC with an EKI update. That is, given an ensemble of particles $\{\boldsymbol{x}_n^i\}_{i=1}^J$ distributed according to the target at inverse temperature β_n, we apply the EKI update in Equation 21 to obtain a particle ensemble that approximates the target at β_{n+1}, which acts as an initialization and preconditioner for subsequent sampling updates. We can then fit for the parameters of the t-distribution reference measure before performing tpCN updates to correctly distribute the particle ensemble according to the target at β_{n+1}. In addition to performing the EKI update at each temperature level, we can also use NF preconditioning to improve the stability of EKI when approximating non-Gaussian measures, and to act as a nonlinear preconditioner for the tpCN sampling. Pseudocode for SKT with NF preconditioning is given in Algorithm 3. For SKT without NF preconditioning, the structure of the algorithm is largely identical, without any NF fits being performed such that EKI and tpCN updates are performed in the original data space. For completeness, we provide pseudocode describing the benchmark SMC implementations used in this work in Appendix C.

The use of EKI as an adaptation step within an annealed sampling scheme has two core benefits. Compared to applying EKI directly to solving the Bayesian inverse

Algorithm 3 Flow Preconditioned Sequential Kalman Tuning

```
Input: Set of $J$ samples from the prior $\{\boldsymbol{x}_0^i \sim \pi_0(\boldsymbol{x})\}_{i=1}^J$, data $\boldsymbol{y}$, observation covariance $\Gamma$, target fractional ESS $\tau$, maximum number of tpCN iterations to perform at each temperature level $M$, initial tpCN step size $\rho$, target tpCN acceptance rate $\alpha^\star$, tpCN autocorrelation threshold $\tau_{\text{corr}}$.
Set $\beta_0 = 0$ and iteration counter $n = 0$.
while $\beta_n < 1$ do
    Solve for target inverse temperature $\beta_{n+1}$ in Equation 27.
    if $\beta_{n+1} = 1$ then
        $n^* \leftarrow n + 1$
    end if
    $\alpha_n \leftarrow \beta_{n+1} - \beta_n$
    Fit NF map, $\boldsymbol{z} = f_n(\boldsymbol{x})$ to current particle locations $\{\boldsymbol{x}_n^i\}_{i=1}^J$.
    Obtain latent space particle locations $\{\boldsymbol{z}_n^i = f_n(\boldsymbol{x}_n^i)\}_{i=1}^J$.
    for $i = 1, \dots, J$ do
        Update latent space particle ensemble with
            $\boldsymbol{z}_{n+1}^i = \boldsymbol{z}_n^i + C_n^{\boldsymbol{z}\mathcal{F}} \left(C_n^{\mathcal{F}\mathcal{F}} + \alpha_n \Gamma\right)^{-1} \left(\boldsymbol{y} - \mathcal{F}(f_n^{-1}(\boldsymbol{z}_n^i)) + \sqrt{\alpha_n}\boldsymbol{\xi}_n^i\right), \quad (32)$
        where $C_n^{\boldsymbol{z}\mathcal{F}}$ and $C_n^{\mathcal{F}\mathcal{F}}$ are defined analogously to Equations 22 and 23 respectively in the NF latent space, and $\boldsymbol{\xi}_n^i \sim \mathcal{N}(0, \Gamma)$.
    end for
    Fit the multivariate $t$-distribution, $t_{\nu_s}(\mu_s, \mathcal{C}_s)$ to the latent space particle ensemble $\{\boldsymbol{z}_{n+1}^i\}_{i=1}^J$ with an EM algorithm. Set component-wise autocorrelations $\hat{\rho}_0(j) = 1 \; \forall j$.
    for $m = 1, \dots, M$ do
        for $i = 1, \dots, J$ do
            Update particle state $\boldsymbol{z}_{n+1}^i$ using Algorithm 2 in NF latent space.
        end for
        $\log \rho \leftarrow \log \rho + (\langle \alpha \rangle - \alpha^\star)/m$
        $\boldsymbol{\mu}_s \leftarrow \boldsymbol{\mu}_s + (\langle \boldsymbol{z}_{n+1} \rangle - \boldsymbol{\mu}_{\boldsymbol{s}})/m$
        Calculate component-wise autocorrelations $\hat{\rho}_m(j)$.
        if $\prod_{l=1}^m \hat{\rho}_l(j) < \tau_{\text{corr}} \; \forall j$ then
            End tpCN iterations.
        end if
    end for
    Map particle ensemble back to the original data space $\{\boldsymbol{x}_{n+1}^i = f_n^{-1}(\boldsymbol{z}_{n+1}^i)\}_{i=1}^J$
    $n \leftarrow n + 1$
end while
Output: Converged particle ensemble $\{\boldsymbol{x}_{n^*}^i\}_{i=1}^J$.
```

problem, using it as an adaptation step for the tpCN sampler that preserves the target measure as its invariant measure, we are able to obtain low bias posterior moment estimates outside the linear, Gaussian setting where standard EKI can otherwise give highly biased results. Beyond allowing us to correct the errors in direct EKI and FAKI, the EKI adaptation significantly accelerates tpCN sampling. The EKI update distributes particles approximately according to the target measure. When used to fit the reference t-distribution for tpCN, we are able to more closely capture the target geometry. Coupled with NF preconditioning, we obtain a doubly preconditioned sampler, with the NF mapping us to an approximately Gaussian latent target space, and the t-preconditioning in tpCN giving improved performance in sampling any residual non-Gaussianity in the target.

For the tpCN implementations, both with the SKT adaptation and the benchmark SMC adaptation, we perform diminishing adaptation [68] of the tpCN step size ρ, and the reference t-distribution mean $\boldsymbol{\mu}_s$. For some sampling iteration m, the tpCN kernel parameters at iteration $m + 1$ are given by

$$\log \rho^{m+1} = \log \rho^m + \frac{\langle \alpha^m \rangle - \alpha^\star}{m}, \tag{33}$$

$$\boldsymbol{\mu}_s^{m+1} = \boldsymbol{\mu}_s^m + \frac{\langle \boldsymbol{x}^m \rangle - \boldsymbol{\mu}_s^m}{m}, \tag{34}$$

where $\langle \alpha^m \rangle$ is the mean tpCN acceptance probability at iteration m, $\alpha^\star$ is some target acceptance probability and $\langle \boldsymbol{x}^m \rangle$ is the mean of the particle ensemble at iteration m. Performing diminishing adaptation in this way helps to ensure the robust performance of the tpCN algorithm across all the adaptive sampling schemes, with similar adaptation previously being implemented in the POCOMC package for NF preconditioned SMC [35, 42].

5. Numerical Experiments

In this section we present the results from three numerical experiments. In Section 5.1, we study the recovery of an initial temperature field evolving under the heat equation. In Section 5.2, we study the recovery of an underlying density field from surface measurements of the gravitational field. Finally, in Section 5.3, we study the recovery of a source term from observations of a signal evolving under the reaction-diffusion equation.

We compare the performance of the adaptive SKT scheme against adaptation with importance resampling SMC, both with and without NF preconditioning. For the purposes of labelling the results from each adaptation algorithm we use the following acronyms:

(i) **SKT**: The SKT algorithm without NF preconditioning, analogous to Algorithm 3 without the NF steps.

(ii) **NF-SKT**: The SKT algorithm with NF preconditioning, as described in Algorithm 3.

(iii) **SMC**: Importance resampling SMC without NF preconditioning, analogous to Algorithm 5 without the NF steps.

(iv) **NF-SMC**: Importance resampling SMC with NF preconditioning, as described in Algorithm 5.

Alongside testing the performance of the adaptation algorithms for the tpCN sampler, we also provide results for standard pCN. When adapting the pCN sampler, we fit for the mean and covariance of the Gaussian base distribution using the empirical mean and covariance of the particle ensemble prior to sampling at each temperature level. Learning the NF maps at each temperature level took approximately 10 seconds for each of the experiments we consider here. This could likely be further improved by implementing e.g., early stopping based on the NF validation loss [42]. However, in general, we expect that learning each NF map will take of order seconds of wall time at each temperature level up to $\mathcal{O}(100)$ dimensions. For problems where the cost of a single forward model evaluation is comparable, the NF training cost becomes negligible compared to the need to apply repeated sampling updates at each temperature level. However, we find that obtaining high quality NF fits requires scaling both the number of particles and the complexity of the NF maps with dimension, which will quickly render NF training prohibitive as we move to $\mathcal{O}(10^3)$ dimensions.

To quantify the performance of the samplers we compare the squared bias on the estimated first and second moments of the target posterior, averaged over the target dimensions, which has previously been used in studying the rate of convergence of MCMC algorithms [65, 39]. The dimension averaged squared bias, normalized by the posterior variance, on the estimate for some quantity $g(\boldsymbol{x})$ is given by

$$\langle b_g^2 \rangle = \left\langle \frac{(\mathbb{E}_{\beta=1}[g_k(\boldsymbol{x})] - \mathbb{E}_\pi[g_k(\boldsymbol{x})])^2}{\sigma_{g,k}^2} \right\rangle_{k \in d}, \tag{35}$$

where $\mathbb{E}_{\beta=1}[g_k(\boldsymbol{x})] = J^{-1} \sum_{i=1}^{J} g_k(\boldsymbol{x}_{n^\star}^i)$ is the mean of $g(\boldsymbol{x})$ for the dimension k, evaluated over the final particle ensemble $\{\boldsymbol{x}_{n^\star}^i\}_{i=1}^J$, $\mathbb{E}_\pi[g_k(\boldsymbol{x})]$ is the expectation value of $g(\boldsymbol{x})$ for the dimension k, evaluated with respect to the true target posterior, $\sigma_{g,k}^2$ is the true posterior variance of $g(\boldsymbol{x})$ for the dimension k, and $\langle \cdot \rangle_{k \in d}$ denotes the average over the dimensions. We estimate $\mathbb{E}_\pi[g_k(\boldsymbol{x})]$ and $\sigma_{g,k}^2$ for each problem from long runs of Hamiltonian Monte Carlo (HMC), using the No-U-Turn Sampler implementation in the NUMPYRO library [69, 70]. We denote the dimension averaged squared bias on the first moment ($g_k(\boldsymbol{x}) = \boldsymbol{x}_k$) as $\langle b_1^2 \rangle$ and on the second moment ($g_k(\boldsymbol{x}) = \boldsymbol{x}_k^2$) as $\langle b_2^2 \rangle$, where $\boldsymbol{x}_k$ is the element of $\boldsymbol{x}$ corresponding to the dimension k.

Given N independent samples from the posterior $\{\hat{\boldsymbol{x}}_i\}_{i=1}^N$, we have the estimator $\mathbb{E}_\pi[g_k(\boldsymbol{x})] = N^{-1} \sum_{i=1}^{N} g_k(\hat{\boldsymbol{x}}_i)$. Invoking the central limit theorem, the squared error on this estimator will be of the order $\sim \sigma_{g,k}^2/N$ [39]. Whilst we do not have independent samples from the posterior from HMC, we ensure that we run chains sufficiently long such that the estimated effective sample size (ESS) $\gtrsim 10^3$. These heuristics can also be used to define a regime for low bias where $\langle b_g^2 \rangle < 10^{-2}$, which corresponds approximately with a dimension averaged squared bias less than one hundredth of the posterior variance.

In comparing the performance of each adaptive algorithm, we consider two sets of tests. In the first we run the SKT and NF-SKT algorithms with 10 tpCN (pCN) iterations at each temperature level, and the SMC and NF-SMC algorithms with 11 tpCN (pCN) iterations at each temperature level. The fixed computational budget at each temperature level allows for a more direct assessment of the performance of the EKI adaptation step. The additional tpCN (pCN) iteration for SMC and NF-SMC is to account for the additional set of forward model evaluations used for the EKI updates in SKT and NF-SKT. We report results for ensemble sizes $J \in \{2d, 4d, 6d, 8d, 10d\}$, where d is the target dimension of each model. In the second set of tests we compare the performance of each algorithm using an adaptive number of sampling iterations at each temperature level, with a correlation threshold of $\tau_{\text{corr}} = 0.1$ and an ensemble size $J = 10d$. To avoid excessive computation, we set the maximum number of sampling iterations at each temperature level to 50 for SKT and NF-SKT, and 51 for SMC and NF-SMC. For the results presented in this section we use a target tpCN (pCN) acceptance rate of $\alpha^\star = 0.234$, an initial tpCN (pCN) step size of $\rho = 1$ and a fractional target ESS of $\tau = 0.5$ when adapting the annealing schedule. Each algorithm is run over ten different random seeds to estimate the corresponding variation in performance. For completeness, we also provide corner plots comparing the converged particle ensembles obtained with each adaptation method against reference HMC samples in Appendix E.

In Appendix F, we demonstrate the performance of EKI and FAKI on our numerical benchmarks without embedding them as part of an annealed sampling scheme. Each of the numerical experiments we consider in this work show varying degrees of non-Gaussianity, which results in highly biased posterior inferences for EKI and FAKI. Whilst both methods converge on their final ensembles with $\sim 30 - 50$ embarrassingly parallel model evaluations, the highly biased posterior moment estimates mean such methods are unsuitable for many scientific inference tasks when applied alone.

In addition to the bias on posterior moment estimates, one may also be interested in the field reconstructions obtained with each method. In Appendix D we show the relevant field and source term reconstructions for each experiment obtained with each algorithm. For FAKI and EKI there are clear discrepancies between the reconstructed fields and source terms and the reference reconstruction from HMC. This is to be expected given the highly biased final ensembles obtained with these methods. For each of the adaptive sampling schemes considered in this work the qualitative reconstruction of the initial fields and source terms was largely comparable to the recovery with HMC, both applied to tpCN and pCN. However, the ability of each method to obtain comparable field and source term recoveries does not fully reflect the ability of the various algorithms in accurately approximating marginal posterior moments, which are the key object of study for many scientific inference tasks.

5.1. Heat Equation

The heat equation is a partial differential equation (PDE) describing the evolution of some field $u(\boldsymbol{x}, t)$ over time. For our experiment we consider the case of a two-dimensional temperature field evolving according to

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t} = D\nabla^2 u(\boldsymbol{x},t) = D\left(\frac{\partial^2 u(\boldsymbol{x},t)}{\partial x_1^2} + \frac{\partial^2 u(\boldsymbol{x},t)}{\partial x_2^2}\right), \tag{36}$$

where we set the thermal diffusivity constant $D = 0.5$. The temperature field is taken to be on a square plate, with the length of a side set to $L = 10$. We impose Dirichlet boundary conditions on the domain $\Omega \subset \mathbb{R}^2$, such that $u(\boldsymbol{x}, t) = 0, \forall \boldsymbol{x} \in \partial\Omega$. The forward model consists in solving Equation 36 for the evolution of some initial temperature field $u(\boldsymbol{x}, t = 0)$ up to a time $t_f = 1$. We solve the heat equation using the forward time centered space (FTCS) method [71] on a 64×64 grid, with 1000 time steps.

For our simulated data, we consider the situation where measurements of the temperature field are made at time t_f on a low resolution 8×8 grid, with the signal in a low resolution grid pixel being the average of the temperature signal from the 64×64 grid pixels contained within it. The observation noise was taken to be independent in each pixel, with a Gaussian noise standard deviation of $\sigma_\eta = 0.2$. The true initial temperature field was generated from the Karhunen-Loeve (KL) expansion of a Gaussian random field (GRF) with a squared exponential covariance kernel

$$C(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2}{2\ell^2}\right), \tag{37}$$

where ℓ is the GRF length scale. The KL expansion, up to some order R, for the GRF is given by

$$u(\boldsymbol{x}, t = 0) = \mu_K + \sigma_K \sum_{k=1}^{R} \sqrt{\lambda_k}\phi_k(\boldsymbol{x})\theta_k, \tag{38}$$

where μ_K is the GRF mean, σ_K^2 is the GRF variance, $\{\lambda_k\}_{k=1}^R$ is a sequence of strictly decreasing, real and positive eigenvalues for the covariance kernel in Equation 37, $\{\phi_k(\boldsymbol{x})\}_{k=1}^R$ are the set of corresponding eigenfunctions of the covariance kernel, and $\{\theta_k \sim \mathcal{N}(0, 1)\}_{k=1}^R$ are a set of standard Gaussian random variables. To generate the true field for this numerical study we set $\mu = 0$, $\sigma^2 = 1$, $\ell = 0.1$ and generate $R = 200$ standard Gaussian random variables $\{\theta_k\}_{k=1}^{200}$. The simulated data are then generated by solving for the time evolution of $u(\boldsymbol{x}, t)$ up to time $t_f = 1$, averaging the signal onto the low resolution grid and adding Gaussian noise realizations to each pixel. The true initial temperature field and the low resolution observed field are shown in Figure 1.

For the test model, we consider recovering the $R = 100$ leading modes of the KL expansion, along with the thermal diffusivity constant. Defining $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{100})^\intercal$, the

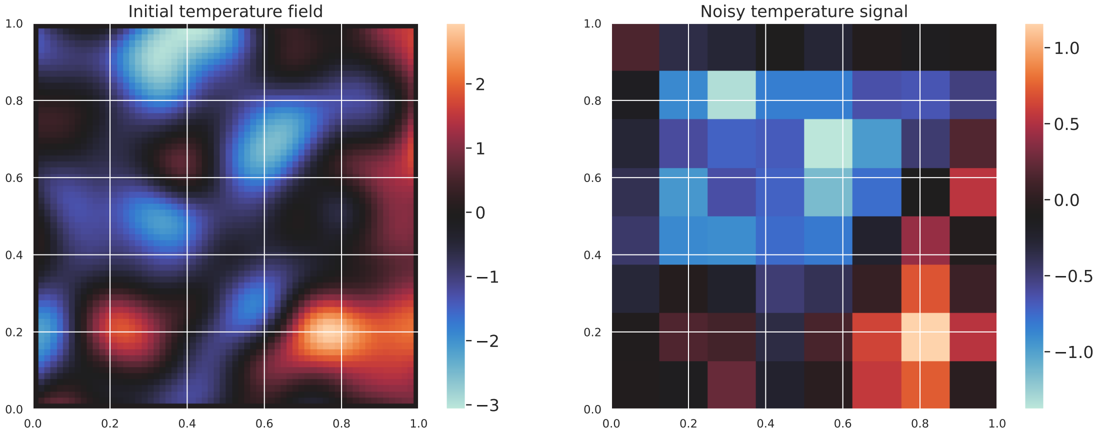


Figure 1: The initial temperature field $u(\boldsymbol{x}, t = 0)$ (left panel), shown alongside the low resolution observed temperature field at time $t_f = 1$ (right panel).

full model is given by

$$D \sim |\mathcal{N}(\mu = 0, \sigma^2 = 0.5^2)|, \tag{39}$$

$$\mu_K \sim \mathcal{N}(\mu = 0, \sigma = 0.1), \tag{40}$$

$$\sigma_K \sim |\mathcal{N}(\mu = 0, \sigma^2 = 1.0)|, \tag{41}$$

$$\boldsymbol{\theta} \sim \mathcal{N}(0, I_{100}), \tag{42}$$

$$\boldsymbol{y} \sim \mathcal{N}(F_H(D, \mu_K, \sigma_K, \boldsymbol{\theta}), \sigma_\eta^2 I_{64}), \tag{43}$$

where $|\mathcal{N}(\mu = 0, \sigma^2)|$ is the Half-Normal distribution with scale σ and $F_H(D, \mu_K, \sigma_K, \boldsymbol{\theta})$ denotes the forward model for the heat equation, mapping from the initial temperature field to the low resolution observations $\boldsymbol{y}$ at time $t_f = 1$. For performing inference we apply log-transformations to D and σ_K to map the all the parameters to an unconstrained space, modifying the the target distribution with the corresponding Jacobian factors. The target dimension for this problem is $d = 103$.

