

A scalable optimal-transport based local particle filter

Matthew M. Graham^{*} and Alexandre H. Thiery^{}**

Department of Statistics and Applied Probability, National University of Singapore

Abstract. Filtering in spatially-extended dynamical systems is a challenging problem with significant practical applications such as numerical weather prediction. Particle filters allow asymptotically consistent inference but require infeasibly large ensemble sizes for accurate estimates in complex spatial models. Localisation approaches, which perform local state updates by exploiting low dependence between variables at distant points, have been suggested as a potential resolution to this issue. Naively applying the resampling step of the particle filter locally however produces implausible spatially discontinuous states. The ensemble transform particle filter replaces resampling with an optimal-transport map and can be localised by computing maps for every spatial mesh node. The resulting local ensemble transport particle filter is however computationally intensive for dense meshes. We propose a new optimal-transport based local particle filter which computes a fixed number of maps independent of the mesh resolution and interpolates these maps across space, reducing the computation required and allowing it to be ensured particles remain spatially smooth. We numerically illustrate that, at a reduced computational cost, we are able to achieve the same accuracy as the local ensemble transport particle filter, and retain its improved robustness to non-Gaussianity and ability to quantify uncertainty when compared to local ensemble Kalman filters.

MSC 2010 subject classifications: Primary 65C35; secondary 86A22.

Key words and phrases: particle filtering, Bayesian filtering, spatial models, inverse problems, localisation, optimal transport.

1. INTRODUCTION

A natural paradigm for modelling geophysical systems such as the atmosphere is as *spatially-extended dynamical systems*: one or more state variables defined over a spatial domain are evolved through time according to a set of *stochastic partial differential equations* (SPDEs). In this article we will consider the problem of inferring the distribution of the unknown state of such a system given noisy observations at a sequence of time points. As well as being an important problem in its own right, state inference is also a vital sub-component of tasks such as forecasting the future state of a system and inferring values for any free parameters in the numerical model used (Fearnhead and Künsch, 2018).

(e-mail: ^{*}m.m.graham@nus.edu.sg; ^{**}a.h.thiery@nus.edu.sg)

A key issue in performing state inference in spatially-extended systems is the typically high dimension of the state space. To allow numerical simulation of the SPDE model the spatial domain is discretised into a mesh (also known as a grid); the system state can then be represented as a finite-dimensional vector consisting of the concatenated values of the state variables at the nodes of the mesh. The resulting state dimension is therefore a multiple of the number of mesh nodes which can be very large. For example in the global atmospheric models used in current operational *numerical weather prediction* (NWP) systems the mesh size can be of the order 10^8 or higher (Bauer, Thorpe and Brunet, 2015).

For large state dimensions, even inference in linear-Gaussian models¹ using the *Kalman filter* (KF) (Kalman, 1960) is computationally infeasible due to the high processing and memory costs of operations involving the full covariance matrix of the state distribution. This motivated the development of *ensemble Kalman filter* (ENKF) methods (Evensen, 1994; Burgers, van Leeuwen and Evensen, 1998) which use an ensemble of particles to represent the state distribution rather than the full mean and covariance statistics. As the ensemble sizes used are typically much smaller than the state dimension² the computational savings can be considerable.

Although ENKF methods are only consistent in an infinite ensemble limit for linear-Gaussian models (Furrer and Bengtsson, 2007; Le Gland, Monbet and Tran, 2011), they have been empirically found to perform well in models with weakly non-linear state update and observation operators, even when using relatively small ensembles of size much less than the state dimension (Evensen, 2009); the performance of the ENKF in non-asymptotic regimes has been theoretically investigated in several recent works (Kelly, Law and Stuart, 2014; Del Moral and Tugaut, 2018; Bishop and Del Moral, 2018; Tong, Majda and Kelly, 2016). A key aspect in allowing ENKF methods to be scaled to large spatially-extended geophysical models is the use of *spatial localisation* (Houtekamer and Mitchell, 1998; Hamill, Whitaker and Snyder, 2001). Localisation exploits the observation that there is often low statistical dependence between state variables at distant points in spatially-extended systems. In ENKF methods this property is used to improve the noisy covariance estimates resulting from the small ensemble sizes used by removing spurious correlations between distant state variables.

ENKF methods have been successfully applied in a variety of settings, including operational NWP systems (Bonavita, Torrisi and Marcucci, 2008; Clayton, Lorenc and Barker, 2013), however the quality of the state distribution estimates is fundamentally limited by the linear-Gaussian assumptions made by the underlying KF updates. For models with non-Gaussian noise processes or strongly non-linear state update or observation operators, ENKF methods tend to produce poor estimates of the state distribution (Lei, Bickel and Snyder, 2010).

Particle filters (PFS) (Gordon, Salmond and Smith, 1993; Del Moral, 1996) offer an alternative ensemble-based approach to sequential state inference that unlike ENKF methods provides consistent estimates for non-Gaussian distributions. The simplest variant, the bootstrap PF, alternates propagating the ensemble members forward in time under the model dynamics, with resampling according to weights

¹Throughout this article we will for brevity refer to dynamical models with linear state update and observation operators and additive Gaussian noise processes as *linear-Gaussian*.

²Current operational NWP ensemble systems are limited to ~ 50 particles due to the high computational cost of numerically integrating the particles forward in time (Buizza et al., 2005).

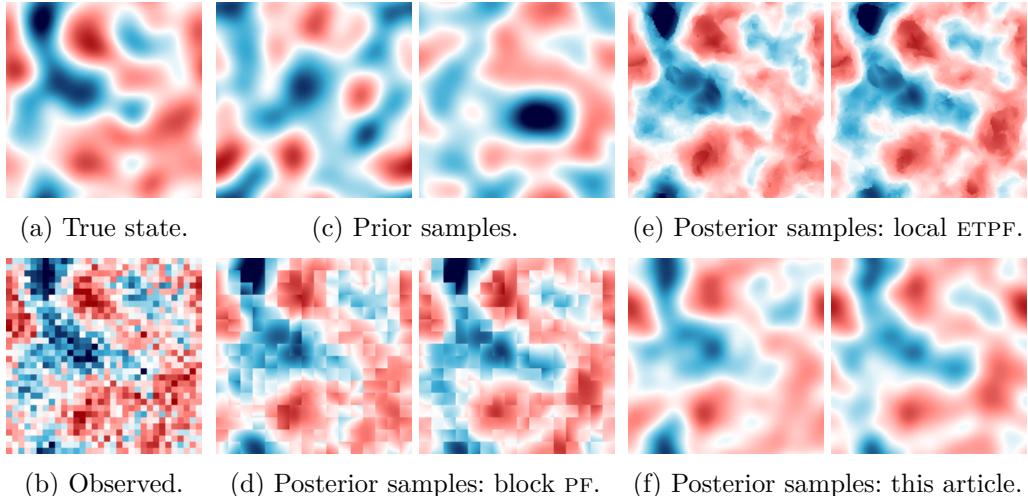


Fig 1: Examples of local PF assimilation updates applied to a Gaussian process model. The smooth true state field is shown in panel (a) and corresponding noisy observations in (b). Panel (c) shows prior samples and (d)–(f) approximate posterior samples after applying different local PF assimilation updates. In each of (c)–(f) 2 out of 40 samples are shown.

calculated from the likelihood of the particles given the observed data.