The first set of results are provided when running each adaptive algorithm with a fixed computational budget at each temperature level. In Figure 2 we show the results for $\langle b_1^2 \rangle$ obtained with the final particle ensembles for each algorithm, with the box plots showing the variation over the 10 runs. Similarly, Figure 3 shows the results for $\langle b_2^2 \rangle$ obtained for each algorithm with a fixed computational budget at each temperature level. The mean and standard deviation of $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$ for each algorithm over their 10 runs are reported in Table 1 for tpCN and in Table 2 for pCN, alongside the number of temperature levels, N_β used by each algorithm. For the second set of results, where the number of sampling iterations at each temperature level is chosen by monitoring sample autocorrelations, we report the mean and standard deviation of $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$ for each algorithm over their 10 runs in Table 3, alongside the number of forward model evaluations divided by the ensemble size N_{eval}/J, which corresponds to the number of embarrassingly parallel forward model evaluations.

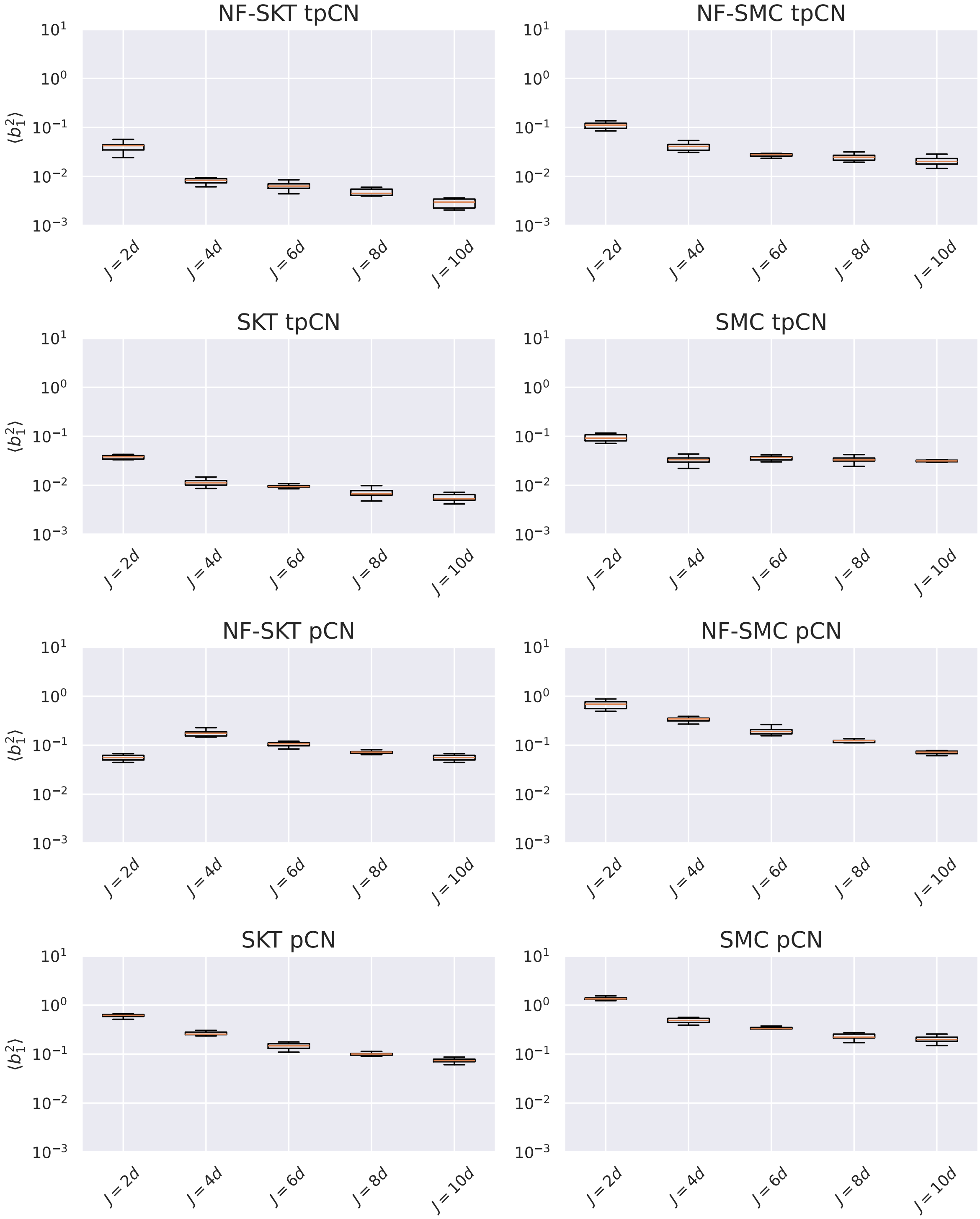


Figure 2: Final dimension averaged squared bias on the first moment for the heat equation model, plotted against the ensemble size J expressed as a multiple of the target dimension $d = 103$. Results are shown for each adaptation algorithm applied to the tpCN and pCN samplers.

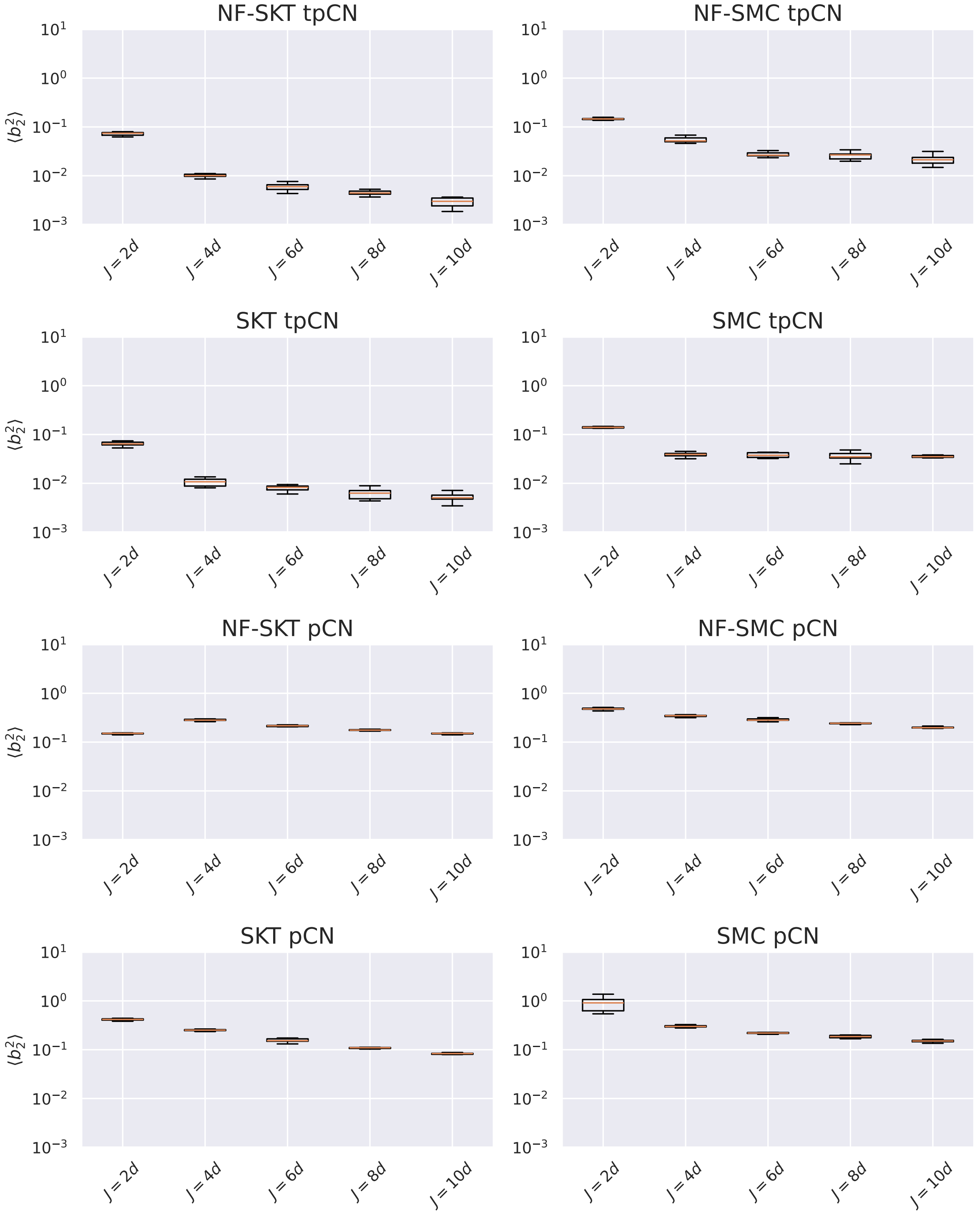


Figure 3: Final dimension averaged squared bias on the second moment for the heat equation model, plotted against the ensemble size J expressed as a multiple of the target dimension $d = 103$. Results are shown for each adaptation algorithm applied to the tpCN and pCN samplers.

Beginning with comparisons where we use a fixed computational budget at each temperature level, we find that tpCN achieves lower squared bias on the first and second moments compared to pCN. Given the same adaptive SKT and SMC schemes, the greater flexibility in the base t-distribution means tpCN is able to better adapt to the non-Gaussian targets, resulting in more rapid convergence to the target at each temperature level.

Comparing SKT and SMC adaptation schemes, we can see that SKT obtains lower values for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$ for all ensemble sizes. The number of temperature levels used by both algorithms is comparable. It is apparent from these tests using SKT adaptation we are able to converge more rapidly at each temperature level, indicating that the EKI update provides a better initialization for the tpCN updates than the importance resampling used in SMC. The EKI update also helps to provide improved preconditioning for the tpCN updates, with the t-distribution being fitted to the annealed target particle approximation obtained via the EKI update.

A similar pattern is observed when comparing NF-SKT and NF-SMC. In comparing each with SKT and SMC respectively, we see that the improvement from NF preconditioning becomes more pronounced as the ensemble size is increased. For $J = 2d$ there are not enough particles for the NF to learn a useful map between between the original data space and a Gaussian latent space. Indeed, in this regime the NF can degrade performance by failing to map to a latent space where the target is effectively Gaussianized, and in the case of NF-SKT introducing additional non-linearity in the forward model evaluation for the EKI update. For larger ensemble sizes ($J \geq 6d$) the use of NF transformations reduces the final bias for both the NF-SKT and NF-SMC algorithms. The effect is more pronounced for NF-SKT, where the NF acts to both relax the Gaussian ansatz of EKI and provide nonlinear preconditioning for the tpCN updates. However, it is worth noting that SKT was able to achieve low bias without NF preconditioning, demonstrating the potential of the EKI update as an adaptation step for tpCN by both initializing and preconditioning the sampling updates within an annealing scheme.

When we select the number of sampling iterations at each temperature level based on the first order autocorrelations, we see that for the NF-SKT and SKT adaptation schemes, the tpCN sampler is able to reach the low bias regime ($\langle b_g^2 \rangle < 10^{-2}$). This is not the case for the NF-SMC and SMC samplers. From the tests using a fixed computational budget at each temperature level, we expect these adaptation schemes to require more tpCN iterations at each temperature level. Despite this, the tpCN sampling updates are terminated earlier for the NF-SMC and SMC adaptation schemes, indicating that a lower value of τ_{corr} is required. For pCN, we do not reach the low bias regime for all adaptation schemes, again indicating a more stringent requirement on the number of sampling iterations is necessary.

Algorithm	J	N_β	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT tpCN	$10d$	15.0 ± 0.0	0.0029 ± 0.0006	0.0029 ± 0.0007
NF-SKT tpCN	$8d$	15.0 ± 0.0	0.0048 ± 0.0008	0.0045 ± 0.0005
NF-SKT tpCN	$6d$	15.0 ± 0.0	0.0065 ± 0.001	0.0060 ± 0.001
NF-SKT tpCN	$4d$	15.0 ± 0.0	0.0084 ± 0.001	0.010 ± 0.007
NF-SKT tpCN	$2d$	14.8 ± 0.4	0.040 ± 0.009	0.072 ± 0.006
SKT tpCN	$10d$	15.0 ± 0.0	0.0056 ± 0.001	0.0053 ± 0.001
SKT tpCN	$8d$	15.0 ± 0.0	0.0071 ± 0.001	0.0062 ± 0.001
SKT tpCN	$6d$	15.0 ± 0.0	0.0094 ± 0.001	0.0080 ± 0.001
SKT tpCN	$4d$	15.0 ± 0.0	0.011 ± 0.002	0.011 ± 0.002
SKT tpCN	$2d$	15.0 ± 0.0	0.038 ± 0.003	0.064 ± 0.007
NF-SMC tpCN	$10d$	15.2 ± 0.4	0.021 ± 0.004	0.021 ± 0.005
NF-SMC tpCN	$8d$	15.7 ± 0.5	0.025 ± 0.004	0.026 ± 0.004
NF-SMC tpCN	$6d$	15.4 ± 0.5	0.029 ± 0.005	0.028 ± 0.005
NF-SMC tpCN	$4d$	15.5 ± 0.5	0.041 ± 0.007	0.055 ± 0.008
NF-SMC tpCN	$2d$	14.9 ± 0.3	0.11 ± 0.02	0.15 ± 0.009
SMC tpCN	$10d$	15.3 ± 0.5	0.032 ± 0.004	0.036 ± 0.004
SMC tpCN	$8d$	15.5 ± 0.5	0.034 ± 0.005	0.037 ± 0.007
SMC tpCN	$6d$	15.7 ± 0.5	0.036 ± 0.004	0.038 ± 0.004
SMC tpCN	$4d$	15.6 ± 0.5	0.033 ± 0.006	0.039 ± 0.004
SMC tpCN	$2d$	15.1 ± 0.3	0.094 ± 0.02	0.14 ± 0.009

Table 1: Results for the number of temperature levels used by each algorithm N_β and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm when performing inference on the heat equation example, adapting the tpCN sampler. We report the mean and standard deviation for each statistic over the 10 algorithm runs, and show results for each of the tested ensemble sizes J. The number of parallelized model evaluations is given by $11N_\beta$ for each algorithm, with the total number of model evaluations being given by $11JN_\beta$. The target dimension is $d = 103$.

5.2. Gravity Survey

For this problem we adapt the two-dimensional gravity surveying problem presented in [72]. We have some mass density field $\varrho(\boldsymbol{x})$, located at a depth δ from the surface at which measurements of the vertical component of the gravitational field are made. The vertical component of the gravitational field at some point $\boldsymbol{s}$ at the surface is given by

$$\zeta(\boldsymbol{s}) = \iint_X \frac{\delta}{\|\boldsymbol{s} - \boldsymbol{x}\|_2^3} \varrho(\boldsymbol{x}) \mathrm{d}\boldsymbol{x}, \tag{44}$$

where $X = [0, 1]^2$ is the domain $\varrho(\boldsymbol{x})$. The forward model therefore consists in solving the integral in Equation 44. We follow [72] in evaluating this integral using midpoint

Algorithm	J	N_β	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT pCN	$10d$	13.2 ± 0.4	0.056 ± 0.008	0.15 ± 0.006
NF-SKT pCN	$8d$	13.0 ± 0.0	0.072 ± 0.006	0.18 ± 0.0004
NF-SKT pCN	$6d$	13.0 ± 0.0	0.10 ± 0.01	0.21 ± 0.007
NF-SKT pCN	$4d$	13.0 ± 0.0	0.18 ± 0.02	0.28 ± 0.01
NF-SKT pCN	$2d$	13.2 ± 0.4	0.056 ± 0.008	0.15 ± 0.006
SKT pCN	$10d$	15.9 ± 0.3	0.074 ± 0.008	0.083 ± 0.005
SKT pCN	$8d$	15.9 ± 0.3	0.097 ± 0.001	0.11 ± 0.006
SKT pCN	$6d$	15.9 ± 0.3	0.15 ± 0.02	0.16 ± 0.01
SKT pCN	$4d$	15.7 ± 0.5	0.26 ± 0.02	0.25 ± 0.01
SKT pCN	$2d$	15.2 ± 0.4	0.61 ± 0.05	0.41 ± 0.02
NF-SMC pCN	$10d$	15.0 ± 0.0	0.073 ± 0.009	0.20 ± 0.006
NF-SMC pCN	$8d$	15.1 ± 0.3	0.12 ± 0.01	0.24 ± 0.006
NF-SMC pCN	$6d$	15.9 ± 0.3	0.20 ± 0.01	0.29 ± 0.02
NF-SMC pCN	$4d$	17.4 ± 0.5	0.33 ± 0.03	0.34 ± 0.02
NF-SMC pCN	$2d$	20.6 ± 0.4	0.67 ± 0.12	0.48 ± 0.03
SMC pCN	$10d$	18.2 ± 0.4	0.21 ± 0.03	0.15 ± 0.008
SMC pCN	$8d$	19.2 ± 0.4	0.23 ± 0.03	0.19 ± 0.01
SMC pCN	$6d$	17.4 ± 0.5	0.33 ± 0.04	0.22 ± 0.01
SMC pCN	$4d$	21.2 ± 0.4	0.48 ± 0.06	0.30 ± 0.01
SMC pCN	$2d$	14.6 ± 0.5	1.36 ± 0.16	0.88 ± 0.26

Table 2: Results for the number of temperature levels used by each algorithm N_β and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm when performing inference on the heat equation example, adapting the pCN sampler. We report the mean and standard deviation for each statistic over the 10 algorithm runs, and show results for each of the tested ensemble sizes J. The number of parallelized model evaluations is given by $11N_\beta$ for each algorithm, with the total number of model evaluations being given by $11JN_\beta$. The target dimension is $d = 103$.

quadrature. Using Q quadrature points along each dimension, the integral expression becomes

$$\zeta(\boldsymbol{s}_i) = \sum_{l=1}^{Q} \omega_l \sum_{k=1}^{Q} \omega_k \frac{\delta}{\|\boldsymbol{s}_i - \boldsymbol{x}_{k,l}\|_2^3} \hat{\varrho}(\boldsymbol{x}_{k,l}) = \sum_{j=1}^{Q^2} \omega_j \frac{\delta}{\|\boldsymbol{s}_i - \boldsymbol{x}_j\|_2^3} \hat{\varrho}(\boldsymbol{x}_j), \tag{45}$$

where $\omega_j = 1/Q^2, \forall j$ are the quadrature weights, $\hat{\varrho}(\boldsymbol{x}_j)$ is the approximate subsurface density at the quadrature point $\boldsymbol{x}_j$, and $\zeta(\boldsymbol{s}_i)$ is the vertical component of the gravitational field at the collocation point on the surface $\boldsymbol{s}_i, i \in \{1, \dots, N^2\}$.

The simulated data was obtained by generating a ground truth subsurface density

Algorithm ($\tau_{\text{corr}} = 0.1$)	N_{eval}/J	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT tpCN	210 ± 140	0.0029 ± 0.0005	0.0028 ± 0.0004
NF-SKT pCN	320 ± 210	0.056 ± 0.008	0.15 ± 0.006
SKT tpCN	390 ± 180	0.0031 ± 0.0007	0.0034 ± 0.0009
SKT pCN	790 ± 24	0.069 ± 0.008	0.079 ± 0.005
NF-SMC tpCN	165 ± 31	0.030 ± 0.005	0.033 ± 0.006
NF-SMC pCN	170 ± 22	0.093 ± 0.012	0.19 ± 0.009
SMC tpCN	180 ± 22	0.039 ± 0.003	0.045 ± 0.004
SMC pCN	930 ± 24	0.19 ± 0.03	0.15 ± 0.0007

Table 3: Results for the number of embarrassingly parallel model evaluations N_{eval}/J, and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm when adapting the number of sampling iterations at each temperature level using $\tau_{\text{corr}} = 0.1$, as applied to the heat equation example. We show results when adapting both the tpCN and pCN samplers with an ensemble size of $J = 10d$, where the target dimension is $d = 103$.

field with profile given by

$$\varrho(\boldsymbol{x}) \propto \sin(\pi x_1) + \sin(3\pi x_2) + x_2 + 1, \quad x_1, x_2 \in [0, 1] \tag{46}$$

normalized to have a maximum value of 1. This signal was projected onto a 64×64 grid. The surface signal was evaluated using Equation 45 on a 10×10 grid, with Gaussian white noise with standard deviation $\sigma_\eta = 0.1$ being added to each surface pixel to give the simulated data. The true subsurface mass density, and the corresponding surface gravitational field measurements used in this example are shown in Figure 4.

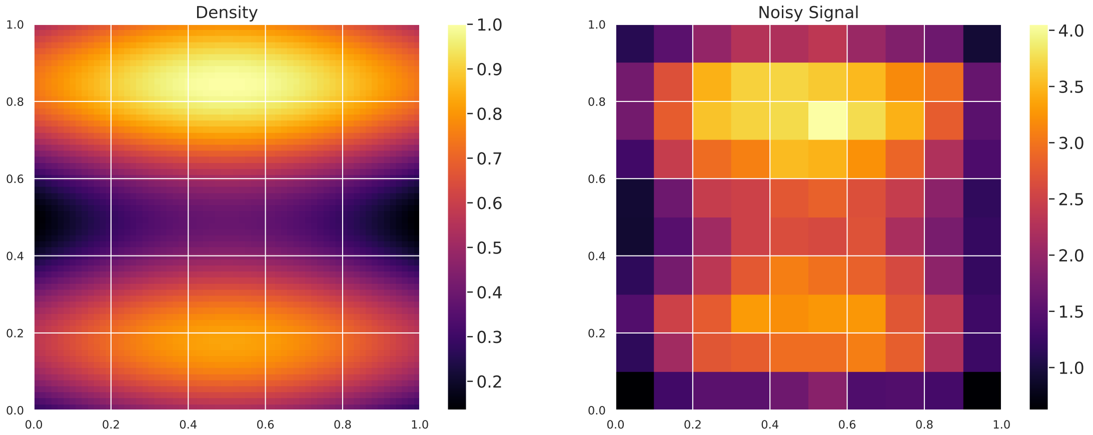


Figure 4: The true subsurface mass density field $\varrho(\boldsymbol{x})$ (left panel), shown alongside the low resolution measurements of the gravitational field at the surface $\zeta(\boldsymbol{x})$ (right panel).

For the inference task we model the subsurface density as a GRF with a Matérn

3/2 covariance kernel,

$$C(\boldsymbol{x}, \boldsymbol{x}') = \left(1 + \frac{\sqrt{3}\,\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\ell}\right) \exp\left(-\frac{\sqrt{3}\,\|\boldsymbol{x} - \boldsymbol{x}'\|_2}{\ell}\right), \tag{47}$$

where ℓ is the correlation length scale. The subsurface density field is parameterized using a KL expansion of the $R = 60$ leading eigenmodes,

$$\varrho(\boldsymbol{x}) = \mu_K + \sigma_K \sum_{k=1}^{R=60} \sqrt{\lambda_k}\phi_k(\boldsymbol{x})\theta_k, \tag{48}$$

where μ_K and σ_K^2 are the field mean and variance respectively, $\{\lambda_k\}_{k=1}^{R=60}$ is a sequence of strictly decreasing, real and positive eigenvalues of the covariance kernel in Equation 47, $\phi_k(\boldsymbol{x})$ are the corresponding eigenfunctions of the covariance kernel and $\{\theta_k \sim \mathcal{N}(0,1)\}_{k=1}^{R=60}$ are a set of standard Gaussian random variables. Defining $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{60})^{\intercal}$, the full model for this example is given by

$$\mu_K \sim \mathcal{N}(\mu = 0, \sigma^2 = 1^2), \tag{49}$$

$$\sigma_K \sim |\mathcal{N}(\mu = 0, \sigma^2 = 0.2^2)|, \tag{50}$$

$$\boldsymbol{\theta} \sim \mathcal{N}(0, I_{60}), \tag{51}$$

$$\boldsymbol{y} \sim \mathcal{N}(F_\zeta(\mu_K, \sigma_K, \boldsymbol{\theta}), \sigma_\eta^2 I_{100}) \tag{52}$$

where $F_\zeta(\mu_K, \sigma_K, \boldsymbol{\theta})$ denotes the full gravity survey forward model, mapping from the subsurface mass density field to the low resolution surface measurements of the gravitational field $\boldsymbol{y}$. When performing inference a log-transformation is applied to σ_K such that all parameters are in an unconstrained space, with the target distribution being modified by the corresponding Jacobian. The target dimension for this problem is $d = 62$.

We start again with tests where we enforce a fixed computational budget at each temperature level. In Figures 5 and 6 we show the recovered estimates for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$ respectively, with box plots again showing the variation over the 10 runs for each algorithm and ensemble size, using a fixed computational budget at each temperature level. The mean and standard deviation for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, along with the number of temperature levels used by each each algorithm over the 10 runs are reported in Table 4 for tpCN and Table 5 for pCN. We report the mean and standard deviation for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, along with the number of embarrassingly parallel model evaluations N_{eval}/J, for the second set of tests, where we select the number of sampling iterations adaptively, in Table 6.

Given a fixed computational budget at each temperature level, we see again that tpCN is able to achieve a lower squared bias on the final ensembles for all adaptation methods. For SMC applied to tpCN, we find that the values for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$ are high and largely independent of the ensemble size. In this case, SMC requires significantly more tpCN iterations at each temperature level in order to correctly distribute the particle ensemble after the importance resampling step. In comparison, SKT is able to achieve a lower bias with the same computational budget being used at each temperature level. The particle ensemble obtained by the EKI update provides a better initialization and preconditioner for the tpCN updates compared to the importance resampled particle ensemble, achieving lower bias with fewer model evaluations.

A similar pattern is again observed when comparing the NF-SKT and NF-SMC algorithms. For larger ensemble sizes the NF is able to map the effective prior at each temperature level to a Gaussian latent space, where the target is approximately Gaussian. For the same computational budget at each temperature level, the NF preconditioning more rapidly distributes particles according to the given target, with the low bias threshold being reached for an ensemble size of $J = 10d$ for the NF-SKT algorithm. For smaller ensemble sizes, the NF is unable to learn useful non-Gaussian features in the geometry of the effective prior, meaning we do not obtain an improvement from NF preconditioning.

When we allow for adaptive selection of the number of sampling iterations we again find that we are able to reach the low bias regime with NF-SKT and SKT adaptation of the tpCN sampler. This is not the case for NF-SMC and SMC adaptation applied to tpCN, or for the pCN sampler using all adaptation algorithms. In these cases, a more stringent adaptation criterion is again required to achieve low bias estimates of posterior moments.

5.3. Reaction-Diffusion Equation

We consider a reaction-diffusion system in one spatial dimension, where some quantity $s(x,t)$ varies with time under the action of some source term $u(x)$. This time evolution is described by a nonlinear reaction-diffusion equation of the form

$$\frac{\partial s(x,t)}{\partial t} = D\frac{\partial^2 s(x,t)}{\partial x^2} + \gamma s^2(x,t) + u(x), \quad x \in \Omega = [0,1], \tag{53}$$

where $D = 0.1$ is the diffusion constant and $\gamma = 0.1$ is the reaction rate. For this problem, we study the recovery of the source function $u(x)$ from observations of $s(x,t)$. To solve Equation 53 we use the implicit, second-order finite difference scheme implemented in [73]. We assume Dirichlet boundary conditions such that $s(x,t) = 0, \forall x \in \partial\Omega$, and the initial condition $s(x, t = 0) = 0$. The solution to Equation 53 is evaluated on a 100×100 grid in (x,t), up to a final time $t_f = 1$.

We parameterize the source term using the Hilbert space expansion of a Gaussian Process (GP) [74] with a squared exponential kernel,

$$u(x) = \mu_H + \sum_{j=1}^{R} \left[S_\Theta\left(\sqrt{\lambda_j}\right)\right]^{1/2} \phi_j(x)\theta_j, \tag{54}$$

where μ_H is the Hilbert space GP mean, $S_\Theta(\omega) = \alpha_H\sqrt{2\pi}\ell_H \exp(-\ell_H^2\omega^2/2)$ is the squared exponential kernel spectral density function, $\Theta = (\alpha_H, \ell_H)$ denotes the kernel

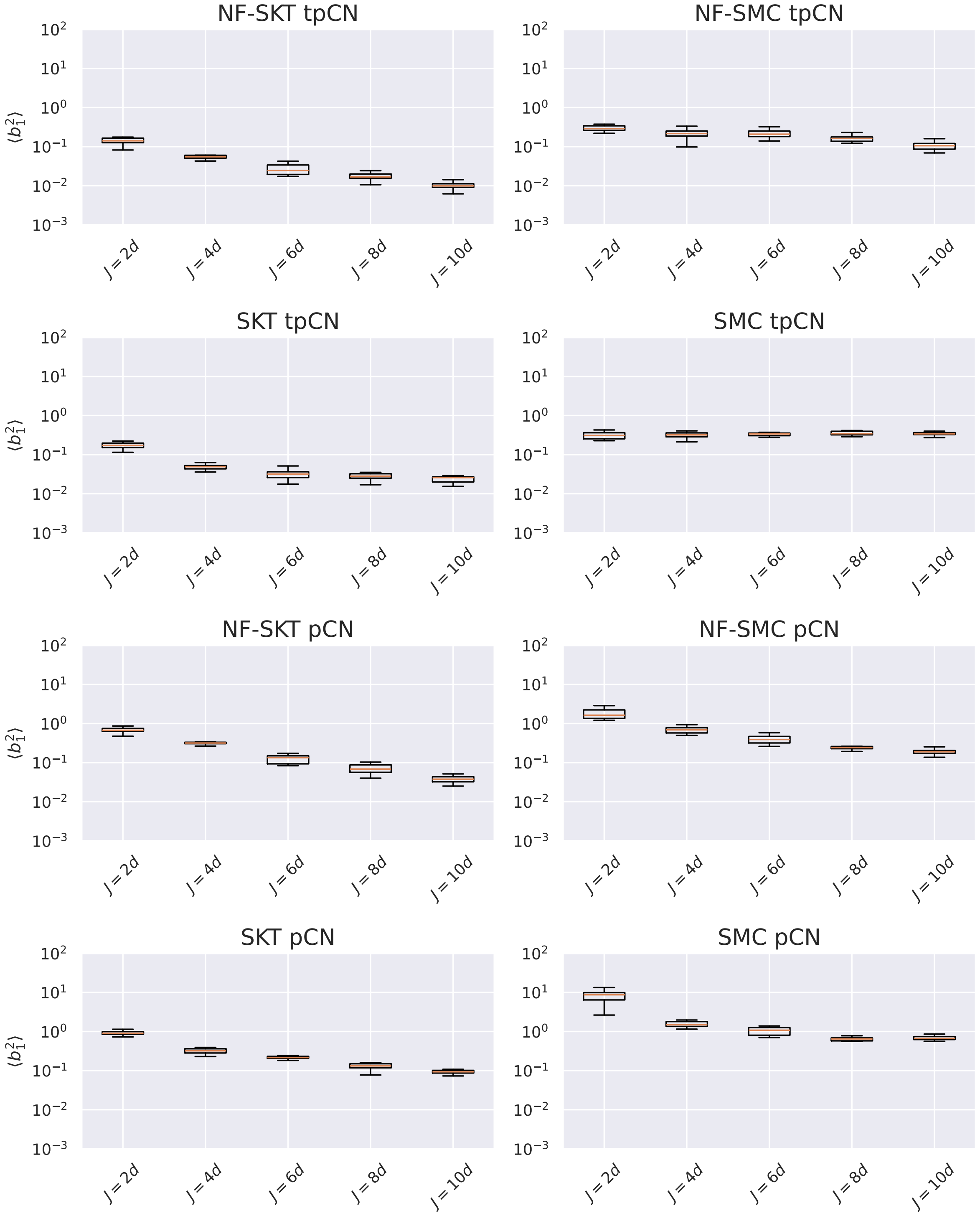


Figure 5: Final dimension averaged squared bias on the first moment for the gravity survey model, plotted against the ensemble size J expressed as a multiple of the target dimension $d = 62$. Results are shown for each adaptation algorithm applied to the tpCN and pCN samplers.

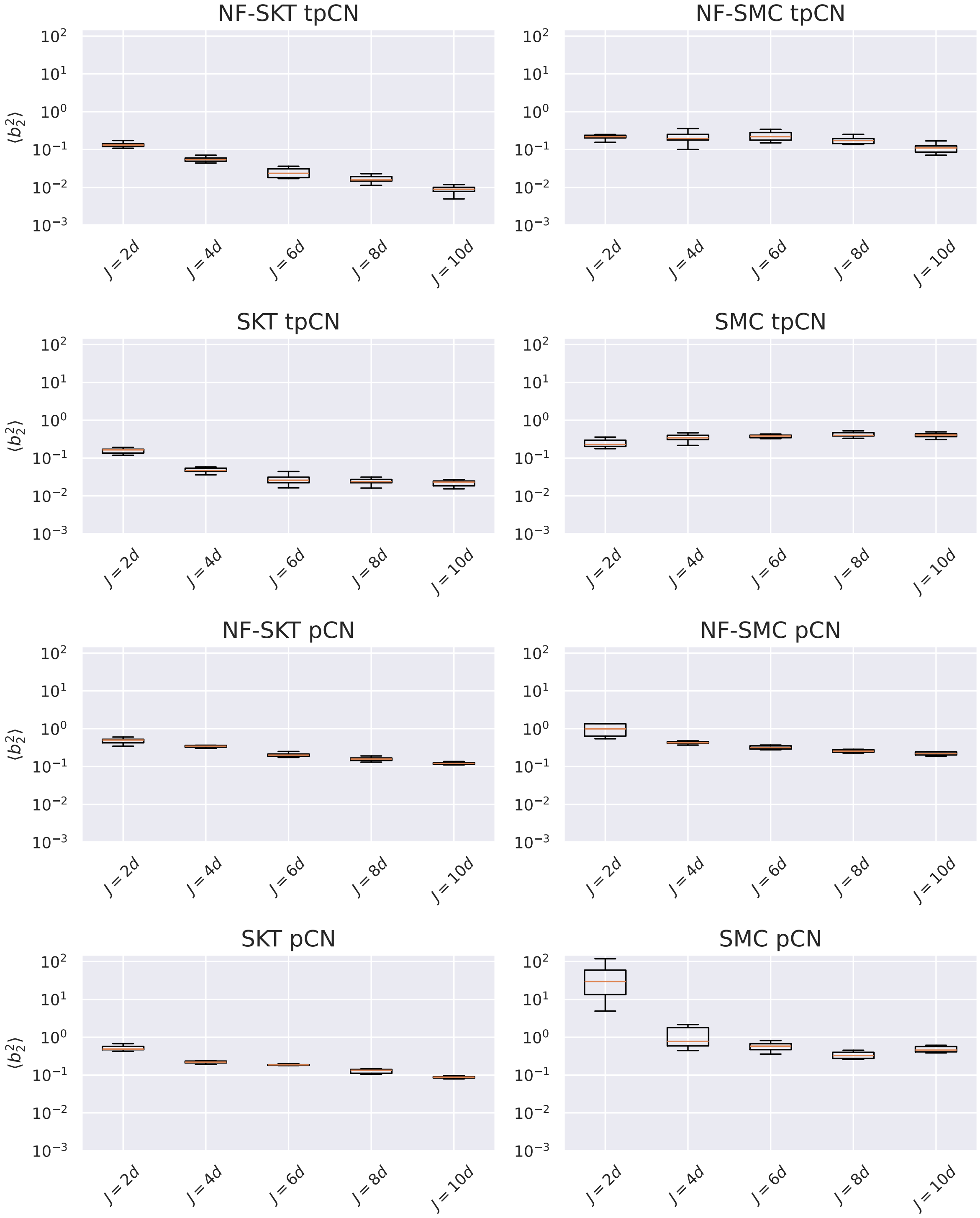


Figure 6: Final dimension averaged squared bias on the second moment for the gravity survey model, plotted against the ensemble size J expressed as a multiple of the target dimension $d = 62$. Results are shown for each adaptation algorithm applied to the tpCN and pCN samplers.

Algorithm	J	N_β	$\langle b_1^2\rangle$	$\langle b_2^2\rangle$
NF-SKT tpCN	$10d$	22.6 ± 0.5	0.0098 ± 0.0024	0.0089 ± 0.0025
NF-SKT tpCN	$8d$	23.0 ± 0.0	0.017 ± 0.004	0.017 ± 0.003
NF-SKT tpCN	$6d$	23.0 ± 0.0	0.027 ± 0.009	0.025 ± 0.007
NF-SKT tpCN	$4d$	23.0 ± 0.0	0.058 ± 0.011	0.056 ± 0.008
NF-SKT tpCN	$2d$	23.2 ± 0.4	0.14 ± 0.03	0.13 ± 0.02
SKT tpCN	$10d$	23.1 ± 0.3	0.024 ± 0.005	0.022 ± 0.004
SKT tpCN	$8d$	23.0 ± 0.0	0.028 ± 0.006	0.024 ± 0.004
SKT tpCN	$6d$	23.2 ± 0.4	0.032 ± 0.009	0.027 ± 0.008
SKT tpCN	$4d$	23.3 ± 0.5	0.050 ± 0.010	0.049 ± 0.009
SKT tpCN	$2d$	23.4 ± 0.5	0.17 ± 0.03	0.16 ± 0.02
NF-SMC tpCN	$10d$	23.0 ± 0.0	0.11 ± 0.03	0.11 ± 0.03
NF-SMC tpCN	$8d$	23.8 ± 0.4	0.17 ± 0.05	0.19 ± 0.05
NF-SMC tpCN	$6d$	24.0 ± 0.0	0.22 ± 0.05	0.23 ± 0.06
NF-SMC tpCN	$4d$	24.1 ± 0.3	0.22 ± 0.07	0.21 ± 0.07
NF-SMC tpCN	$2d$	23.2 ± 0.3	0.31 ± 0.07	0.23 ± 0.08
SMC tpCN	$10d$	23.9 ± 0.3	0.35 ± 0.05	0.42 ± 0.07
SMC tpCN	$8d$	24.0 ± 0.0	0.35 ± 0.05	0.42 ± 0.06
SMC tpCN	$6d$	24.2 ± 0.4	0.34 ± 0.07	0.39 ± 0.10
SMC tpCN	$4d$	24.0 ± 0.6	0.31 ± 0.06	0.34 ± 0.08
SMC tpCN	$2d$	23.5 ± 0.5	0.32 ± 0.07	0.25 ± 0.06

Table 4: Results for the number of temperature levels used by each algorithm N_β and squared bias results, $\langle b_1^2\rangle$ and $\langle b_2^2\rangle$, obtained with the final particle ensemble for each algorithm, when performing inference on the gravity survey example. We report the mean and standard deviation for each statistic over the 10 algorithm runs, and show results for each of the tested ensemble sizes J. The number of parallelized model evaluations is given by $11N_\beta$ for each algorithm, with the total number of model evaluations being given by $11JN_\beta$. The target dimension is $d = 62$.

hyperparmeters i.e., the kernel variance α_H and length scale ℓ_H, $\{\lambda_j\}_{j=1}^{\infty}$ and $\{\phi_j(x)\}_{j=1}^{\infty}$ are the eigenvalues and eigenfunctions of the Laplacian operator on some domain $\Omega_L = [-L, L]$ respectively, and $\theta_j \sim \mathcal{N}(0,1)$ are a set of standard Gaussian random variables. The eigenvalues and eigenfunctions of the Laplacian operator are given by

$$\lambda_j = \left(\frac{j\pi}{2L}\right)^2, \tag{55}$$

$$\phi_j(x) = \sqrt{\frac{1}{L}}\sin\left(\sqrt{\lambda_j}(x+L)\right). \tag{56}$$

Without loss of generality, we can evaluate $u(x)$ on the symmetric interval $[-0.5, 0.5]$, choosing the domain for the Laplacian operator $\Omega_L = [-1, 1]$ such that it contains the

Algorithm	J	N_β	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT pCN	$10d$	20.0 ± 0.0	0.040 ± 0.010	0.12 ± 0.01
NF-SKT pCN	$8d$	19.7 ± 0.5	0.070 ± 0.020	0.16 ± 0.02
NF-SKT pCN	$6d$	19.3 ± 0.5	0.13 ± 0.03	0.20 ± 0.02
NF-SKT pCN	$4d$	19.3 ± 0.5	0.33 ± 0.06	0.35 ± 0.04
NF-SKT pCN	$2d$	18.4 ± 0.5	0.69 ± 0.11	0.48 ± 0.08
SKT pCN	$10d$	24.6 ± 0.5	0.095 ± 0.014	0.16 ± 0.02
SKT pCN	$8d$	24.4 ± 0.5	0.13 ± 0.03	0.13 ± 0.02
SKT pCN	$6d$	23.6 ± 0.5	0.22 ± 0.03	0.18 ± 0.01
SKT pCN	$4d$	23.5 ± 0.5	0.34 ± 0.08	0.23 ± 0.03
SKT pCN	$2d$	22.0 ± 0.4	0.94 ± 0.15	0.52 ± 0.08
NF-SMC pCN	$10d$	21.0 ± 0.0	0.19 ± 0.04	0.22 ± 0.02
NF-SMC pCN	$8d$	21.5 ± 0.5	0.25 ± 0.04	0.26 ± 0.02
NF-SMC pCN	$6d$	22.3 ± 0.5	0.40 ± 0.10	0.32 ± 0.03
NF-SMC pCN	$4d$	23.7 ± 0.5	0.69 ± 0.14	0.44 ± 0.07
NF-SMC pCN	$2d$	29.2 ± 1.1	1.82 ± 0.58	1.27 ± 0.82
SMC pCN	$10d$	27.3 ± 0.6	0.69 ± 0.09	0.48 ± 0.09
SMC pCN	$8d$	28.2 ± 0.4	0.65 ± 0.07	0.34 ± 0.07
SMC pCN	$6d$	29.1 ± 0.5	1.05 ± 0.24	0.58 ± 0.15
SMC pCN	$4d$	31.4 ± 1.2	1.63 ± 0.45	1.11 ± 0.65
SMC pCN	$2d$	9.4 ± 1.6	8.28 ± 2.86	50.8 ± 51.0

Table 5: Results for the number of temperature levels used by each algorithm N_β and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm, when performing inference on the gravity survey example. We report the mean and standard deviation for each statistic over the 10 algorithm runs, and show results for each of the tested ensemble sizes J. The number of parallelized model evaluations is given by $11N_\beta$ for each algorithm, with the total number of model evaluations being given by $11JN_\beta$. The target dimension is $d = 62$.

full spatial domain of $u(x)$ [74].