While PFs offer asymptotically consistent inference for general state space models, in practice they typically suffer from *weight-degeneracy* in high-dimensional systems: after propagation only a single particle has non-negligible weight. For even simple linear-Gaussian models, PFs have been shown to require an ensemble size which scales exponentially with the number of observations to avoid degeneracy (Snyder et al., 2008; Bengtsson, Bickel and Li, 2008; Snyder, 2011).

Given the importance of localisation in scaling ENKF methods to large spatial systems, it is natural to consider whether PF methods can be localised to overcome weight-degeneracy issues (Snyder et al., 2008; Van Leeuwen, 2009). Rebeschini and van Handel (2015) analysed a simple local PF scheme in which the spatial domain is partitioned into disjoint blocks and independent PFs run for each block, with local particle weights computed from the observations within each block. The authors demonstrate this *block* PF algorithm can overcome the need to exponentially scale the ensemble size with dimension to prevent degeneracy. However as the variables in each block are resampled independently from those in other blocks, dependencies between blocks are ignored; this introduces a systematic bias that is difficult to control (Bertoli and Bishop, 2014).

This issue is illustrated for a two-dimensional Gaussian process model in Fig. 1. The smooth true state field, shown in Fig. 1a, is partially and noisily observed (Fig. 1b). While the samples in the prior ensemble (Fig. 1c) reflect the smoothness of the true state field, the posterior samples shown in Fig. 1d, computed using a block PF assimilation update show spatial discontinuities at the block boundaries. Such discontinuities can cause numerical instabilities in the computation of spatial derivatives when integrating the SPDES model to forward propagate the particles.

The *ensemble transform particle filter* (ETPF) (Reich, 2013) uses an *optimal*

transport (OT) map to linearly transform an ensemble instead of resampling. The ETPF can be localised by computing OT maps for each mesh node using local particle weights (Cheng and Reich, 2015); updating the particles using the resulting spatially varying maps significantly reduces the introduction of spatial discontinuities compared to independent resampling. This can be seen in the samples computed using the local ETPF shown in Fig. 1e, which show greater spatial regularity than the block PF samples in Fig. 1d, though they remain less smooth than the true state field.

The requirement in the local ETPF to solve an OT problem at every node can be computationally burdensome when the mesh size is large. Solving each OT problem has complexity $\tilde{\mathcal{O}}(P^3)$ where P is the ensemble size ($\tilde{\mathcal{O}}$ indicates limiting complexity excluding polylogarithmic factors); although solvers can be run in parallel this still represents a large computational overhead.

In this article we propose an alternative smooth and computationally scalable local ETPF scheme. A finite set of *patches* which cover the spatial domain are defined, with a non-negative *bump function* supported on the patch. The set of bump functions is constrained to be a *partition of unity* (POU): the functions sum to unity at all points in the spatial domain. A single OT map is calculated for each spatial patch. The POU is then used to interpolate these local per-patch maps across the spatial domain, defining maps for all nodes in the spatial mesh.

Through an appropriate choice of bump functions this scheme can maintain a prescribed level of smoothness in the transformed state fields while also significantly reducing the number of OT problems needing to be solved. Examples posterior samples computed using the proposed scheme are shown in Fig. 1f. Here the POU is a set of smooth bump functions tiled in a 8×8 grid. As well as giving more plausibly smooth fields than those computed using the local ETPF, in this example the number of OT problems solved was reduced from 16 384 to 64.

The remainder of the article is structured as follows. In Section 2 we briefly introduce our notation and some preliminaries on the filtering problem and ensemble methods, followed by a review of SPDE models and existing local filtering approaches in Section 3. The new method we propose is described in Section 4 and a numerical study comparing the approach to existing local ensemble filters is presented in Section 5, with a concluding discussion in Section 6.

2. ENSEMBLE APPROACHES TO FILTERING

2.1 Notation

Random variables are denoted by sans-serif symbols, e.g. x , and $x \sim \mu$ indicates x has distribution μ . The probability of an event x taking a value in a set \mathcal{A} is $\mathbb{P}(x \in \mathcal{A})$ and the expected value of x is $\mathbb{E}[x]$. The conditional probability of $x \in \mathcal{A}$ given $y = y$ is denoted $\mathbb{P}(x \in \mathcal{A} | y = y)$ and likewise the conditional expectation of x given $y = y$ is $\mathbb{E}[x | y = y]$. A Gaussian distribution with mean m and covariance C is denoted $\mathcal{N}(m, C)$. The set of integers from A to B inclusive is $A:B$ and quantities sub- or superscripted by an integer range indicate an indexed set, e.g. $\phi_{1:M} = \{\phi_m\}_{m \in 1:M}$. The D vector of ones is 1_D and the $D \times D$ identity matrix I_D , with the subscript omitted when unambiguous. The indicator function on a set \mathcal{S} is $\mathbb{1}_{\mathcal{S}}$. The set of real numbers is \mathbb{R} , non-negative reals $\mathbb{R}_{\geq 0}$ and complex numbers \mathbb{C} . For $z \in \mathbb{C}$, $\Re(z)$ and $\Im(z)$ indicate its real and imaginary parts.

2.2 State-space models

The class of models we aim to perform inference in is *state-space models* (SSMs). Let \mathcal{X} be a vector-space representing the *state-space* of the system of interest. We assume observations of the system are available at a set of T times, with the observations at each discrete *time index* $t \in 1:T$ belonging to a common vector-space \mathcal{Y} . We denote the unknown system state at each time index as a random variable $x_t \in \mathcal{X}$ and the corresponding observations as a random variable $y_t \in \mathcal{Y}$. The modelled state dynamics are assumed to be Markovian and specified by a set of *state-update operators* $F_{1:T}$ such that

$$x_1 = F_1(u_1), \quad u_1 \sim \mu_1; \quad x_t = F_t(x_{t-1}, u_t), \quad u_t \sim \mu_t \quad \forall t \in 2:T, \quad (2.1)$$

with each $u_t \in \mathcal{U}$ a *state noise* variable drawn from a distribution μ_t , representing the stochasticity in the state initialisation and dynamics at each time step. The observations y_t at each time index t are assumed to depend only on the current state x_t and are generated via a set of *observation operators* $G_{1:T}$,

$$y_t = G_t(x_t, v_t), \quad v_t \sim \nu_t \quad \forall t \in 1:T. \quad (2.2)$$

Any stochasticity in the observation process at each time index is introduced by the *observation noise* variable $v_t \in \mathcal{V}$ with distribution ν_t . In SSMs where the operators $F_{1:T}$ and $G_{1:T}$ are all linear and the distributions $\mu_{1:T}$ and $\nu_{1:T}$ are all Gaussian – the aforementioned linear-Gaussian case – the joint distribution on all states $x_{1:T}$ and observations $y_{1:T}$ is Gaussian and a KF can be used to perform exact inference. In this article we will focus on approximate inference methods for SSMs outside this class where exact inference is intractable.