To generate a simulated data set, we obtain a realisation of $u(x)$ from Equation 54 with $\mu_H = 0$, $\alpha_H = 1$ and $\ell_H = 0.1$. We solve for $s(x,t)$ subject to the corresponding Dirichlet boundary conditions, up to a time $t_f = 1$ on the 100×100 grid in (x,t). The field $s(x,t)$ is then observed at 10 equally spaced spatial locations, at 10 equally spaced times, with Gaussian observation noise corresponding to a noise standard deviation of $\sigma_\eta = 0.01$. The true source function is shown in Figure 7, alongside the corresponding solution for $s(x,t)$ and the locations of the $s(x,t)$ measurements.

For the inference task here we consider recovering the first $R = 50$ terms in the

Algorithm ($\tau_{\text{corr}} = 0.1$)	N_{eval}/J	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT tpCN	350 ± 100	0.0078 ± 0.0032	0.0072 ± 0.0032
NF-SKT pCN	420 ± 230	0.059 ± 0.011	0.12 ± 0.01
SKT tpCN	810 ± 90	0.0055 ± 0.0010	0.0060 ± 0.0012
SKT pCN	1200 ± 26	0.15 ± 0.03	0.12 ± 0.01
NF-SMC tpCN	320 ± 73	0.046 ± 0.023	0.046 ± 0.024
NF-SMC pCN	190 ± 6.6	0.15 ± 0.03	0.12 ± 0.01
SMC tpCN	630 ± 14	0.036 ± 0.008	0.037 ± 0.010
SMC pCN	1400 ± 43	0.69 ± 0.11	0.49 ± 0.09

Table 6: Results for the number of embarrassingly parallel model evaluations N_{eval}/J, and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm when adapting the number of sampling iterations at each temperature level using $\tau_{\text{corr}} = 0.1$, as applied to the gravity survey example. We show results when adapting both the tpCN and pCN samplers with an ensemble size of $J = 10d$, where the target dimension is $d = 62$.

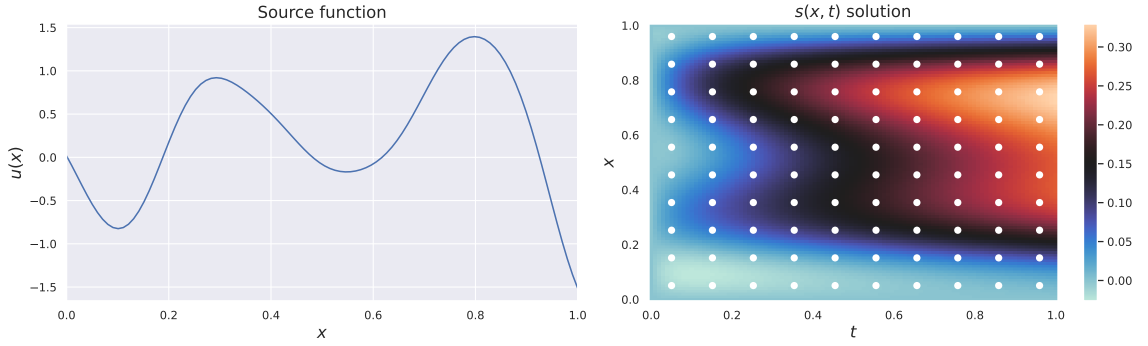


Figure 7: The true source term $u(x)$ (left panel), shown alongside the corresponding solution for $s(x, t)$ (right panel). White circles on the right panel denote the locations where measurements of $s(x, t)$ were made.

Hilbert space expansion. Denoting $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_{50})^\intercal$, the full model is given by

$$\mu_H \sim \mathcal{N}(\mu = 0, \sigma^2 = 0.1^2), \tag{57}$$
$$\alpha_H \sim |\mathcal{N}(\mu = 0, \sigma^2 = 1^2)|, \tag{58}$$
$$\ell_H \sim \text{InverseGamma}(\alpha = 4, \beta = 0.3), \tag{59}$$
$$\boldsymbol{\theta} \sim \mathcal{N}(0, I_{50}), \tag{60}$$
$$\boldsymbol{y} \sim \mathcal{N}(F_{RD}(\mu_H, \alpha_H, \ell_H, \boldsymbol{\theta}), \sigma_\eta^2 I_{100}), \tag{61}$$

where $F_{RD}(\mu_H, \alpha_H, \ell_H, \boldsymbol{\theta})$ denotes the full forward model, mapping from the source function $u(x)$ to the $s(x, t)$ observations $\boldsymbol{y}$. When running our set of inference algorithms, we apply log-transformations to α_H and ℓ_H such that all parameters are mapped to an unconstrained space, making the corresponding Jacobian adjustments to the target. The target dimension for this problem is $d = 53$.

For the first set of results we apply a fixed computational budget at each temperature level. In Figures 8 and 9 we show the recovered estimates for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$ respectively, with box plots showing the variation over the 10 runs for each algorithm and ensemble size. We report the corresponding mean and standard deviation for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, along with the mean and standard deviation on the number of temperature levels used by each algorithm over the 10 runs in Table 7 for tpCN, and in Table 8 for pCN. In Table 9 we report the mean and standard deviation for $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, along with the number of embarrassingly parallel model evaluations N_{eval}/J, for the second set of tests where we select the number of sampling iterations at each temperature level adaptively.

Whilst the values of $\langle b_1^2 \rangle$ are comparable for tpCN and pCN on this example, the values for $\langle b_2^2 \rangle$ are significantly lower for tpCN. The tpCN sampler is still able to better adapt to the targets at each temperature level, despite the target being closer to Gaussian. Comparing SKT with SMC, we can see that SKT is able to achieve significantly lower bias with the final particle ensemble using the same computational budget at each temperature level. This again indicates the the SKT update provides a better initialization and preconditioner for the subsequent tpCN iteration than the importance resampled ensemble in SMC. For this problem, we only obtain a small improvement in the final bias with NF preconditioning for larger ensemble sizes ($J \geq 6d$). The target posterior for this problem is close to Gaussian, meaning the NF map does not introduce a latent space where the target geometry is such that sampling is significantly easier.

When we make an adaptive selection of the number of sampling iterations, we find that we are able to reach the low bias regime for the tpCN sampler using the NF-SKT and SKT adaptation algorithms. This is not the case for the NF-SMC and SMC adaptation algorithms, or for the pCN sampler using any adaptation method. It is worth noting here that for pCN using SKT adaptation, for one of the random seeds the resultant pCN sampler was very poorly adapted to the target, resulting in high values for the mean and standard deviation of $\langle b_2^2 \rangle$. For NF-SMC and SMC adaptation applied to tpCN, and any adaptation method applied to pCN, we would again require a more stringent criterion for selecting the number of sampling iterations at each temperature level.

6. Conclusions

In this work we have considered the problem of performing Bayesian inference on inverse problems where the forward model is expensive to evaluate and we do not have access to derivatives of the forward model. In such a situation, standard sampling methods such as MCMC and SMC algorithms can quickly become intractable, requiring a large number of serial model evaluations to attain low bias estimates of posterior moments

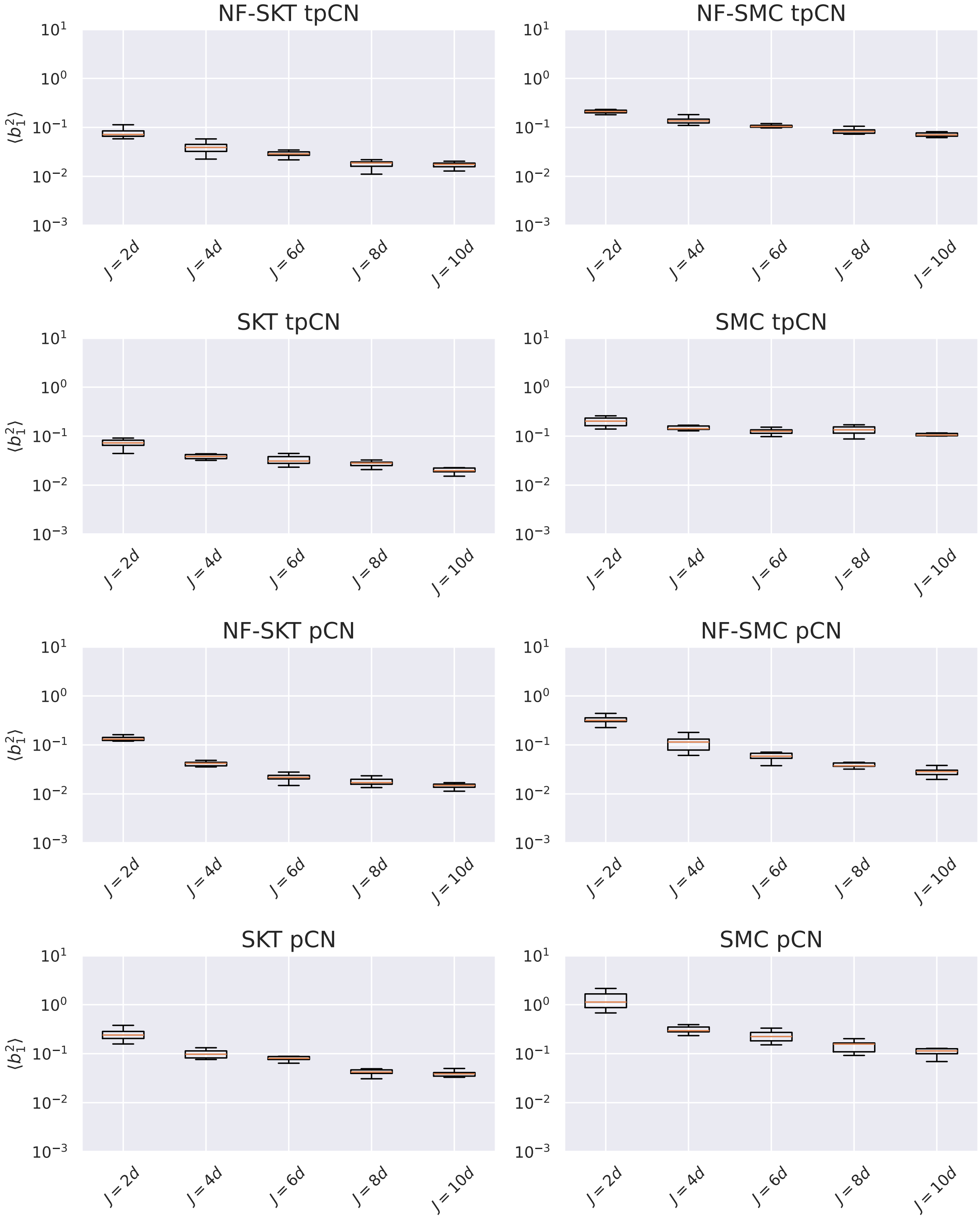


Figure 8: Final dimension averaged squared bias on the first moment for the reaction-diffusion model, plotted against the ensemble size J expressed as a multiple of the target dimension $d = 53$. Results are shown for each adaptation algorithm applied to the tpCN and pCN samplers.

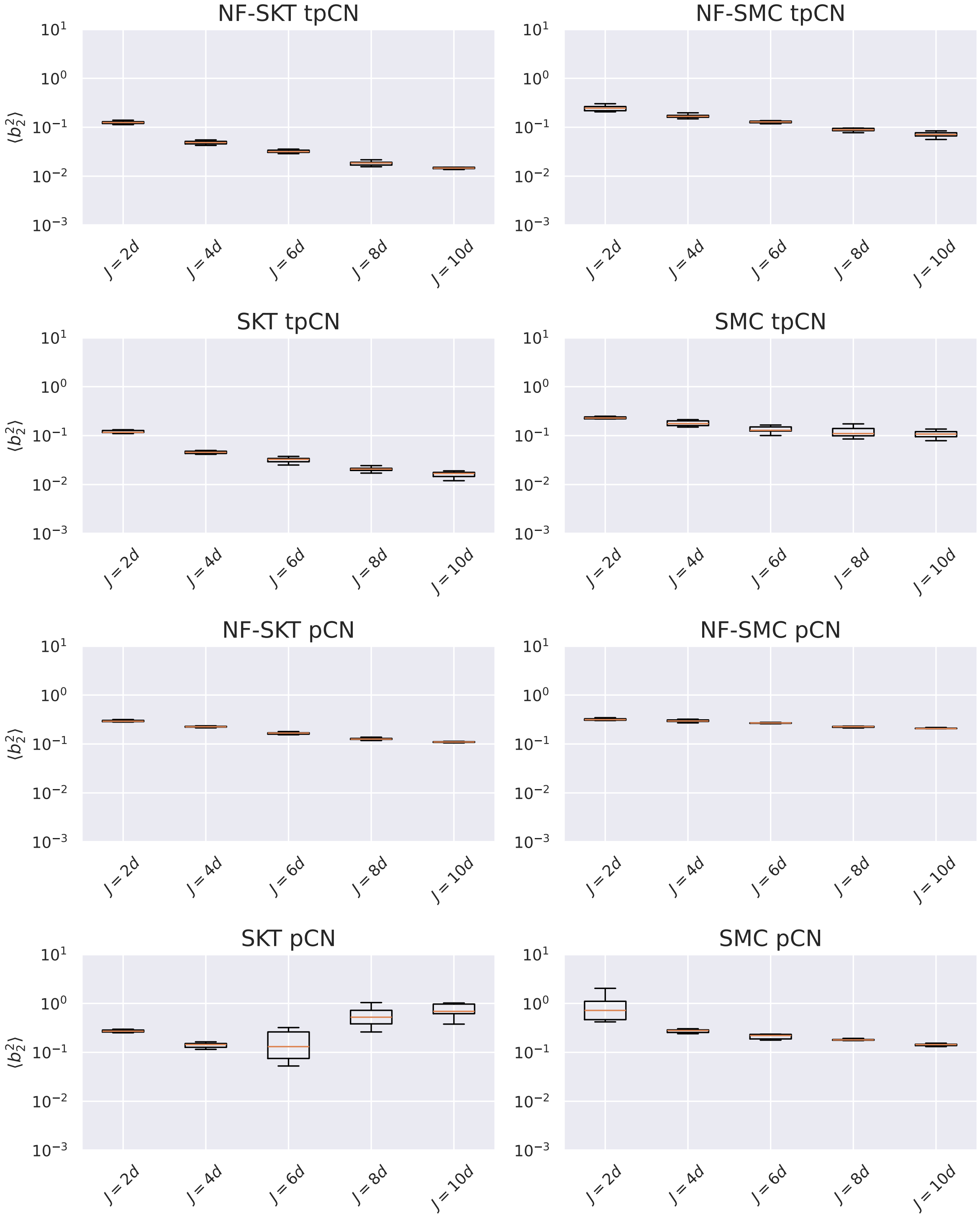


Figure 9: Final dimension averaged squared bias on the second moment for the reaction-diffusion model, plotted against the ensemble size J expressed as a multiple of the target dimension $d = 53$. Results are shown for each adaptation algorithm applied to the tpCN and pCN samplers.

Algorithm	J	N_β	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT tpCN	$10d$	17.0 ± 0.0	0.017 ± 0.002	0.015 ± 0.002
NF-SKT tpCN	$8d$	17.0 ± 0.0	0.018 ± 0.003	0.018 ± 0.002
NF-SKT tpCN	$6d$	17.8 ± 0.4	0.029 ± 0.004	0.032 ± 0.002
NF-SKT tpCN	$4d$	18.0 ± 0.0	0.040 ± 0.010	0.049 ± 0.004
NF-SKT tpCN	$2d$	19.1 ± 0.3	0.076 ± 0.016	0.13 ± 0.01
SKT tpCN	$10d$	17.0 ± 0.0	0.020 ± 0.003	0.016 ± 0.002
SKT tpCN	$8d$	17.1 ± 0.3	0.028 ± 0.004	0.021 ± 0.002
SKT tpCN	$6d$	17.6 ± 0.5	0.033 ± 0.007	0.032 ± 0.004
SKT tpCN	$4d$	18.0 ± 0.0	0.038 ± 0.004	0.046 ± 0.003
SKT tpCN	$2d$	19.0 ± 0.4	0.072 ± 0.013	0.12 ± 0.007
NF-SMC tpCN	$10d$	18.0 ± 0.0	0.072 ± 0.007	0.072 ± 0.008
NF-SMC tpCN	$8d$	18.6 ± 0.5	0.084 ± 0.010	0.093 ± 0.015
NF-SMC tpCN	$6d$	19.0 ± 0.0	0.11 ± 0.009	0.13 ± 0.01
NF-SMC tpCN	$4d$	19.8 ± 0.4	0.14 ± 0.02	0.18 ± 0.03
NF-SMC tpCN	$2d$	21.3 ± 0.5	0.21 ± 0.02	0.25 ± 0.03
SMC tpCN	$10d$	18.0 ± 0.0	0.11 ± 0.01	0.11 ± 0.02
SMC tpCN	$8d$	18.1 ± 0.3	0.13 ± 0.03	0.12 ± 0.03
SMC tpCN	$6d$	18.9 ± 0.3	0.13 ± 0.02	0.13 ± 0.02
SMC tpCN	$4d$	19.5 ± 0.5	0.15 ± 0.01	0.18 ± 0.02
SMC tpCN	$2d$	21.1 ± 0.5	0.20 ± 0.04	0.24 ± 0.07

Table 7: Results for the number of temperature levels used by each algorithm N_β and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm, when performing inference on the reaction-diffusion example. We report the mean and standard deviation for each statistic over the 10 algorithm runs, and show results for each of the tested ensemble sizes J. The number of parallelized model evaluations is given by $11N_\beta$ for each algorithm, with the total number of model evaluations being given by $11JN_\beta$. The target dimension is $d = 53$.

[16]. In contrast, EKI methods have been proposed that can rapidly converge on an ensemble approximation to the target posterior. However, EKI as applied to the Bayesian inverse problem is only exact in the regime of Gaussian targets and linear forward models, otherwise giving an uncontrolled approximation. This is insufficient for many scientific inference tasks where we seek accurate uncertainty estimates and hence low bias estimates for higher order posterior moments.

To address this shortcoming, we proposed integrating EKI updates within an adaptive SMC framework, replacing the standard importance resampling step at each temperature level with an EKI update. Instead of relying solely on the EKI updates to approximate the posterior, it was used to adapt the proposal kernel of the t-preconditioned Crank-Nicolson (tpCN) sampler. In this way, the EKI approximation at

Algorithm	J	N_β	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT pCN	$10d$	16.0 ± 0.0	0.015 ± 0.002	0.11 ± 0.003
NF-SKT pCN	$8d$	16.0 ± 0.0	0.018 ± 0.003	0.13 ± 0.006
NF-SKT pCN	$6d$	15.9 ± 0.3	0.022 ± 0.004	0.16 ± 0.008
NF-SKT pCN	$4d$	15.5 ± 0.5	0.042 ± 0.005	0.23 ± 0.006
NF-SKT pCN	$2d$	14.9 ± 0.5	0.14 ± 0.04	0.30 ± 0.01
SKT pCN	$10d$	21.0 ± 0.9	0.038 ± 0.007	0.79 ± 0.32
SKT pCN	$8d$	21.4 ± 2.0	0.041 ± 0.008	0.23 ± 0.01
SKT pCN	$6d$	19.9 ± 1.0	0.082 ± 0.016	0.22 ± 0.25
SKT pCN	$4d$	18.6 ± 0.5	0.10 ± 0.02	0.13 ± 0.04
SKT pCN	$2d$	17.7 ± 0.6	0.25 ± 0.06	0.28 ± 0.02
NF-SMC pCN	$10d$	16.7 ± 0.5	0.018 ± 0.003	0.13 ± 0.006
NF-SMC pCN	$8d$	17.0 ± 0.0	0.041 ± 0.008	0.23 ± 0.01
NF-SMC pCN	$6d$	17.9 ± 0.5	0.061 ± 0.022	0.27 ± 0.008
NF-SMC pCN	$4d$	18.9 ± 0.7	0.11 ± 0.04	0.30 ± 0.02
NF-SMC pCN	$2d$	24.7 ± 0.8	0.35 ± 0.10	0.32 ± 0.03
SMC pCN	$10d$	19.8 ± 0.4	0.11 ± 0.02	0.14 ± 0.007
SMC pCN	$8d$	20.9 ± 0.7	0.14 ± 0.03	0.18 ± 0.005
SMC pCN	$6d$	21.5 ± 0.7	0.23 ± 0.06	0.21 ± 0.02
SMC pCN	$4d$	24.2 ± 1.0	0.10 ± 0.02	0.28 ± 0.03
SMC pCN	$2d$	11.1 ± 1.0	1.28 ± 0.49	0.86 ± 0.48

Table 8: Results for the number of temperature levels used by each algorithm N_β and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm, when performing inference on the reaction-diffusion example. We report the mean and standard deviation for each statistic over the 10 algorithm runs, and show results for each of the tested ensemble sizes J. The number of parallelized model evaluations is given by $11N_\beta$ for each algorithm, with the total number of model evaluations being given by $11JN_\beta$. The target dimension is $d = 53$.

each temperature level provides a highly effective initialization and preconditioner for the tpCN sampler. Moreover, performing tpCN sampling prevents the accumulation of errors that would result in EKI using the incorrect prior ensemble to approximate each annealed target. The tpCN proposal kernel is reversible with respect to the multivariate t-distribution, in contrast to the standard pCN proposal which is reversible with respect to the multivariate Gaussian. In this paper we have proposed the Sequential Kalman Tuning (SKT) adaptation scheme for tpCN, and its NF preconditioned variant NF-SKT, that provides an efficient, tuning-free sampler for the solution of Bayesian inverse problems.

We compared the performance of the SKT and NF-SKT adaptation schemes, applied to tpCN and pCN, with standard importance resampling SMC and NF-SMC,

Algorithm ($\tau_{\mathrm{corr}} = 0.1$)	N_{eval}/J	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
NF-SKT tpCN	510 ± 43	0.0089 ± 0.0010	0.0082 ± 0.0009
NF-SKT pCN	450 ± 140	0.013 ± 0.004	0.12 ± 0.003
SKT tpCN	600 ± 43	0.0094 ± 0.0008	0.0095 ± 0.0013
SKT pCN	980 ± 15	0.039 ± 0.011	0.76 ± 2.08
NF-SMC tpCN	430 ± 19	0.031 ± 0.005	0.024 ± 0.004
NF-SMC pCN	420 ± 62	0.039 ± 0.011	0.22 ± 0.009
SMC tpCN	700 ± 18	0.022 ± 0.005	0.033 ± 0.006
SMC pCN	1000 ± 0.0	0.12 ± 0.03	0.14 ± 0.008

Table 9: Results for the number of embarrassingly parallel model evaluations N_{eval}/J, and squared bias results, $\langle b_1^2 \rangle$ and $\langle b_2^2 \rangle$, obtained with the final particle ensemble for each algorithm when adapting the number of sampling iterations at each temperature level, as applied to the reaction-diffusion example. We show results when adapting both the tpCN and pCN samplers with an ensemble size of $J = 10d$, where the target dimension is $d = 53$.

running each algorithm on three Bayesian inverse problems. Across our numerical experiments the tpCN sampler out-performed standard pCN for all the adaptation schemes we considered. The more flexible tail behaviour behaviour of the tpCN kernel means it can be more readily adapted for sampling from non-Gaussian targets. When using the same computational budget at each temperature level, the SKT and NF-SKT adaptation schemes resulted in lower bias estimates for the first and second posterior moments, compared to adapting the tpCN kernel within a standard SMC or NF-SMC scheme. When we selected the number of sampling iterations adaptively the tpCN sampler was able to rapidly reach low bias when using the SKT and NF-SKT adaptation schemes. For adaptation in SMC and NF-SMC the tpCN sampler was not able to reach low bias, and the pCN sampler failed to reach low bias for all adaptation schemes. In these cases a more stringent criterion would need to be imposed when selecting the number of sampling iterations and therefore significantly more forward model evaluations.

It is worth noting that we achieved lower bias adapting tpCN with the SKT scheme compared to using NF-SMC, demonstrating the ability of the EKI ensemble update to provide an effective initialization and preconditioner for the subsequent sampling steps. This is particularly promising for regimes where learning high fidelity NF maps becomes intractable e.g., moving beyond $\mathcal{O}(100)$ dimensions, or where one wishes to avoid the additional computational overhead from NF training. When using SKT adaptation, NF preconditioning is primarily useful for inverse problems where the cost of NF training (typically of order seconds up to $\mathcal{O}(100)$ dimensions), is insignificant compared to the cost of forward model evaluations during sampling, and where one can afford ensemble sizes that are sufficiently large to learn nonlinear features in the target geometry.

Several avenues exist for extending this work. In the first instance we plan to incorporate SKT and NF-SKT adaptation for tpCN within the pocoMC sampling package [35, 42], which currently implements adaptive variants of SMC and NF-SMC for tpCN. It would be interesting to explore alternative NF architectures that are able to learn useful features in the target geometry with smaller ensemble sizes [46], and waste-free SMC methods that allow us to exploit the full sampling history in the SMC framework [34]. In this work we have only considered one variant of the EKI-type updates within SMC. It would be worth studying the performance of deterministic ensemble updates [14, 16], which have been shown to have superior empirical performance compared to the stochastic EKI update used in this work [16]. It would also be useful to consider extensions allowing for parameter dependent noise covariances [75] and general likelihoods [76].

One could also consider adaptation schemes for tpCN that fall outside the Bayesian annealing framework e.g., ensemble sampling schemes that directly target the full posterior [25], leveraging ideas from measure transport [77], directly fitting for NF approximations to the target [61] etc. However, such adaptation schemes would require careful study to ensure stable kernel tuning during the burn-in phase where samples are far from the typical set. This problem is avoided in our sequential approach, where transitioning through a sequence of annealed targets allows for stable tuning of the tpCN kernel. This is particularly enhanced by the use of EKI updates as part of the adaptation. Moving to a non-sequential approach would require modification if we still wished to exploit Kalman-based approximations as part of the adaptation process, for example leveraging the proposed update rules in [16].

Acknowledgements

This research was funded by NSFC (grant No. 12250410240) and the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Contract No. DE-AC02-05CH11231 at Lawrence Berkeley National Laboratory to enable research for Data-intensive Machine Learning and Analysis, and by NSF grant number 2311559. RDPG was supported by a Tsinghua Shui Mu Fellowship. The authors thank Qijia Jiang and David Nabergoj for helpful discussions.

Appendix A. Proof of Lemma 2.1

Proof. Consider the current location $\boldsymbol{x}$ and the proposal location $\boldsymbol{x}' = \boldsymbol{\mu}_s + \sqrt{1-\rho^2}(\boldsymbol{x} - \boldsymbol{\mu}_s) + \rho\sqrt{Z}\boldsymbol{W}$, where $Z^{-1} \sim \text{Gamma}(k = \frac{1}{2}(d+\nu_s), \theta = 2/(\nu_s + \langle \boldsymbol{x}, \boldsymbol{x}\rangle_s))$ and $\boldsymbol{W} \sim \mathcal{N}(0, \mathcal{C}_s)$.

We have that

$$\sqrt{\frac{(d+\nu_s)Z}{\nu_s + \langle \boldsymbol{x}, \boldsymbol{x}\rangle_s}}\boldsymbol{W} \sim t_{d+\nu_s}(0, \mathcal{C}_s). \tag{A.1}$$

Using the change of variables formula, we obtain the proposal transition kernel for the tpCN algorithm as

$$\mathcal{K}_t(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}') = \frac{\gamma_1}{\rho^d}\left(\frac{d+\nu_s}{\nu_s + \langle \boldsymbol{x}, \boldsymbol{x}\rangle_s}\right)^{d/2} [1+ \\ (\rho^2(\nu_s + \langle \boldsymbol{x}, \boldsymbol{x}\rangle_s))^{-1}(\langle \boldsymbol{x}', \boldsymbol{x}'\rangle_s + (1-\rho^2)\langle \boldsymbol{x}, \boldsymbol{x}\rangle_s - 2\sqrt{1-\rho^2}\langle \boldsymbol{x}, \boldsymbol{x}'\rangle_s)\Big]^{-(2d+\nu_s)/2} \mathrm{d}\boldsymbol{x}', \tag{A.2}$$

where γ_1 is a normalizing constant. Considering the multivariate t-measure

$$p_s(\mathrm{d}\boldsymbol{x}) = \gamma_2\left[1 + \frac{\langle \boldsymbol{x}, \boldsymbol{x}\rangle_s}{\nu_s}\right]^{-(d+\nu_s)/2} \mathrm{d}\boldsymbol{x}, \tag{A.3}$$

where γ_2 is a normalizing constant, we have that

$$p_s(\mathrm{d}\boldsymbol{x})\mathcal{K}_t(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}') = \gamma_1\gamma_2\nu_s^{(d+\nu_s)/2}(d+\nu_s)^{d/2}\rho^{d+\nu_s}\left[\rho^2\nu_s + \langle \boldsymbol{x}', \boldsymbol{x}'\rangle_s + \langle \boldsymbol{x}, \boldsymbol{x}\rangle_s \right. \\ \left. -2\sqrt{1-\rho^2}\langle \boldsymbol{x}, \boldsymbol{x}'\rangle_s\right]^{-(2d+\nu_s)/2} \mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{x}' = p_s(\mathrm{d}\boldsymbol{x}')\mathcal{K}_t(\boldsymbol{x}', \mathrm{d}\boldsymbol{x}). \tag{A.4}$$

The variables $\boldsymbol{x}$ and $\boldsymbol{x}'$ are exchangeable in Equation A.4. Therefore the tpCN proposal transition kernel is reversible with respect to the multivariate t-distribution $t_{\nu_s}(\boldsymbol{\mu}_s, \mathcal{C}_s)$.

The tpCN acceptance probability follows from the fact that for some general proposal kernel, $\mathcal{K}(\boldsymbol{x}, \mathrm{d}\boldsymbol{x}')$ with probability density function $\kappa(\boldsymbol{x}, \boldsymbol{x}')$, the Metropolis-Hastings (MH) acceptance probability is given by

$$\alpha(\boldsymbol{x}, \boldsymbol{x}') = \min\left\{1, \frac{p(\boldsymbol{x}')\kappa(\boldsymbol{x}', \boldsymbol{x})}{p(\boldsymbol{x})\kappa(\boldsymbol{x}, \boldsymbol{x}')}\right\}. \tag{A.5}$$

From the reversibility expression in Equation A.4 we have that

$$\frac{\kappa_t(\boldsymbol{x}', \boldsymbol{x})}{\kappa_t(\boldsymbol{x}, \boldsymbol{x}')} = \frac{p_s(\boldsymbol{x})}{p_s(\boldsymbol{x}')} = \frac{(1 + \langle \boldsymbol{x}, \boldsymbol{x}\rangle_s/\nu_s)^{-(d+\nu_s)/2}}{(1 + \langle \boldsymbol{x}', \boldsymbol{x}'\rangle_s/\nu_s)^{-(d+\nu_s)/2}}, \tag{A.6}$$

which gives the MH acceptance probability in Equation 11. □

Appendix B. Importance Resampling

Given a set of samples and associated normalized importance weights $\{\boldsymbol{x}_n^i, \tilde{w}_n(\boldsymbol{x}_n^i)\}_{i=1}^J$, where $\tilde{w}_n(\boldsymbol{x}_n^i) = w_n(\boldsymbol{x}_n^i)/\sum_{k=1}^J w_n(\boldsymbol{x}_n^k)$, we can apply a resampling algorithm to obtain a set of equal weight samples. A simple approach would be to apply multinomial resampling, where the duplication counts for each member of the ensemble $\{N^1, \dots, N^J\}$ are obtained by sampling from the multinomial distribution $\mathrm{Mult}(J; \tilde{w}_n(\boldsymbol{x}_n^1), \dots, \tilde{w}_n(\boldsymbol{x}_n^J))$. Whilst multinomial resampling is straightforward, lower variance methods are available [54]. In this work we use systematic resampling for all our importance resampling SMC benchmarks. The systematic resampling algorithm pseudocode is given in Algorithm 4.

Algorithm 4 Systematic Resampling

```
**Input:** Set of $J$ samples and corresponding normalized importance weights $\{\boldsymbol{x}^i, \tilde{w}^i\}_{i=1}^J$.
Draw uniform random variable $U \sim \text{Unif}(0,1)$.
Set $U^i = U + (i-1)/J$, $i \in \{1, \dots, J\}$.
Set $D_w = \tilde{w}^0$.
Set index counter $k = 1$
**for** $i = 1, \dots, J$ **do**
    **while** $U^i > D_w$ **do**
        $k \leftarrow k + 1$
        $D_w \leftarrow D_w + \tilde{w}^k$
    **end while**
    Set $\tilde{\boldsymbol{x}}^i = \boldsymbol{x}^k$.
**end for**
**Output**: Equal weight particle ensemble $\{\tilde{\boldsymbol{x}}^i\}_{i=1}^J$.
```

Appendix C. Sequential Monte Carlo Implementations

In Algorithm 5 we give the pseudocode for the normalizing flow preconditioned SMC implementation, used as a benchmark for comparing the performance of the SKT samplers. The SMC implementation without NF preconditioning follows the same structure as Algorithm 5, without the NF fits such that the tpCN iterations are performed in the original data space. Similarly to the SKT samplers, we perform diminishing adaptation of the tpCN step size and reference measure mean at each temperature level.

Appendix D. Field and Source Term Reconstructions

In Figure D1 we show the true initial temperature field from the heat equation example in Section 5.1, alongside the reconstructed initial field from HMC samples, and the field reconstructions obtained by evaluating the average over the final particle ensembles for each of the NF-SKT, NF-SMC, SKT and SMC adaptation algorithms as applied to tpCN and pCN, for a single random seed initialization with an ensemble size $J = 10d$. We also show reconstructed fields obtained with the final FAKI and EKI ensembles. In Figure D2 we similarly show the true and reconstructed subsurface density fields for the gravity survey example in Section 5.2, and in Figure D3 we show the true and reconstructed source functions, $u(x)$ for the reaction-diffusion example in Section 5.3.