We require that the conditional distributions on y_t given x_t have known densities $g_{1:T}$ with respect to a common dominating measure v on \mathcal{Y} , i.e.

$$\mathbb{P}(y_t \in dy | x_t = x_t) = g_t(y | x_t) v(dy) \quad \forall t \in 1:T. \quad (2.3)$$

For the state updates we assume only that the state-update operators F_t can be computed for any set of inputs and that we can generate samples from the state noise distributions μ_t ; the resulting state transition distributions will not necessarily have tractable densities.

2.3 Filtering and predictive distributions

Our main objects of interest from an inference perspective are the *filtering distributions*: the conditional distributions on the state at time index $t \in 1:T$ given the observations at time indices up to and including t . We will denote the filtering distribution at each time index t as

$$\pi_t(dx) = \mathbb{P}(x_t \in dx | y_{1:T} = y_{1:T}). \quad (2.4)$$

The *filtering problem* is then the task of inferring the filtering distributions $\pi_{1:T}$ given a SSM for the system and a sequence of observations $y_{1:T}$.

A further concept that will be important for our discussion of inference methods is the *predictive distribution* on the state at the next time index $t + 1$ given the observations up to the current time index t . We will denote the predictive distribution at time index t as

$$\vec{\pi}_{t+1}(dx) = \mathbb{P}(x_{t+1} \in dx | y_{1:T} = y_{1:T}). \quad (2.5)$$

2.4 Prediction and assimilation updates

A key property for filtering algorithms is that the filtering distribution at any time index can be expressed recursively in terms of the distributions at the previous time indices. Generally this recursion is split into two steps, here termed the *prediction* and *assimilation* updates.

The prediction update transforms the filtering distribution π_t to the predictive distribution $\vec{\pi}_{t+1}$. This update corresponds to propagating the state distribution forward in time according to the modelled dynamics, with no new observations introduced. Denoting the Dirac measure at a point $x \in \mathcal{X}$ by δ_x the prediction update can be expressed as

$$\vec{\pi}_{t+1}(dx) = \int_{\mathcal{U}} \int_{\mathcal{X}} \delta_{F_{t+1}(x', u)}(dx) \pi_t(dx') \mu_{t+1}(du). \quad (2.6)$$

The assimilation update then relates the predictive distribution $\vec{\pi}_{t+1}$ to the filtering distribution at the next time step π_{t+1} . It corresponds to an application of Bayes' theorem, with the predictive distribution forming the prior and the filtering distribution at the next time index the posterior after a new observed data point has been assimilated. The observation density g_{t+1} defines the likelihood term, with the assimilation update then

$$\pi_{t+1}(dx) = \frac{g_{t+1}(y_{t+1} | x)}{\int_{\mathcal{X}} g_{t+1}(y_{t+1} | x') \vec{\pi}_{t+1}(dx')} \vec{\pi}_{t+1}(dx). \quad (2.7)$$

The combination of prediction and assimilation updates together define a map from the filtering distribution at time index t to the distribution at $t + 1$:

$$\dots \longrightarrow \pi_t \xrightarrow{\text{prediction}} \vec{\pi}_{t+1} \xrightarrow{\text{assimilation}} \pi_{t+1} \longrightarrow \dots;$$

sequentially alternating prediction and assimilation updates is in theory therefore all that is needed to compute the filtering distributions at all times indices. In practice however for most SSMS the integrals in Eqs. (2.6) and (2.7) will be intractable to solve exactly, necessitating some form of approximation.

2.5 Ensemble filtering

A particularly common approximation is to use an ensemble of state particles to represent the filtering distribution at each time index. Specifically the filtering distribution π_t at time index t is represented by an empirical measure defined by placing point masses at the values of a set of P state particles $x_t^{1:P}$

$$\pi_t(dx) \approx \frac{1}{P} \sum_{p \in 1:P} \delta_{x_t^p}(dx). \quad (2.8)$$

A key advantage of using an ensemble representation of the filtering distribution is that a simple algorithm can be used to implement a prediction update consistent with Eq. (2.6). Specifically if a set of P independent state noise samples $u_{t+1}^{1:P}$ are generated from μ_{t+1} , then given particles $x_t^{1:P}$ approximating π_t , a new set of P particles can be computed as

$$\vec{x}_{t+1}^p = F_{t+1}(x_t^p, u_{t+1}^p) \quad \forall p \in 1:P. \quad (2.9)$$

This new particle ensemble can then be used to form an empirical measure approximation to the predictive distribution $\vec{\pi}_{t+1}$

$$\vec{\pi}_{t+1}(dx) \approx \frac{1}{P} \sum_{p \in 1:P} \delta_{\vec{x}_{t+1}^p}(dx). \quad (2.10)$$

2.6 Linear ensemble transform filters

Although Eq. (2.9) specifies an approach for performing a prediction update, a method for approximating the assimilation update in Eq. (2.7) to account for the observed data is also required. One possibility is to require that the filtering ensemble $\mathbf{x}_t^{1:\mathbb{P}}$ is formed as a linear combination of the predictive ensemble $\vec{\mathbf{x}}_t^{1:\mathbb{P}}$

$$\mathbf{x}_t^p = \sum_{q \in 1:\mathbb{P}} a_t^{p,q} \vec{\mathbf{x}}_t^q \quad (2.11)$$

where $a_t^{1:\mathbb{P},1:\mathbb{P}} \in \mathbb{R}^{\mathbb{P} \times \mathbb{P}}$ are a set of coefficients describing the transformation. In general the coefficients may depend non-linearly on both the observation y_t and predictive ensemble particles $\vec{\mathbf{x}}_t^{1:\mathbb{P}}$, however the form of the update constrains the filtering ensemble $\mathbf{x}_t^{1:\mathbb{P}}$ to lie in the linear subspace spanned by the predictive ensemble members. The class of ensemble filters using an assimilation update of the form in Eq. (2.11) was termed *linear ensemble transform filters* (LETFS) in Cheng and Reich (2015), and encompasses both ensemble Kalman and particle filtering methods, as will be discussed in the following subsections.