Appendix E. Converged Ensemble Corner Plots

In this section we provide corner plots showing the final particle ensembles over the first 4 dimensions, obtained by running each adaptation method on the tpCN and pCN

Algorithm 5 Flow Preconditioned Sequential Monte Carlo

```
Input: Set of $J$ samples from the prior $\{\boldsymbol{x}_0^i \sim \pi_0(\boldsymbol{x})\}_{i=1}^J$, data $\boldsymbol{y}$, observation covariance $\Gamma$, target fractional ESS $\tau$, number of tpCN iterations to perform at each temperature level $M$, initial tpCN step size $\rho$, target tpCN acceptance rate $\alpha^\star$, tpCN autocorrelation threshold $\tau_{\mathrm{corr}}$.
Set $\beta_0 = 0$ and iteration counter $n = 0$.
while $\beta_n < 1$ do
    Solve for target inverse temperature $\beta_{n+1}$ in Equation 27.
    $w_n(\boldsymbol{x}_n^i) \leftarrow \pi(\boldsymbol{y}|\boldsymbol{x}_n^i)^{\beta_{n+1}}/\pi(\boldsymbol{y}|\boldsymbol{x}_n^i)^{\beta_n}, \quad i \in \{1, \dots, J\}$.
    if $\beta_{n+1} = 1$ then
        $n^* \leftarrow n + 1$
    end if
    Fit NF map, $\boldsymbol{z} = f_n(\boldsymbol{x})$ to current particle locations $\{\boldsymbol{x}_n^i\}_{i=1}^J$.
    Obtain latent space particle locations $\{\boldsymbol{z}_n^i = f_n(\boldsymbol{x}_n^i)\}_{i=1}^J$.
    Resample weighted particles $\{\boldsymbol{z}_n^i, w_n(\boldsymbol{x}_n^i = f_n^{-1}(\boldsymbol{z}_n^i))\}$ using systematic resampling to give equal weight ensemble $\{\boldsymbol{z}_{n+1}^i\}_{i=1}^J$.
    Fit the multivariate $t$-distribution, $t_{\nu_s}(\mu_s, \mathcal{C}_s)$ to the latent space particle ensemble $\{\boldsymbol{z}_{n+1}^i\}_{i=1}^J$ with an EM algorithm. Set component-wise autocorrelations $\hat{\rho}_0(j) = 1\,\forall j$.
    for $m = 1, \dots, M$ do
        for $i = 1, \dots, J$ do
            Update particle state $\boldsymbol{z}_{n+1}^i$ using Algorithm 2 in NF latent space.
        end for
        $\log\rho \leftarrow \log\rho + (\langle\alpha\rangle - \alpha^\star)/m$
        $\boldsymbol{\mu}_s \leftarrow \boldsymbol{\mu}_s + (\langle\boldsymbol{z}_{n+1}\rangle - \boldsymbol{\mu}_{\boldsymbol{s}})/m$
        Calculate component-wise autocorrelations $\hat{\rho}_m(j)$.
        if $\prod_{l=1}^m \hat{\rho}_l(j) < \tau_{\mathrm{corr}}\,\forall j$ then
            End tpCN iterations.
        end if
    end for
    Map particle ensemble back to the original data space $\{\boldsymbol{x}_{n+1}^i = f_n^{-1}(\boldsymbol{z}_{n+1}^i)\}_{i=1}^J$
    $n \leftarrow n + 1$
end while
Output: Converged particle ensemble $\{\boldsymbol{x}_{n^*}^i\}_{i=1}^J$.
```

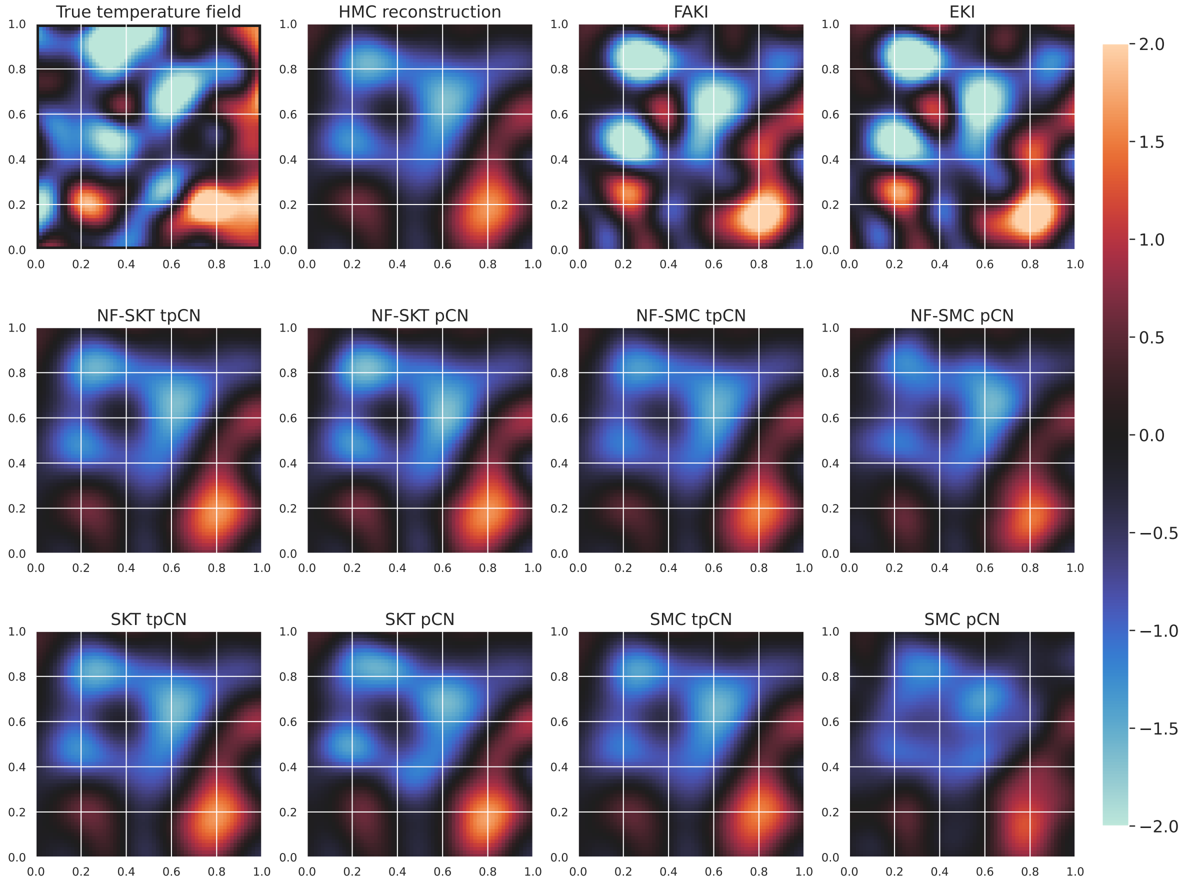


Figure D1: True initial temperature field $u(\boldsymbol{x}, t = 0)$ for the heat equation example (Section 5.1), plotted alongside the initial field reconstructions obtained using samples from HMC, which is treated as the target posterior predictive mean, and from the final particle ensembles of the FAKI, EKI, NF-SKT, NF-SMC, SKT and SMC algorithms (where results are shown for an ensemble size $J = 10d$, and for a single random seed initialization). Results for the adaptation algorithms are shown as applied to tpCN and pCN.

samplers, plotted alongside reference HMC samples. All the ensembles shown in this section were obtained using an ensemble size of $J = 10d$ and selecting the number of sampling iterations adaptively using an autocorrelation threshold of $\tau_{\mathrm{corr}} = 0.1$. The figures included in this section are as follows:

- Figure E1: Corner plot showing the final ensembles for the heat equation example, using NF-SKT and SKT adaptation applied to the tpCN and pCN samplers.
- Figure E2: Corner plot showing the final ensembles for the heat equation example, using NF-SMC and SMC adaptation applied to the tpCN and pCN samplers.
- Figure E3: Corner plot showing the final ensembles for the gravity survey example, using NF-SKT and SKT adaptation applied to the tpCN and pCN samplers.
- Figure E4: Corner plot showing the final ensembles for the gravity survey example,

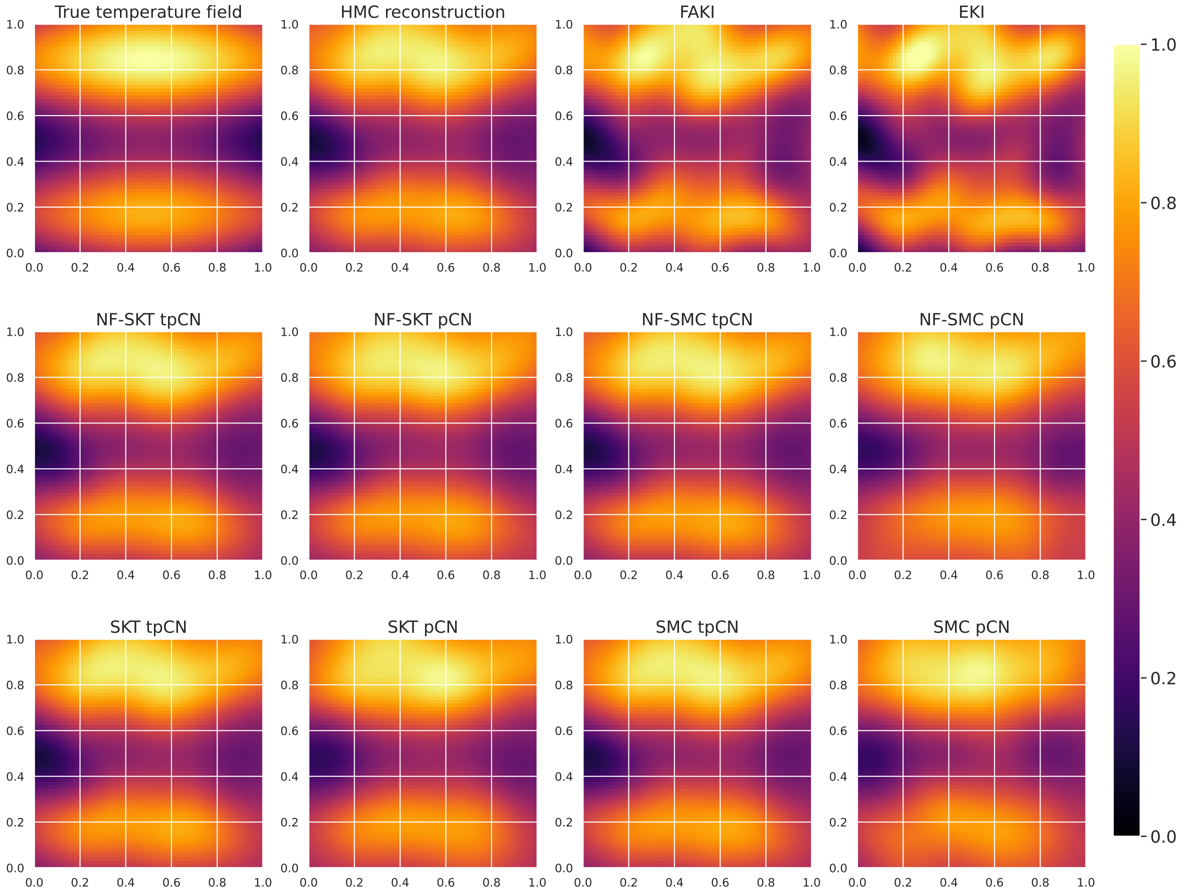


Figure D2: True subsurface mass density field $\varrho(\boldsymbol{x})$ for the gravity survey example (Section 5.2), plotted alongside the density field reconstructions obtained using samples from HMC, which is treated as the target posterior predictive mean, and from the final particle ensembles of the FAKI, EKI, NF-SKT, NF-SMC, SKT and SMC algorithms (where results are shown for an ensemble size $J = 10d$, and for a single random seed initialization). Results for the adaptation algorithms are shown as applied to tpCN and pCN.

using NF-SMC and SMC adaptation applied to the tpCN and pCN samplers.

- Figure E5: Corner plot showing the final ensembles for the reaction-diffusion example, using NF-SKT and SKT adaptation applied to the tpCN and pCN samplers.
- Figure E6: Corner plot showing the final ensembles for the reaction-diffusion example, using NF-SMC and SMC adaptation applied to the tpCN and pCN samplers.

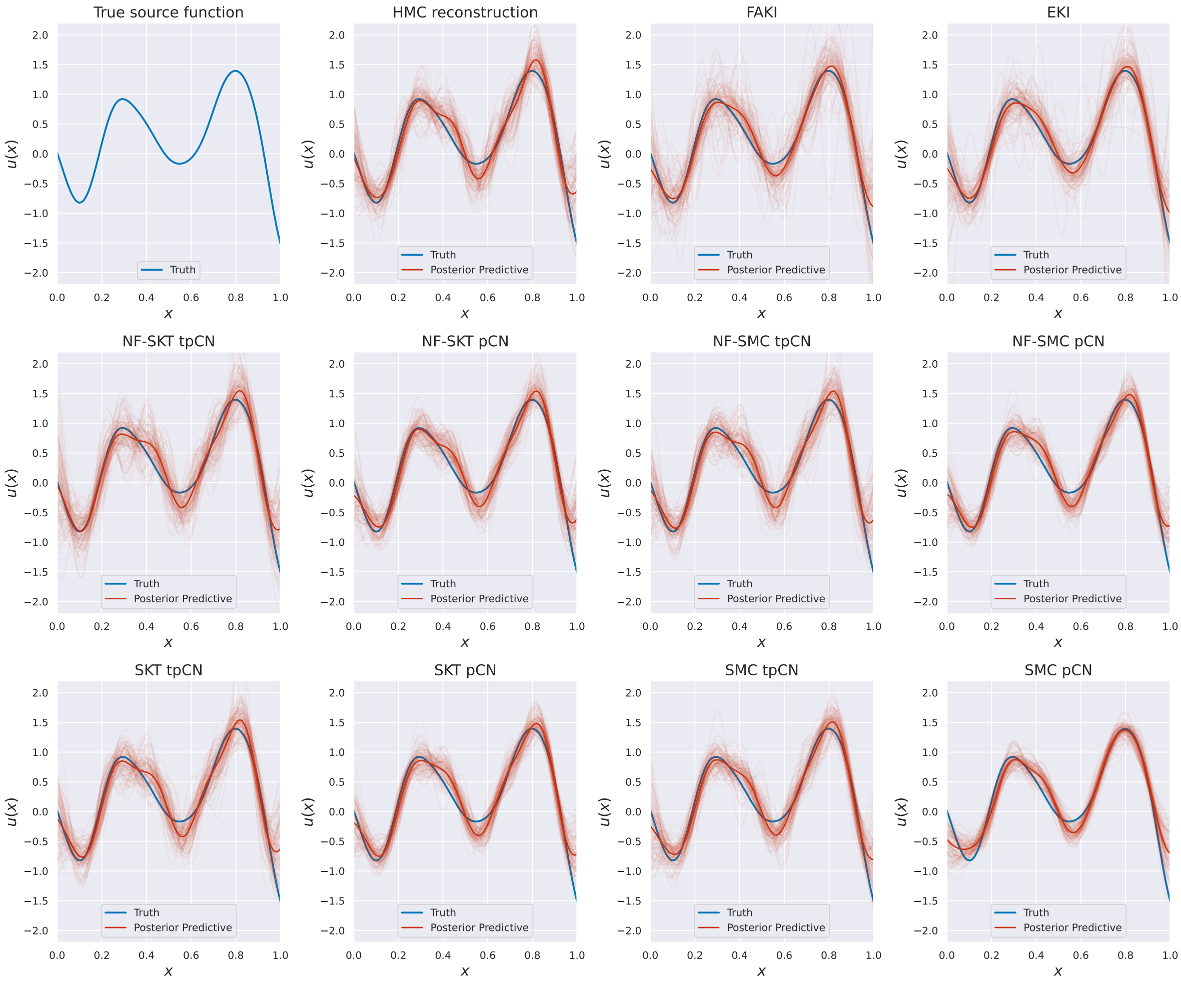


Figure D3: True source function $u(x)$ for the reaction-diffusion example (Section 5.3), plotted alongside posterior predictive samples obtained with HMC, which are treated as true posterior predictive samples, and from the final particle ensembles of the FAKI, EKI, NF-SKT, NF-SMC, SKT and SMC algorithms (where results are shown for an ensemble size $J = 10d$, and for a single random seed initialization). For each algorithm we show 100 posterior predictive samples, with the dark red line showing the estimated posterior predictive mean in each case. Results for the adaptation algorithms are shown as applied to tpCN and pCN.

Appendix F. Flow Annealed Kalman Inversion and Ensemble Kalman Inversion Ablation

In Figures F1, F2 and F3 we show corner plots of the recovered particle distributions from running EKI and FAKI on the heat equation, gravity survey and reaction-diffusion examples respectively. For all plots we also show the sample distributions from our reference HMC samples. We use an ensemble size of $J = 10d$ throughout for EKI and FAKI, set $\tau = 0.5$ for temperature level adaptation, and show the particle distributions over the first 4 dimensions for illustrative purposes. In Table F1 we state the average

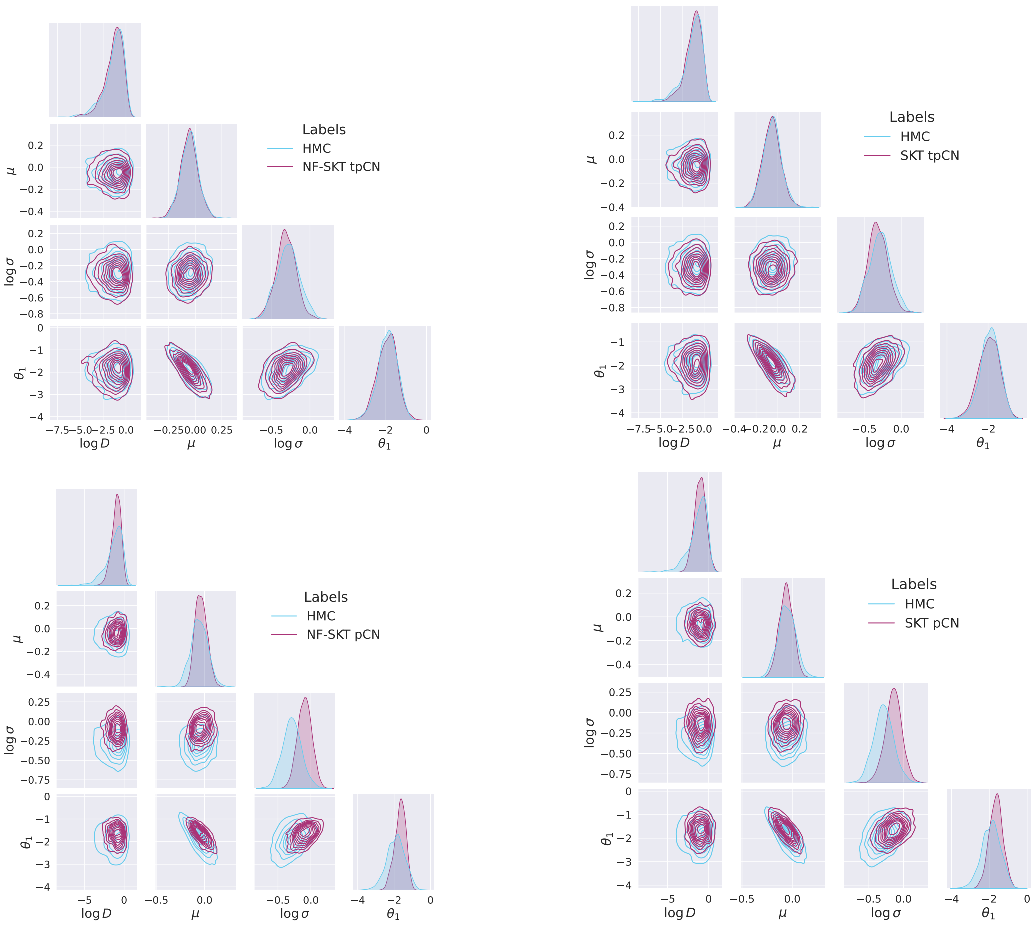


Figure E1: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using NF-SKT and SKT adaptation applied to the tpCN and pCN samplers, plotted alongside reference HMC samples for the heat equation example. The ensemble size was $J = 10d$, where $d = 103$.

number of iterations and bias-squared results for EKI and FAKI over 10 runs for each of the numerical examples.

From these corner plots, we can immediately see that the particle distributions from EKI and FAKI are strongly offset from the reference HMC sample distributions, which manifests in the high bias results reported in Table F1. For our numerical examples, we break the core assumptions underlying EKI. This means that in moving from the prior to the first annealed target, the updated ensemble will not be correctly distributed according to the annealed target. When updating the particles for the next temperature level, we do not have the correct effective prior ensemble, meaning these errors will accumulate as we move from the prior to the posterior. For FAKI, we still have these problems, given that the NF maps do not address any errors arising due to nonlinearity. If the particle ensemble is not correctly distributed at a given temperature level, the NF will not Gaussianize the correct effective prior, meaning we lose the additional benefits from mapping the particle ensemble to a Gaussian latent space at each iteration. These

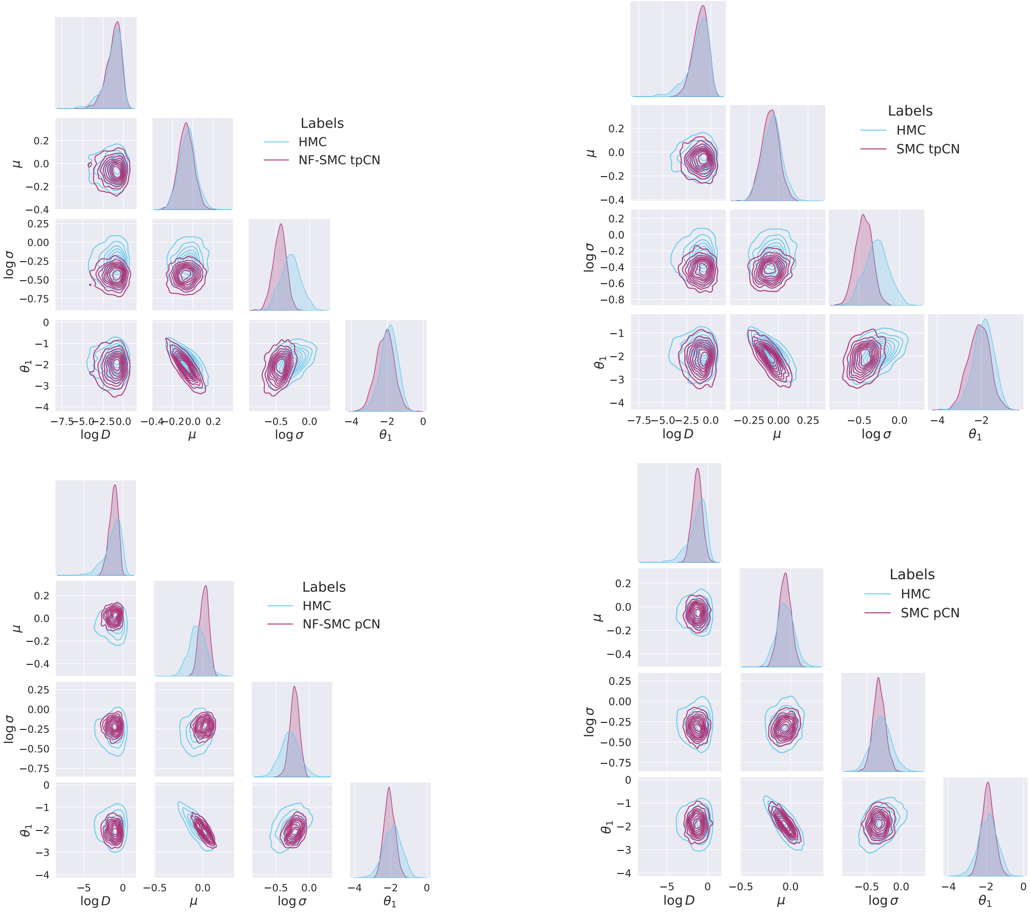


Figure E2: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using NF-SMC and SMC adaptation applied to the tpCN and pCN samplers, plotted alongside reference HMC samples for the heat equation example. The ensemble size was $J = 10d$, where $d = 103$.

results all demonstrate the importance of using the sampling iterations in SMC to correct the EKI and FAKI updates in order to obtain reliable estimates for posterior moments.

Appendix G. Ensemble Kalman Sampler

The idea of exploiting the EKI ensemble structure for preconditioning in sampling has previously been used in the Ensemble Kalman Sampler (EKS) [26]. Assuming that we have a Gaussian prior such that $\boldsymbol{x} \sim \mathcal{N}(0, \Gamma_0)$, EKS iterates over ensemble updates for the particle j given by,

$$\hat{\boldsymbol{x}}^j_{n+1} = \boldsymbol{x}^j_n - \frac{\Delta t_n}{J} \sum_{k=1}^{J} \left(\mathcal{F}(\boldsymbol{x}^k_n) - \bar{\mathcal{F}}_n\right)^\intercal \Gamma^{-1} \left(\mathcal{F}(\boldsymbol{x}^j_n) - \boldsymbol{y}\right) \boldsymbol{x}^k_n + \frac{d+1}{J}\left(\boldsymbol{x}^j_n - \bar{\boldsymbol{x}}_n\right) - \Delta t_n C^{\boldsymbol{x},\boldsymbol{x}}_n \Gamma_0^{-1} \hat{\boldsymbol{x}}^j_n, \quad \text{(G.1)}$$

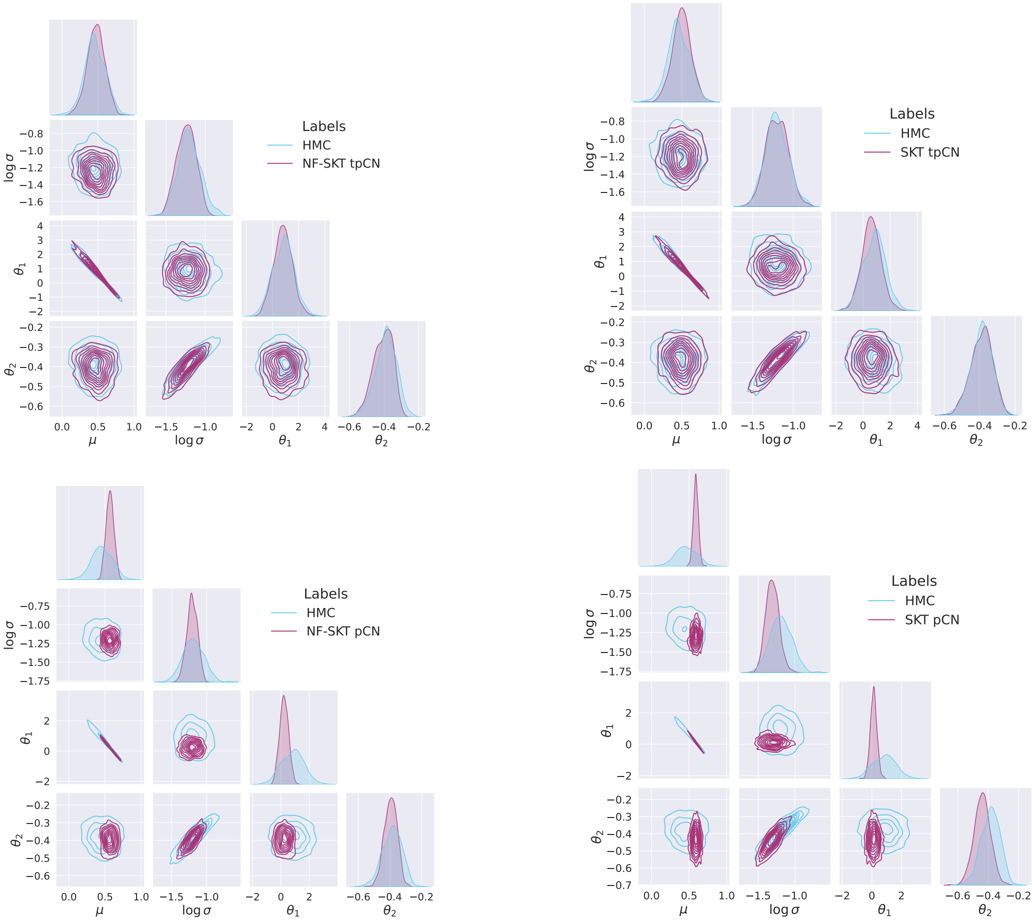


Figure E3: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using NF-SKT and SKT adaptation applied to the tpCN and pCN samplers, plotted alongside reference HMC samples for the gravity survey example. The ensemble size was $J = 10d$, where $d = 62$.

$$\boldsymbol{x}_{n+1}^{j} = \hat{\boldsymbol{x}}_{n}^{j} + \sqrt{2\Delta t_n C_n^{\boldsymbol{x},\boldsymbol{x}}}\,\boldsymbol{\xi}_n^j, \tag{G.2}$$

where $\bar{\mathcal{F}}_n = J^{-1}\sum_{k=1}^{J}\mathcal{F}(\boldsymbol{x}_n^k)$, $\bar{\boldsymbol{x}}_n = J^{-1}\sum_{k=1}^{J}\boldsymbol{x}_n^k$, $\boldsymbol{\xi}_n^j \sim \mathcal{N}(0, I_d)$ and

$$C_n^{\boldsymbol{x},\boldsymbol{x}} = \frac{1}{J}\sum_{k=1}^{J}\left(\boldsymbol{x}_n^k - \bar{\boldsymbol{x}}_n\right)\otimes\left(\boldsymbol{x}_n^k - \bar{\boldsymbol{x}}_n\right). \tag{G.3}$$

Following [26] the time step may be chosen adaptively such that

$$\Delta t_n = \frac{\Delta t_0}{\left\|D_n\right\|_F + \epsilon}, \tag{G.4}$$

where $\left\|\cdot\right\|_F$ denotes the Frobenius norm, $\Delta t_0 = 1$ and $\epsilon = 10^{-5}$. The matrix $D_n \in \mathbb{R}^{J\times J}$ is defined as

$$D_n = \frac{1}{J}(F_n - \bar{F}_n)\Gamma^{-1}(F_n - Y)^{\intercal}, \tag{G.5}$$

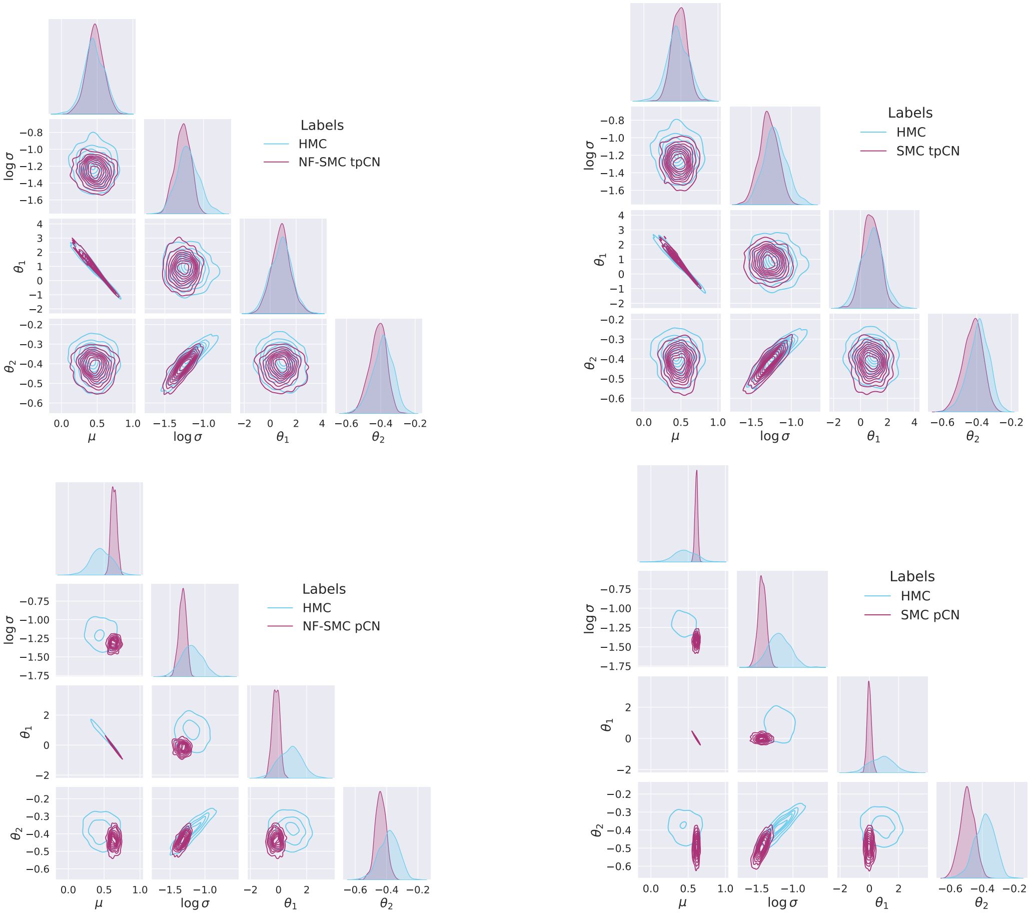


Figure E4: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using NF-SMC and SMC adaptation applied to the tpCN and pCN samplers, plotted alongside reference HMC samples for the gravity survey example. The ensemble size was $J = 10d$, where $d = 62$.

where $F_n \in \mathbb{R}^{J\times n_y}$ is a matrix where row k contains the vector $\mathcal{F}(\boldsymbol{x}_n^k)$, $\bar{F}_n \in \mathbb{R}^{J\times n_y}$ is a matrix where every row contains the vector $\bar{\mathcal{F}}_n$, and $Y \in \mathbb{R}^{J\times n_y}$ is a matrix where every row contains the data vector $\boldsymbol{y}$.

The EKS ensemble will converge to an approximation of the posterior, which is only exact for linear forward models. In contrast to EKI, the prior is explicitly accounted for in the update equations, and noise is added in parameter space as opposed to data space. We ran EKS on each of our heat equation, gravity survey and reaction-diffusion examples. Given the EKS updates assume a Gaussian prior, we began by fitting an NF to the prior samples before performing EKS updates in the NF latent space. The EKS sampler was run with an ensemble size of $J = 100d$ for each problem. This very large ensemble size was required to ensure the numerical stability of the EKS updates over a large number of iterations. Using an ensemble size of e.g., $J = 10d$ resulted in serious numerical instabilities that meant the update procedure failed within ~ 10 iterations.

In Figure G1 we show corner plots comparing the ensemble distributions obtained

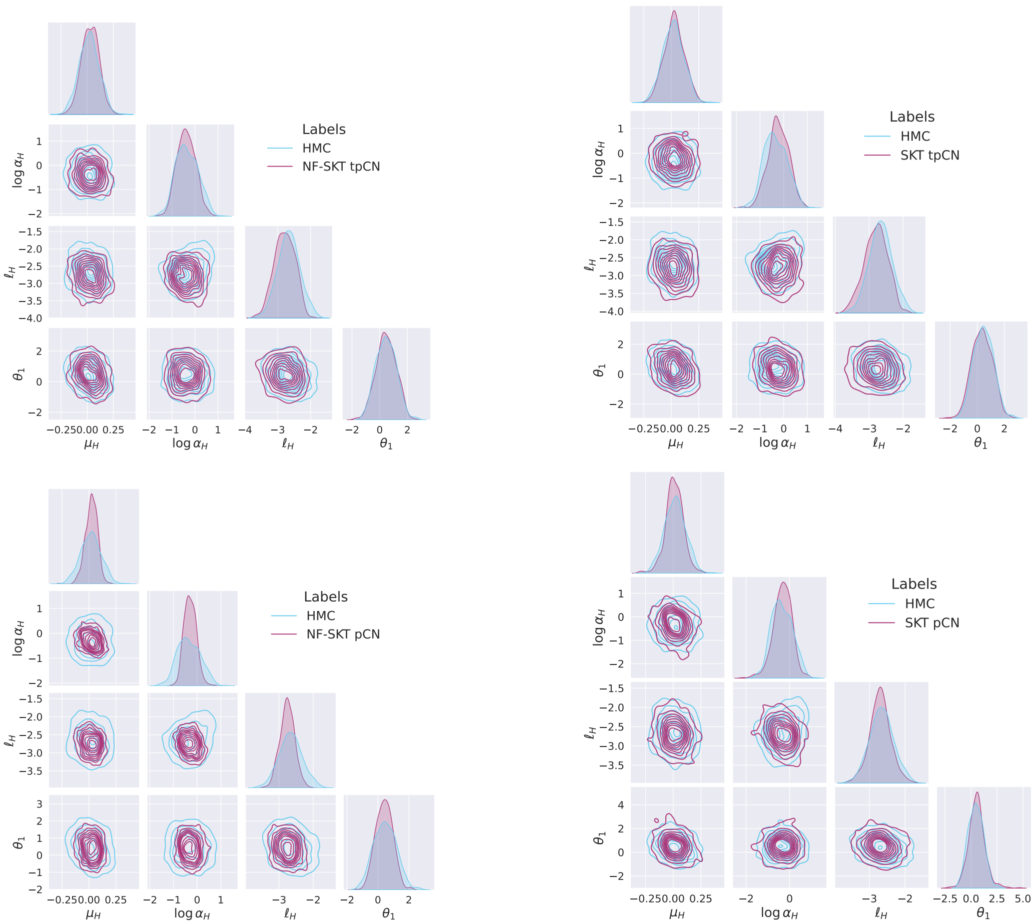


Figure E5: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using NF-SKT and SKT adaptation applied to the tpCN and pCN samplers, plotted alongside reference HMC samples for the reaction-diffusion example. The ensemble size was $J = 10d$, where $d = 53$.

with EKS after 100 iterations with reference samples obtained using HMC. For all three experiments EKS converges on a highly biased approximation to the posterior. In contrast, the tpCN sampler is able to preserve the exact target as its invariant measure, and when run with the SKT scheme is able to rapidly converge on low bias estimates of posterior moments.

References

[1] Kaipio J and Somersalo E 2005 *Statistical and Computational Inverse Problems* (Springer)

[2] MacKay D J C 2003 *Information theory, inference, and learning algorithms* (Cambridge University Press)

[3] Lewis A and Bridle S 2002 *Physical Review D* **66** 103511

[4] Blas D, Lesgourgues J and Tram T 2011 *Journal of Cosmology and Astroparticle Physics* **2011** 034

[5] Jasak H, Jemcov A and Tukovic Z 2007 Openfoam: A c++ library for complex physics simulations

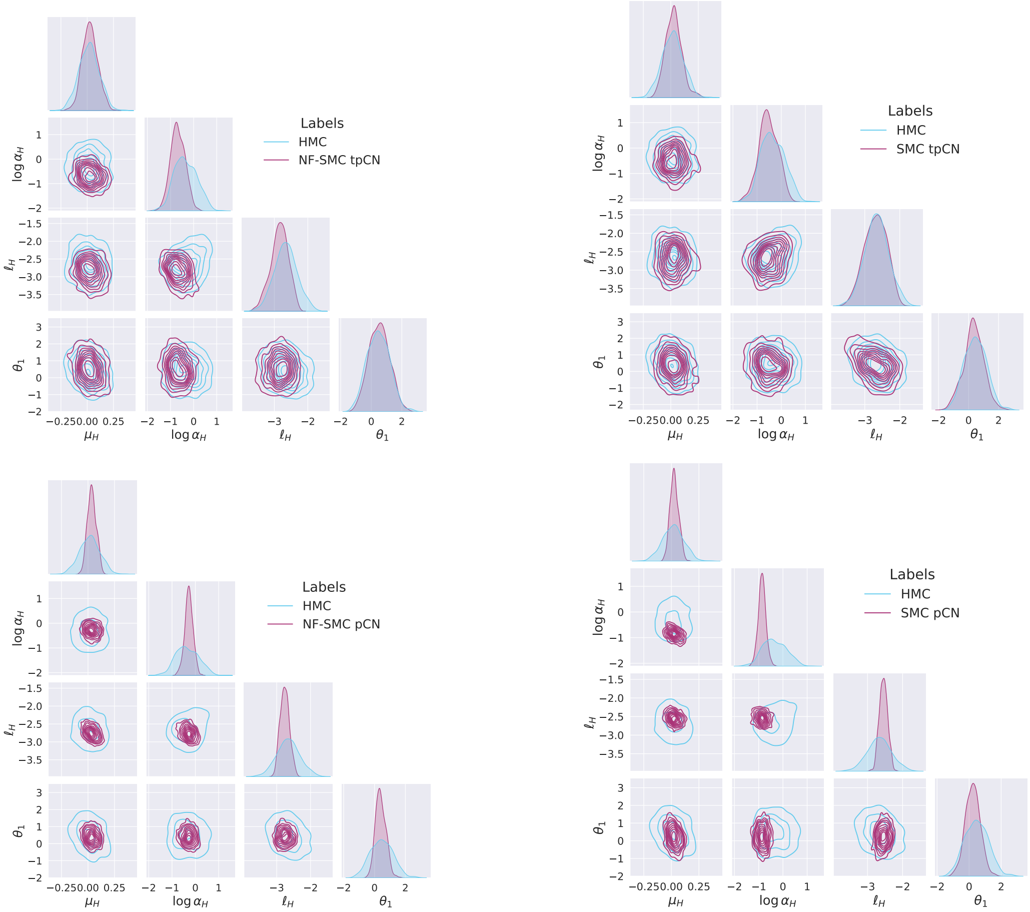


Figure E6: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using NF-SMC and SMC adaptation applied to the tpCN and pCN samplers, plotted alongside reference HMC samples for the reaction-diffusion example. The ensemble size was $J = 10d$, where $d = 53$.