2.7 Ensemble Kalman filters

In a linear-Gaussian SSM the predictive and filtering distributions are Gaussian at all time indices: $\pi_t = \mathcal{N}(m_t, C_t)$ and $\vec{\pi}_t = \mathcal{N}(\vec{m}_t, \vec{C}_t)$ for all $t \in 1:T$, and so can be fully described by the mean and covariance parameters. The *Kalman filter* (KF) (Kalman, 1960) gives an efficient scheme for performing exact inference in linear-Gaussian SSMs by iteratively updating the mean and covariance parameters. For an observation operator and noise distribution

$$G_t(x, v) = H_t x + v, \quad v_t \sim \mathcal{N}(0, R_t), \quad (2.12)$$

the KF assimilation update can be written

$$C_t = \vec{C}_t - \vec{C}_t H_t^\top (R_t + H_t \vec{C}_t H_t^\top)^{-1} H_t \vec{C}_t, \quad (2.13a)$$

$$m_t = \vec{m}_t + C_t H_t^\top R_t^{-1} (y_t - H_t \vec{m}_t). \quad (2.13b)$$

Ensemble Kalman filter (ENKF) methods are a class of LETFS which use an assimilation update consistent with the KF updates in Eq. (2.13) for linear-Gaussian SSMs in the limit of an infinite ensemble, in effect replacing the predictive mean \vec{m}_t and covariance \vec{C}_t with ensemble estimates. The use of an ensemble representation rather than the full means and covariances used in the KF both gives a significant computational gain (by avoiding the need to store and perform operations on the full covariance matrices) while also allowing application of the approach to SSMs with non-linear state updates via the prediction update in Eq. (2.9).

The originally proposed ENKF method (Evensen, 1994; Burgers, van Leeuwen and Evensen, 1998) generates simulated observations from the observation model in Eq. (2.12) for each predictive ensemble member to form a Monte Carlo estimate of the $R_t + H_t \vec{C}_t H_t^\top$ term in Eq. (2.13a). Although simple to implement, the introduction of artificial observation noise adds an additional source of variance which can be significant for small ensemble sizes. This additional variance can be eliminated by the use of *square-root* ENKF variants (Anderson, 2001; Bishop, Etherton and Majumdar, 2001; Whitaker and Hamill, 2002) which typically giving more stable and accurate filtering for small ensemble sizes.

Of particular interest here is the *ensemble transform Kalman filter* (ETKF) proposed by [Bishop, Etherton and Majumdar \(2001\)](#), with this approach particularly efficient in the regime of interest where the ensemble size P is much smaller than the state and observation dimensionalities. As we will use a localised variant of the ETKF as a baseline in the numerical experiments in Section 5 we outline the ETKF algorithm in Appendix A and show how it can be expressed in the form of the LETF assimilation update in Eq. (2.11).

2.8 Particle filters

Particle filtering offers an alternative LETF approach that gives consistent estimates of the filtering distributions as $P \rightarrow \infty$ for the non-Gaussian case. The PF assimilation update transforms the empirical approximation to the predictive distribution $\tilde{\pi}_t$ in Eq. (2.10) to an empirical approximation of the filtering distribution π_t by attaching importance weights to the predictive ensemble

$$\tilde{w}_t^p = g_t(y_t | \vec{x}_t^p), \quad w_t^p = \frac{\tilde{w}_t^p}{\sum_{q \in 1:P} \tilde{w}_t^q} \quad \forall p \in 1:P, \quad \pi_t(dx) \approx \sum_{p \in 1:P} w_t^p \delta_{\vec{x}_t^p}(dx). \quad (2.14)$$

Directly iterating this importance weighting scheme, at each time index propagating the ensemble forward in time according to Eq. (2.9) and incrementally updating a set of (unnormalised) importance weights gives an algorithm termed *sequential importance sampling*. While appealingly simple, sequential importance sampling requires an exponentially growing ensemble size as the number of observation times T increases. The key additional step in particle filtering is to resample the particle ensemble according to the importance weights between prediction updates. That is the filtering distribution ensemble at time index t is defined in terms of the corresponding predictive distribution ensemble as

$$\vec{x}_t^p = \sum_{q \in 1:P} r_t^{p,q} \vec{x}_t^q \quad \forall p \in 1:P, \quad (2.15)$$

where $r_t^{1:P,1:P} \in \{0, 1\}^{P \times P}$ are a set of binary random variables satisfying

$$\sum_{q \in 1:P} r_t^{p,q} = 1, \quad \mathbb{E} \left[\sum_{q \in 1:P} r_t^{q,p} | w_t^p = w_t^p \right] = P w_t^p \quad \forall p \in 1:P. \quad (2.16)$$

This has the effect of removing particles with low weights from the ensemble and so ensures computational effort is concentrated on the most plausible particles. There are multiple algorithms available for generating random variables $r_t^{1:P,1:P}$ satisfying Eq. (2.16) - see for example the reviews in ([Douc and Cappé, 2005](#); [Hol, Schon and Gustafsson, 2006](#); [Gerber, Chopin and Whiteley, 2019](#)). Distributed versions of particle filters have recently been proposed and analyzed ([Bolic, Djuric and Hong, 2005](#); [Vergé et al., 2015](#); [Whiteley, Lee and Heine, 2016](#); [Sen and Thiery, 2019](#); [Lee and Whiteley, 2015](#)).

The iterated application of prediction updates according to Eq. (2.9) and resampling assimilation updates according to Eq. (2.15) together defines the *bootstrap PF* algorithm. Although simple, the bootstrap PF algorithm does not exploit all the information available at each time index – specifically the prediction update in Eq. (2.9) does not take in to account future observations. Alternative PF schemes can be employed which use prediction updates which take in to account

future observations. Although such schemes typically express the resulting particle weights in terms of the state transition densities we describe in Appendix B how they can be implemented in SSMS with intractable transition densities.

While adjusting the prediction update can significantly improve performance compared to the bootstrap PF for a fixed ensemble size, when applied to systems with high state and observation dimensionalities these PF methods will still tend to suffer from weight degeneracy. In particular, even when using ‘locally optimal’ updates in a simple linear-Gaussian model, the resulting PF has been shown to still generally require an ensemble size which still grows exponentially with the dimension of the observation space to avoid weight degeneracy (Snyder et al., 2008; Snyder, Bengtsson and Morzfeld, 2015).

2.9 Ensemble transform particle filters

Although typically the resampling variables in PF assimilation updates are generated independently of the predictive ensemble particle values given the weights, this is not required. Reich (2013) exploited this flexibility to propose an alternative particle filtering approach termed the *ensemble transform particle filter* (ETPF) which uses OT methods to compute a resampling scheme which minimises the expected distances between the particles before and after resampling.

A valid resampling scheme can be parametrised by a set of resampling probabilities $\rho_t^{1:P,1:P} \in [0, 1]^{P \times P}$ with $\rho_t^{p,q} = \mathbb{P}(r_t^{p,q} = 1 | w_t^q = w_t^q)$ satisfying

$$\sum_{q \in 1:P} \rho_t^{p,q} = 1, \quad \sum_{q \in 1:P} \rho_t^{q,p} = P w_t^p \quad \forall p \in 1:P. \quad (2.17)$$

A simple choice satisfying Eq. (2.17) is $\rho_t^{p,q} = w_t^q / \sum_{q \in 1:P} w_t^q \quad \forall p \in 1:P, q \in 1:P$ with this corresponding to the probabilities used in standard PF resampling schemes.

If we denote the set of resampling probabilities satisfying Eq. (2.17) for a given set of weights $w_t^{1:P}$ by $\mathcal{R}(w_t^{1:P})$ and the realisations of the predictive particles $\vec{x}_t^{1:P}$ at time index t by $\vec{x}_t^{1:P}$, Reich (2013) instead proposed to compute the resampling probabilities as the solution to the optimal transport problem

$$\rho_t^{1:P,1:P} = \operatorname{argmin}_{\varrho^{1:P,1:P} \in \mathcal{R}(w_t^{1:P})} \sum_{p \in 1:P} \sum_{q \in 1:P} \varrho^{p,q} |\vec{x}_t^p - \vec{x}_t^q|_2^2. \quad (2.18)$$

The optimal transport problem can be posed as a linear program and efficiently solved using the network simplex algorithm (Orlin, 1997) with a computational complexity of order $\tilde{\mathcal{O}}(P^3)$. While the resulting resampling probabilities could then be used to generate binary variables $r_t^{1:P,1:P}$ and the standard PF resampling assimilation update in Eq. (2.15) applied, Reich (2013) instead proposes to use the resampling probabilities to directly update the particles as follows

$$x_t^p = \sum_{q \in 1:P} \rho_t^{p,q} \vec{x}_t^q \quad \forall p \in 1:P. \quad (2.19)$$

For $P \rightarrow \infty$ this assimilation update remains consistent as, due to properties of the optimal transport problem solution, the resampling probabilities tend to binary $\{0, 1\}$ values (Reich, 2013, Theorem 1) and thus Eq. (2.19) becomes equivalent to updating using realisations of the binary random variables.

While the ETPF does not in itself help overcome the weight degeneracy issue, the deterministic and distance minimising nature of the ETPF update naturally lends itself to spatial localisation approaches which can help overcome the poor scaling of PFs with dimensionality, as will be discussed in the following section.

3. SPATIAL MODELS AND LOCAL ENSEMBLE FILTERS

Our particular focus in this article is on filtering in models of spatially-extended dynamical systems. Let \mathcal{S} be a D -dimensional compact metric space equipped with distance function $d : \mathcal{S} \times \mathcal{S} \rightarrow [0, \infty)$, representing the spatial domain the state of the modelled system is defined over, and $\mathcal{Z} \subseteq \mathbb{R}^N$ be the space the state variables at each spatial coordinate in \mathcal{S} take values in. The state-space of the system is then a function space $\mathcal{Z}^{\mathcal{S}}$ with the state at each time index $z_t : \mathcal{S} \rightarrow \mathcal{Z}$ a spatial field. The dynamics of the system will typically be modelled by a set of SPDES, with $z_{1:T}$ then corresponding to a solution of these equations at T times, given an initial state sampled from some distribution.

In practice in most problems we cannot solve the SPDE model exactly and instead use numerical integration schemes to generate approximate solutions. The states are assumed to be restricted to a function space with a fixed dimensional representation, with typically a state field $z_t : \mathcal{S} \rightarrow \mathcal{Z}$ represented as a linear combination of a finite set of M basis functions $\beta_m : \mathcal{S} \rightarrow \mathbb{R}$

$$z_t(s) = \sum_{m \in 1:M} x_{t,m} \beta_m(s) \quad \forall t \in 1:T, s \in \mathcal{S}, \quad (3.1)$$

with coefficients $x_{t,m} \in \mathcal{Z} \quad \forall m \in 1:M$. For the purposes of inference we will therefore consider the state space to be a vector space $\mathcal{X} = \mathcal{Z}^M \subseteq \mathbb{R}^{MN}$ with state vectors consisting of the concatenation of the basis function coefficients.

Typically the basis functions will be defined by partitioning the spatial domain \mathcal{S} in to a *mesh* of polytopic spatial elements, for example triangles or quadrilaterals for $D = 2$. The vertices of these polytopes (and potentially additional points such as the midpoints of edges) define a collection of M *nodes* with spatial locations $s_{1:M}$. Typically each node is associated with a basis function β_m satisfying

$$\beta_m(s_m) = 1, \quad \beta_m(s_n) = 0 \quad \forall m \in 1:M, n \in 1:M, n \neq m, \quad (3.2)$$

which combined with Eq. (3.1) implies that $z_t(s_m) = x_{t,m} \quad \forall m \in 1:M$.

We will assume that there are L observations $y_{t,1:L}$ at every time point, each of dimension K , with the overall observation vector y_t then a length KL vector

$$y_t^T = [y_{t,1}^T \ y_{t,2}^T \ \cdots \ y_{t,L}^T] \quad \forall t \in 1:T. \quad (3.3)$$

We also assume that $y_{t,l} \perp y_{t,m} | x_t \quad \forall l \neq m$ i.e. the observations are conditionally independent given the state and that each observation $y_{t,l}$ depends only on the value of the state field z_t at a fixed spatial location s_l^o . Together these two assumptions mean we can express the logarithm of the observation density as

$$\log g_t(y_t | x_t) = \sum_{l \in 1:L} \log g_{t,l}(y_{t,l} | z_t(s_l^o)) \quad \forall t \in 1:T. \quad (3.4)$$

3.1 Decay of spatial correlations

The combination of high state and observation space dimensionalities, and low feasible ensemble sizes, make filtering in spatial SSMS a significant computational challenge. Fortunately SSMS of spatially extended systems often also exhibit a favourable *decay of spatial correlations* property which can be exploited to make approximate filtering more tractable by performing local updates to the particles.

If we assume the spatial field z_t is defined as in Eq. (3.1) and x_t is distributed according to the filtering distribution π_t then the spatial correlation function $c_{t,f} : \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$ of a square integrable function $f \in \mathcal{L}^2$ is defined as

$$c_{t,f}(s, s') = \frac{\mathbb{E}[f(z_t(s))f(z_t(s'))] - \mathbb{E}[f(z_t(s))]\mathbb{E}[f(z_t(s'))]}{(\mathbb{E}[(f(z_t(s)) - \mathbb{E}[f(z_t(s))])^2]\mathbb{E}[(f(z_t(s')) - \mathbb{E}[f(z_t(s'))])^2])^{\frac{1}{2}}}, \quad (3.5)$$

and the *maximal spatial correlation function* as $\bar{c}_t(s, s') = \sup_{f \in \mathcal{L}^2} c_{t,f}(s, s')$.

The decay of spatial correlations property can then be stated as

$$\bar{c}_t(s, s') \rightarrow 0 \quad \text{as} \quad d(s, s') \rightarrow \infty \quad \forall s \in \mathcal{S}, s' \in \mathcal{S}, \quad (3.6)$$

which indicates that the dependence between state variables at distinct spatial locations decays to zero as the distance between the locations increases.

While it will typically not be possible to analytically verify Eq. (3.6) holds exactly, it has been empirically observed that models of spatially extended systems in which the underlying dynamics are governed by local interactions between the state variables exhibit an approximate decay of correlations property. In particular weak long-range spatial correlations are a defining feature of *spatio-temporal chaos* (Hunt, Kostelich and Szunyogh, 2007) with many spatial models of interest, such as the atmospheric models used in NWP, exhibiting such behaviour.