International workshop on coupled methods in numerical dynamics, volume 1000, pages 1–20. IUC Dubrovnik Croatia

[6] Tan Z, Kaul C M, Pressel K G, Cohen Y, Schneider T and Teixeira J 2018 *Journal of Advances in Modeling Earth Systems* **10** 770–800

[7] Iglesias M A, Law K J and Stuart A M 2013 *Inverse Problems* **29** 045001

[8] Iglesias M A 2016 *Inverse Problems* **32** 025002

[9] Iglesias M, Park M and Tretyakov M 2018 *Inverse Problems* **34** 105002

[10] Chada N K, Iglesias M A, Roininen L and Stuart A M 2018 *Inverse Problems* **34** 055009

[11] Kovachki N B and Stuart A M 2019 *Inverse Problems* **35** 095005

[12] Chada N K, Stuart A M and Tong X T 2020 *SIAM Journal on Numerical Analysis* **58** 1263–1294

[13] Iglesias M and Yang Y 2021 *Inverse Problems* **37** 025008

[14] Ding Z, Li Q and Lu J 2021 *Foundations of Data Science* **3** 371–411

[15] Huang D Z, Schneider T and Stuart A M 2022 *Journal of Computational Physics* **463** 111262

[16] Huang D Z, Huang J, Reich S and Stuart A M 2022 *Inverse Problems* **38** 125006

[17] Chada N and Tong X 2022 *Mathematics of Computation* **91** 1247–1280

[18] Grumitt R D P, Karamanis M and Seljak U 2024 Flow annealed kalman inversion for gradient-free inference in bayesian inverse problems *Physical Sciences Forum* vol 9 (MDPI) p 21

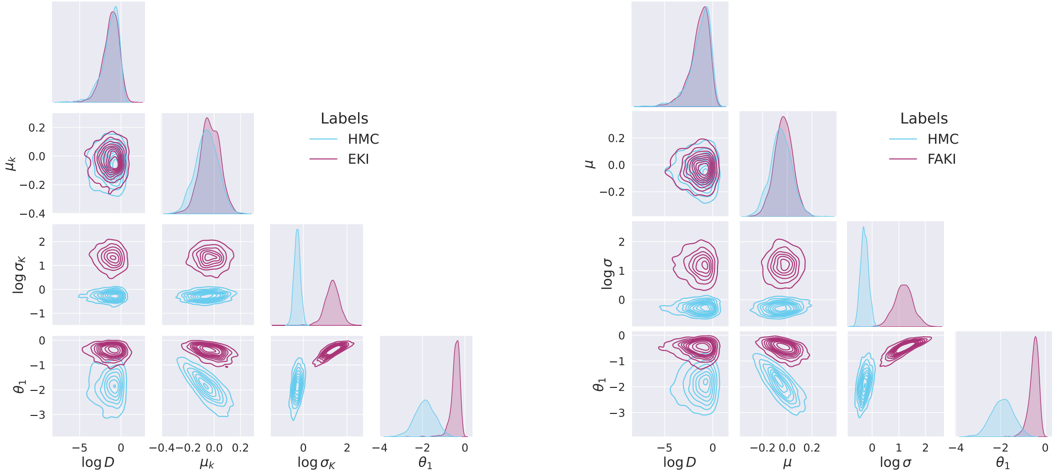


Figure F1: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using EKI (left panel) and FAKI (right panel), plotted alongside reference HMC samples for the heat equation example. The ensemble size was $J = 10d$, where $d = 103$.

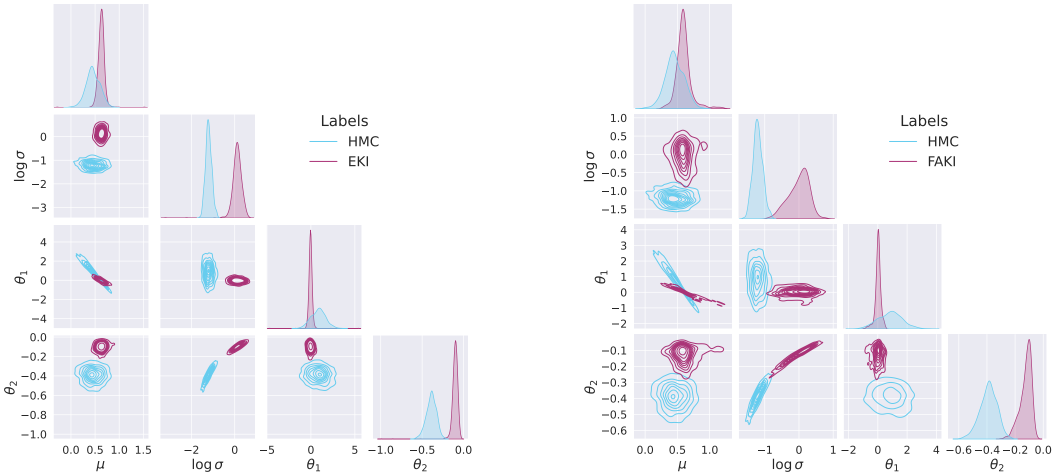


Figure F2: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using EKI (left panel) and FAKI (right panel), plotted alongside reference HMC samples for the gravity survey example. The ensemble size was $J = 10d$, where $d = 62$.

[19] Geyer C J 1992 *Statistical Science* **7** 473–483 ISSN 08834237

[20] Gelman A, Gilks W R and Roberts G O 1997 *The Annals of Applied Probability* **7** 110 – 120

[21] Neal R M *et al.* 2011 *Handbook of Markov Chain Monte Carlo* **2** 2

[22] Cotter S L, Roberts G O, Stuart A M and White D 2013 *Statistical Science* **28** 424 – 446

[23] Vrugt J, ter Braak C, Diks C, Robinson B, Hyman J and Higdon D 2009 *International Journal of Nonlinear Sciences and Numerical Simulation* **10** 273–290 ISSN 1565-1339

[24] Foreman-Mackey D, Hogg D W, Lang D and Goodman J 2013 *Publications of the Astronomical Society of the Pacific* **125** 306

[25] Leimkuhler B, Matthews C and Weare J 2018 *Statistics and Computing* **28** 277–290

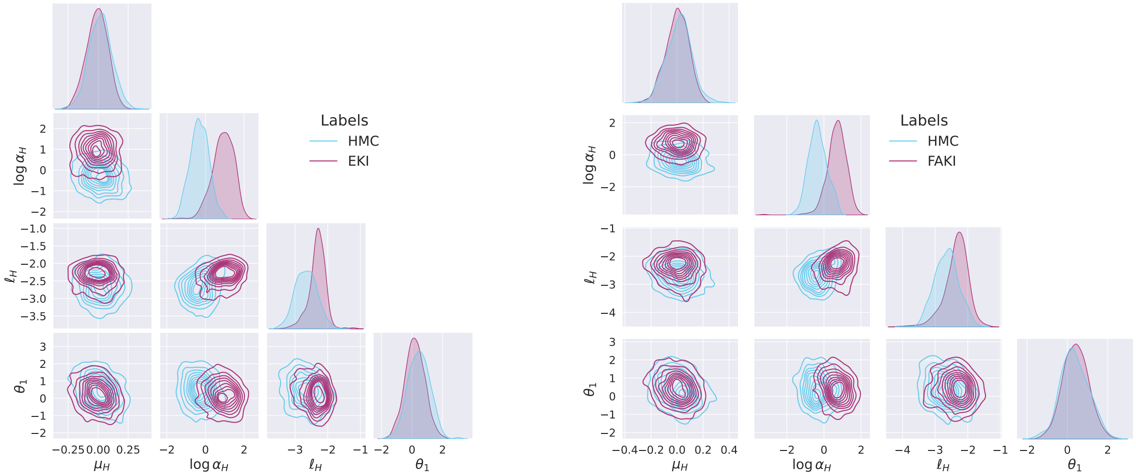


Figure F3: Corner plots showing the final particle ensemble for the first 4 dimensions, obtained using EKI (left panel) and FAKI (right panel), plotted alongside reference HMC samples for the reaction-diffusion example. The ensemble size was $J = 10d$, where $d = 53$.

Experiment	N_β	$\langle b_1^2 \rangle$	$\langle b_2^2 \rangle$
Heat Equation FAKI	26.2 ± 0.6	1.72 ± 0.12	3.25 ± 0.64
Heat Equation EKI	37.4 ± 1.5	2.07 ± 0.13	5.36 ± 1.01
Gravity Survey FAKI	29.9 ± 0.7	3.02 ± 0.32	1.12 ± 0.03
Gravity Survey EKI	54.9 ± 8.5	3.23 ± 0.26	1.13 ± 0.03
Reaction-Diffusion FAKI	35.3 ± 3.2	0.17 ± 0.02	0.07 ± 0.01
Reaction-Diffusion EKI	34.1 ± 4.4	0.21 ± 0.02	0.13 ± 0.03

Table F1: Number of iterations (N_β) and bias-squared results for the final ensembles obtained using FAKI and EKI on the heat equation, gravity survery and reaction-diffusion examples. The ensemble size was $J = 10d$, where $d = 103$ for the heat equation example, $d = 62$ for the gravity survey example and $d = 53$ for the reaction-diffusion example.

[26] Garbuno-Inigo A, Hoffmann F, Li W and Stuart A M 2020 *SIAM Journal on Applied Dynamical Systems* **19** 412–441

[27] Karamanis M and Beutler F 2021 *Statistics and Computing* **31** 1–18

[28] Grumitt R D P, Dai B and Seljak U 2022 *Advances in Neural Information Processing Systems* **35** 11629–11641

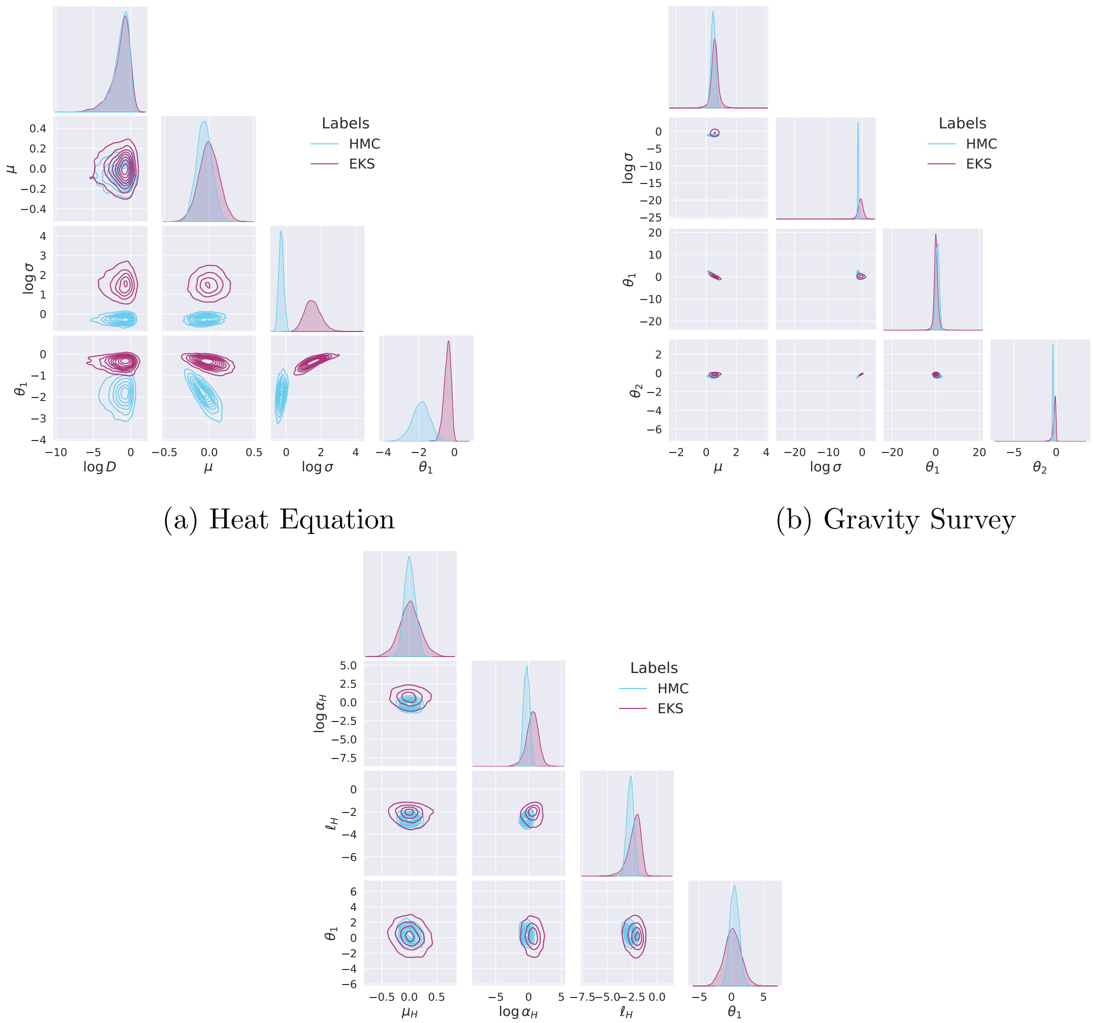
[29] Evensen G 2006 *Data Assimilation: The Ensemble Kalman Filter* (Berlin, Heidelberg: Springer-Verlag) ISBN 354038300X

[30] Evensen G 2009 *IEEE Control Systems Magazine* **29** 83–104

[31] Schillings C and Stuart A M 2017 *SIAM Journal on Numerical Analysis* **55** 1264–1290

[32] Del Moral P, Doucet A and Jasra A 2006 *Journal of the Royal Statistical Society Series B: Statistical Methodology* **68** 411–436 ISSN 1369-7412

[33] Wu J, Wen L, Green P L, Li J and Maskell S 2022 *Statistics and Computing* **32** 20



(a) Heat Equation

(b) Gravity Survey

(c) Reaction-Diffusion

Figure G1: Corner plots showing the particle distributions over the first four dimensions obtained after 100 iterations of EKS, shown alongside reference samples from long runs of HMC. For all examples we use an ensemble size of $J = 100d$, where $d = 103$ for the heat equation example, $d = 62$ for the gravity survey example, and $d = 53$ for the reaction-diffusion example.

[34] Dau H D and Chopin N 2022 *Journal of the Royal Statistical Society Series B: Statistical Methodology* **84** 114–148

[35] Karamanis M, Beutler F, Peacock J A, Nabergoj D and Seljak U 2022 *Monthly Notices of the Royal Astronomical Society* **516** 1644–1653

[36] Hoffman M D, Gelman A *et al.* 2014 *Journal of Machine Learning Research* **15** 1593–1623

[37] Hoffman M, Radul A and Sountsov P 2021 An adaptive-mcmc scheme for setting trajectory lengths in hamiltonian monte carlo *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics* (*Proceedings of Machine Learning Research* vol 130) ed Banerjee A and Fukumizu K (PMLR) pp 3907–3915

[38] Sountsov P and Hoffman M D 2021 *arXiv preprint arXiv:2110.11576*

[39] Hoffman M D and Sountsov P 2022 Tuning-free generalized hamiltonian monte carlo *Proceedings*

of The 25th International Conference on Artificial Intelligence and Statistics (*Proceedings of Machine Learning Research* vol 151) ed Camps-Valls G, Ruiz F J R and Valera I (PMLR) pp 7799–7813

[40] Riou-Durand L, Sountsov P, Vogrinc J, Margossian C and Power S 2023 Adaptive tuning for metropolis adjusted langevin trajectories *International Conference on Artificial Intelligence and Statistics* (PMLR) pp 8102–8116

[41] Buchholz A, Chopin N and Jacob P E 2021 *Bayesian Analysis* **16** 745–771

[42] Karamanis M, Nabergoj D, Beutler F, Peacock J and Seljak U 2022 *The Journal of Open Source Software* **7** 4634 (*Preprint* `2207.05660`)

[43] Dinh L, Sohl-Dickstein J and Bengio S 2017 Density estimation using real NVP *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings* (OpenReview.net)

[44] Papamakarios G, Murray I and Pavlakou T 2017 Masked autoregressive flow for density estimation *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA* ed Guyon I, von Luxburg U, Bengio S, Wallach H M, Fergus R, Vishwanathan S V N and Garnett R pp 2338–2347

[45] Kingma D P and Dhariwal P 2018 Glow: Generative flow with invertible 1x1 convolutions *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada* ed Bengio S, Wallach H M, Larochelle H, Grauman K, Cesa-Bianchi N and Garnett R pp 10236–10245

[46] Dai B and Seljak U 2021 Sliced iterative normalizing flows *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event* (*Proceedings of Machine Learning Research* vol 139) ed Meila M and Zhang T (PMLR) pp 2352–2364

[47] Kamatani K 2018 *Bernoulli* **24** 3711 – 3750

[48] Kamatani K 2017 *Journal of Applied Probability* **54** 638–654

[49] Hairer M, Stuart A M and Vollmer S J 2014 *The Annals of Applied Probability* **24** 2455 – 2490

[50] Kamatani K and Uchida M 2015 *Statistical Inference for Stochastic Processes* **18** 177–204

[51] Nishihara R, Murray I and Adams R P 2014 *The Journal of Machine Learning Research* **15** 2087–2112

[52] Mandel J, Cobb L and Beezley J D 2011 *Applications of Mathematics* **56** 533–541

[53] Chopin N 2002 *Biometrika* **89** 539–551 ISSN 00063444

[54] Douc R and Cappé O 2005 Comparison of resampling schemes for particle filtering *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.* (Ieee) pp 64–69

[55] Zhang J, Vrugt J A, Shi X, Lin G, Wu L and Zeng L 2020 *Water Resources Research* **56** e2019WR025474

[56] Drovandi C, Everitt R G, Golightly A and Prangle D 2022 *Bayesian Analysis* **17** 223–260

[57] De Simon L, Iglesias M, Jones B and Wood C 2018 *Energy and Buildings* **177** 220–245

[58] Moral P D, Doucet A and Jasra A 2012 *Bernoulli* **18** 252–278 ISSN 13507265

[59] Beskos A, Jasra A, Kantas N and Thiery A 2016 *The Annals of Applied Probability* **26** 1111–1146 ISSN 10505164

[60] Durkan C, Bekasov A, Murray I and Papamakarios G 2019 *Advances in neural information processing systems* **32**

[61] Gabrié M, Rotskoff G M and Vanden-Eijnden E 2022 *Proceedings of the National Academy of Sciences* **119** e2109420119

[62] Wong K W K, Gabrié M and Foreman-Mackey D 2022 *arXiv e-prints* arXiv:2211.06397 (*Preprint* `2211.06397`)

[63] Roberts G O and Rosenthal J S 2001 *Statistical Science* **16** 351 – 367

[64] Beskos A, Pillai N, Roberts G, Sanz-Serna J M and Stuart A 2013 *Bernoulli* **19** 1501–1534 ISSN 13507265

[65] Hoffman M, Sountsov P, Dillon J V, Langmore I, Tran D and Vasudevan S 2019 *arXiv e-prints* arXiv:1903.03704 (*Preprint* `1903.03704`)

[66] Meng X L and van Dyk D A 1997 *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **59**

[67] Liu C and Rubin D B 1995 *Statistica Sinica* **5** 19–39

[68] Roberts G O and Rosenthal J S 2007 *Journal of Applied Probability* **44** 458–475 ISSN 00219002

[69] Phan D, Pradhan N and Jankowiak M 2019 *arXiv preprint arXiv:1912.11554*

[70] Bingham E, Chen J P, Jankowiak M, Obermeyer F, Pradhan N, Karaletsos T, Singh R, Szerlip P A, Horsfall P and Goodman N D 2019 *J. Mach. Learn. Res.* **20** 28:1–28:6

[71] Anderson D, Tannehill J C, Pletcher R H, Munipalli R, and Shankar V 2020 *Computational Fluid Mechanics and Heat Transfer (4th ed.)* (CRC Press)

[72] Lykkegaard M B, Dodwell T J, Fox C, Mingas G and Scheichl R 2023 *SIAM/ASA Journal on Uncertainty Quantification* **11** 1–30

[73] Wang S, Wang H and Perdikaris P 2021 *Science advances* **7** eabi8605

[74] Riutort-Mayol G, Bürkner P C, Andersen M R, Solin A and Vehtari A 2023 *Statistics and Computing* **33** 17

[75] Botha I, Adams M P, Frazier D, Tran D K, Bennett F R and Drovandi C 2023 *Inverse Problems* **39** 125014

[76] Duffield S and Singh S S 2022 *Statistics & Probability Letters* **187** 109523

[77] Marzouk Y, Moselhy T, Parno M and Spantini A 2016 *arXiv preprint arXiv:1602.05023*