3.2 Local linear ensemble transform filters

For SSMS exhibiting a decay of spatial correlations property, localising the LETF assimilation update in Eq. (2.11), as proposed by Cheng and Reich (2015), can offer significant performance gains compared to algorithms employing global updates. Rather than using a single set of transform coefficients $a_t^{1:P,1:P}$ for the assimilation update, M sets of coefficients $a_{t,1:M}^{1:P,1:P}$ are defined, one for each spatial mesh node location $s_{1:M}$ with the assimilation update then

$$x_{t,m}^p = \sum_{q \in 1:P} a_{t,m}^{p,q} \vec{x}_{t,m}^q \quad \forall p \in 1:P, \forall m \in 1:M. \quad (3.7)$$

As previously mentioned, the global LETF update in Eq. (2.11) restricts the filtering ensemble members $x_t^{1:P}$ to lie in the P dimensional linear subspace of \mathcal{X} spanned by the predictive ensemble $\vec{x}_t^{1:P}$. When \mathcal{X} is high-dimensional, as is generally the case in spatially extended models, this can be highly restrictive.

The local LETF update in Eq. (3.7) overcomes this restriction of the global LETF update, with the filtering ensemble members now formed from local linear combinations of the predictive ensemble members and thus no longer constrained to a P dimensional linear subspace. In particular for models exhibiting a decay of correlations property, the state variables at each mesh node can be updated using coefficients computed using only the subset of observations which are within some localisation radius of the mesh node while still retaining accuracy.

Local variants of the ENKF (Houtekamer and Mitchell, 1998; Hamill, Whitaker and Snyder, 2001) are the prototypical examples of local LETFs, and have been successfully used to perform filtering in large complex spatio-temporal models including operational ensemble NWP systems (Bowler et al., 2009). In Appendix A we briefly introduce a local variant of the ETKF (Hunt, Kostelich and Szunyogh, 2007) which we use as a baseline in the numerical experiments.

3.3 Local particle filters

It has been speculated that spatial localisation may be key to achieving useful results from PFS in large spatio-temporal models ([Morfeld, Hodges and Snyder, 2017](#)) based on its importance to the success of ENKF methods in such models. In [Farchi and Bocquet \(2018\)](#) the authors systematically compare a wide range of localised PF and related algorithms which have been proposed in the literature including localised variants of the ETPF which we will discuss in the following subsection. Below we briefly introduce concepts from a local PF algorithm proposed by [Penny and Miyoshi \(2015\)](#) which are relevant to this article, however we refer readers to [Farchi and Bocquet \(2018\)](#) for a much more extensive review.

For the standard PF, the logarithms of the unnormalised particle weights are

$$\log \tilde{w}_t^p = \sum_{l \in 1:L} \log g_{t,l}(y_{t,l} | \vec{z}_t^p(s_l^o)) \quad \forall p \in 1:P. \quad (3.8)$$

i.e. a summation of contributions due to the observations at all locations $s_{1:L}^o$.

For a model exhibiting a decay of spatial correlations property we would expect that only a local subset of observations should have a strong influence on the distribution of the state variables at each mesh node. We can formalise this intuition into a concrete approach for computing local particle weights via the use of a *localisation function* $\ell_r : [0, \infty) \rightarrow [0, 1]$ and localisation radius r satisfying

$$\ell_r(0) = 1, \quad \ell_r(d) = 0 \quad \forall d > r > 0. \quad (3.9)$$

Local unnormalised weights for each mesh node can then be defined

$$\log \tilde{w}_{t,m}^p = \sum_{l \in 1:L} \log g_{t,l}(y_{t,l} | \vec{z}_t^p(s_l^o)) \ell_r(d(s_m, s_l^o)) \quad \forall p \in 1:P, m \in 1:M, \quad (3.10)$$

and corresponding local normalised weights

$$w_{t,m}^p = \frac{\tilde{w}_{t,m}^p}{\sum_{q \in 1:P} \tilde{w}_{t,m}^q} \quad \forall p \in 1:P, m \in 1:M. \quad (3.11)$$

This formulation for the local particle weights has the desired property of using only a local subset of observations to update the state variables at each mesh node (with the terms in the sum zero when $d(s_m, s_l^o) > r$).

Typical choices for the localisation function include the *uniform* or top-hat function $\ell_r(d) = \mathbb{1}_{[0,r]}(d)$ and the *triangular* function $\ell_r(d) = (1 - \frac{d}{r})\mathbb{1}_{[0,r]}(d)$. In this article we exclusively use the smooth and compactly supported 5th order piecewise rational function proposed by [Gaspari and Cohn \(1999\)](#) and defined as

$$\ell_r(d) = \begin{cases} -8\frac{d^5}{r^5} + 8\frac{d^4}{r^4} + 5\frac{d^3}{r^3} - \frac{20}{3}\frac{d^2}{r^2} + 1 & 0 \leq d < \frac{r}{2} \\ \frac{8}{3}\frac{d^5}{r^5} - 8\frac{d^4}{r^4} + 5\frac{d^3}{r^3} + \frac{20}{3}\frac{d^2}{r^2} - 10\frac{d}{r} + 4 - \frac{1}{3}\frac{r}{d} & \frac{r}{2} \leq d < r \end{cases}. \quad (3.12)$$

[Penny and Miyoshi \(2015\)](#) propose a local PF algorithm which uses local particle weights defined as in Eq. (3.11) for the specific case of a Gaussian observation density and uniform localisation function $\ell_r(d) = \mathbb{1}_{[0,r]}(d)$. The local weights are used to generate binary resampling variables $r_{t,1:M}^{1:P,1:P}$ for each mesh node satisfying

$$\sum_{q \in 1:P} r_{t,m}^{p,q} = 1, \quad \mathbb{E} \left[\sum_{q \in 1:P} r_{t,m}^{q,p} | w_{t,m}^p = w_{t,m}^p \right] = \mathbb{P} w_{t,m}^p \quad \forall p \in 1:P, m \in 1:M. \quad (3.13)$$

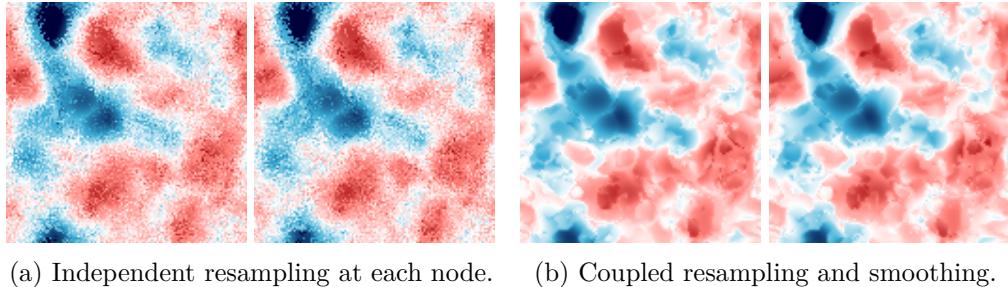


Fig 2: Examples of local PF assimilation updates applied to the same spatial Gaussian process model as Figure 1.

Generating the resampling variables for each mesh node independently means the state variables at adjacent mesh nodes for a post-resampling particle will typically originate from different prior particles, tending to lead to highly discontinuous and noisy spatial fields. An example of this is shown in Fig. 2a which show examples of the posterior state field samples generated using independent resampling at each mesh node with local weights for the smooth spatial Gaussian process example encountered previously in Figure 1.

To ameliorate the issues associated within using independent resampling variables, it is proposed in Penny and Miyoshi (2015) to use a variant of the *systematic resampling* scheme (Douc and Cappé, 2005) often used as variance reduction method in standard PF algorithms. A single random standard uniform variable is used to generate the resample variables for all mesh nodes, resulting in per-node sets of resampling variables which each satisfy the marginal requirements in Eq. (3.13) while also being strongly correlated to the resampling variables for other nodes. The correlation introduced between the resampling variables when using this ‘coupled resampling’ scheme significantly reduces but does not eliminate the introduction of discontinuities into the resampled fields.

Rather than directly use these resampling variables in a local equivalent to the PF assimilation update in Eq. (2.15), Penny and Miyoshi (2015) instead propose to use a ‘smoothed’ update which uses a weighted average of the resampling variables at the current mesh node and all neighbouring nodes to update the particles values at each node. Fig. 2b shows examples of posterior state fields samples computed using this smoothed assimilation update with the resampling variables generated using the coupled scheme. The previously observed discontinuities are now removed, however the samples still remain significantly less smooth than the true state used to generate the observations (Fig. 1a) and prior samples (Fig. 1c).

3.4 Local ensemble transform particle filter

While techniques such as the smoothed and coupled resampling update used in Penny and Miyoshi (2015) can help reduce the introduction of spatial discontinuities, the resampling variables $r_{t,1:M}^{1:P,1:P}$ are still calculated without taking into account the values of the predictive particles $\vec{x}_t^{1:P}$ values other than via the local particle weights. The ETPF assimilation update discussed in Section 2.9 explicitly tries to minimise a distance between the values of the transformed and pre-update particles and does not require introducing any randomness and so is a natural candidate for a local PFs with improved spatial smoothness properties.

[Cheng and Reich \(2015\)](#) proposed a localised variant of the ETPF as a particular instance of their LETF framework. Local particle weights are calculated as in Eqs. (3.10) and (3.11) for each mesh node, and a set of OT problems solved

$$\rho_{t,m}^{1:P,1:P} = \operatorname{argmin}_{\varrho^{1:P,1:P} \in \mathcal{R}(w_{t,m}^{1:P})} \sum_{p \in 1:P} \sum_{q \in 1:P} \varrho^{p,q} c_{t,m}^{p,q} \quad \forall m \in 1:M. \quad (3.14)$$

Here the transport cost terms $c_{t,1:M}^{1:P,1:P}$ are analogous to the inter-particle Euclidean distances used in Eq. (2.18). Rather than compute global transport costs based on distances between the state variables values at points across the full spatial domain, [Cheng and Reich \(2015\)](#) proposed to compute localised transports costs for each mesh node index $m \in 1:M$ by integrating a distance between the state variables values against a localisation function centred at the mesh node location s_m and with support on points $s \in \mathcal{S} : d(s, s_m) < r'$

$$c_{t,m}^{p,q} = \int_{\mathcal{S}} |\vec{z}_t^p(s) - \vec{z}_t^q(s)|_2^2 \ell'_{r'}(d(s_m, s)) ds \quad \forall m \in 1:M, p \in 1:P, q \in 1:P. \quad (3.15)$$

The localisation function $\ell'_{r'}$ and localisation radius r' are denoted with primes here to emphasise they may be different from those used for the local weights computation. A more pragmatic definition of the localised transport costs is

$$c_{t,m}^{p,q} = \sum_{n \in 1:M} \left| \vec{x}_{t,n}^p - \vec{x}_{t,n}^q \right|_2^2 \ell'_{r'}(d(s_m, s_n)) \quad \forall m \in 1:M, p \in 1:P, q \in 1:P. \quad (3.16)$$

In the common case of a rectilinear mesh with equal spacing between the nodes across the domain, the summation in Eq. (3.16) can be seen, as a quadrature approximation to the integral in Eq. (3.15) up to a constant multiplier which does not affect the OT solutions.

If the localisation functions ℓ_r and $\ell'_{r'}$ are smooth, then both the local weights $w_{t,1:M}^{1:P}$ and local transport costs $c_{t,1:M}^{1:P,1:P}$ will vary smoothly as functions of the mesh node locations $s_{1:M}$. However, the solutions $\rho_{t,1:M}^{1:P,1:P}$ to the linear programs defined by the local optimal transport problems in Eq. (3.14) will not vary smoothly with the mesh node locations $s_{1:M}$ even if the local weights and transport costs do. This can be seen in the spatial Gaussian process example in Fig. 1, with the local ETPF scheme used to compute the posterior samples illustrated in Fig. 1e. Although less apparent than the discontinuities in Fig. 1d, the fields in Fig. 1e still show spatial artefacts due to the non-smooth variation of the OT solutions.

One option to increase the smoothness of the update is to regularise the OT problems. In particular the *entropically regularised* OT problems defined by

$$\rho_{t,m}^{1:P,1:P} = \operatorname{argmin}_{\varrho_{t,m}^{1:P,1:P} \in \mathcal{R}(w_{t,m}^{1:P})} \sum_{p \in 1:P} \sum_{q \in 1:P} \left(\varrho_{t,m}^{p,q} c_{t,m}^{p,q} + \lambda \varrho_{t,m}^{p,q} (\log \varrho_{t,m}^{p,q} - 1) \right), \quad (3.17)$$

for some positive regularisation coefficient λ have a unique optimal solution which smoothly varies as a function of the local weights $w_{t,m}^{1:P}$ and transport costs $c_{t,1:M}^{1:P,1:P}$ ([Peyré and Cuturi, 2019](#)) and tends to the solution of the non-regularised problem with the highest entropy as $\lambda \rightarrow 0$. Further the entropically regularised problems can be efficiently iteratively solved using Sinkhorn–Knopp iteration ([Sinkhorn and Knopp, 1967](#); [Cuturi, 2013](#)) with complexity $\tilde{\mathcal{O}}(P^2)$ per problem ([Altschuler, Weed and Rigollet, 2017](#)).

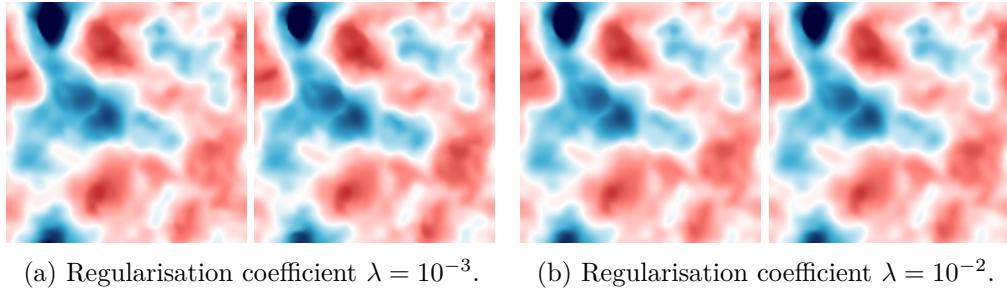


Fig 3: Examples of the [Cheng and Reich \(2015\)](#) local ETPF assimilation update applied to the same spatial Gaussian process model as Figure 1 using entropically regularised OT maps for different values of the regularisation coefficient λ .

Figs. 3a and 3b show examples of posterior fields samples computed using entropically regularised local ETPF updates for two regularisation coefficients λ . It can be seen that introducing entropic regularisation increases the smoothness of the updated fields compared to the unregularised samples shown in Fig. 1e and that the level of smoothness increases with the regularisation coefficient λ .

However the increase in smoothness comes at the cost of a decreased diversity in the post-update particles as λ increases - in particular for the $\lambda = 10^{-2}$ case shown in Fig. 3b, the four samples shown appear almost identical. This is a consequence of the assimilation updates in the local ETPF linearly transforming by the OT maps as in Eq. (2.19) as opposed to resampling using binary random variables generated according to the resampling probabilities encoded by the OT maps. For the regularised OT problems in Eq. (3.17), as the regularisation coefficient $\lambda \rightarrow \infty$ we have that $\rho_{t,m}^{p,q} \rightarrow w_{t,m}^q \quad \forall p \in 1:P, q \in 1:P, m \in 1:M$. In this case applying the local ETPF assimilation update will tend to assigning the weighted mean of the state variables at each mesh-node to the post-update particles, and thus a lack of diversity or *under-dispersion* in the post-update particles.

[Acevedo, de Wiljes and Reich \(2017\)](#) proposed a variant of the ETPF which overcomes this under-dispersion issue when using entropically regularised OT maps. For each OT map a correction terms is computed which ensures the empirical covariance of the updated particles matches the values that would be obtained using the standard PF update. Although this *second-order accurate* ETPF scheme overcomes the under-dispersion issues when using entropically regularised OT maps, in localised variants the correction factors must be computed separately for the OT map associated with each mesh node, with the computation of each correction factor having a $\mathcal{O}(P^3)$ complexity, potentially negating any gains from using a cheaper Sinkhorn solver for the regularised OT problems.

In the review article of [Farchi and Bocquet \(2018\)](#) a local ETPF variant is proposed which computes OT maps for *blocks* of state variables rather than for each mesh node individually. Computing OT maps per-block rather than per-node potentially can give significant computational savings in higher spatial dimensions — for instance for three dimensional domains, even using cubic blocks which cover just two mesh nodes in each dimension would lead to a reduction in the number of OT problems needing to be solved by eight. In the numerical experiments in [Farchi and Bocquet \(2018\)](#) it was found however that the accuracy of the local

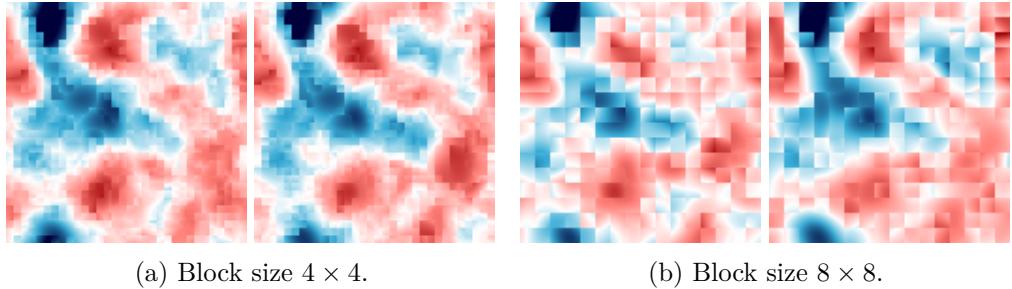


Fig 4: Examples of the Farchi and Bocquet (2018) local block ETPF assimilation update applied to the same Gaussian process model as Figure 1.

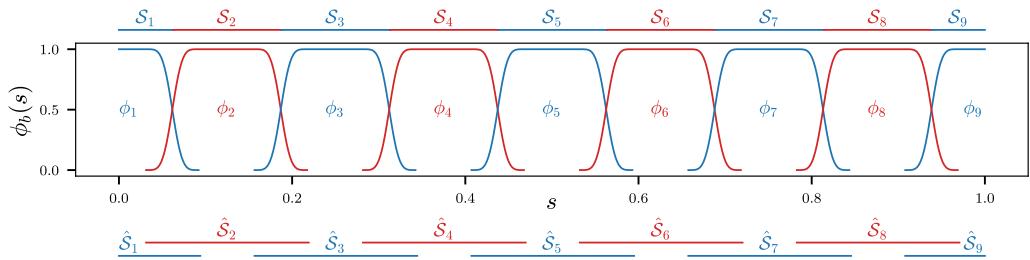


Fig 5: Example smooth partition of unity of a one-dimensional spatial domain $\mathcal{S} = [0, 1]$ with nine bump functions $\phi_{1:9}$. The patches $\hat{\mathcal{S}}_{1:9}$ covering \mathcal{S} and which the bump functions have support on are visualised below the plot axes.

block ETPF method was highest when using blocks containing just one mesh-node, i.e. corresponding to the local ETPF scheme of Cheng and Reich (2015). As the state variables in each block are updated independently given the computed per-block OT maps, the poorer performance with larger blocks may be at least in part due to the spatially inhomogeneous error introduced at the block boundaries. Fig. 4 shows examples of posterior state field samples computed using this block ETPF scheme for the earlier spatial Gaussian process example from Fig. 1 for two different block sizes; in both the boundaries of the blocks are clearly visible due to the discontinuities introduced into the fields.

4. SMOOTH AND SCALABLE LOCAL PARTICLE FILTERING

Grouping mesh nodes into spatially contiguous blocks and computing OT maps per-block rather than per-node as proposed in Farchi and Bocquet (2018) is a natural way to reduce the computational cost of local ETPF assimilation update. However this approach further decreases the smoothness of the updated fields. Here we propose an alternative approach. Rather than computing OT maps for disjoint blocks defining a partition of the spatial domain \mathcal{S} we instead ‘softly’ partition \mathcal{S} into patches with overlapping support, computing an OT map for each patch and smoothly interpolating between the OT maps associated with different patches in the overlaps. The construct we will use to both define the soft partitioning of the domain and interpolation across it is a *partition of unity*.

4.1 Partitions of unity

Let $\hat{\mathcal{S}}_{1:\mathbb{B}}$ be a cover of the spatial domain \mathcal{S} such that $\bigcup_{b=1}^{\mathbb{B}} \hat{\mathcal{S}}_b = \mathcal{S}$ with each $\hat{\mathcal{S}}_b$ termed a *patch*. We associate a *bump function* $\phi_b : \mathcal{S} \rightarrow [0, 1] \quad \forall b \in 1:\mathbb{B}$ with each patch with $\phi_b(s) = 0 \quad \forall s \notin \hat{\mathcal{S}}_b, b \in 1:\mathbb{B}$ and require that

$$\sum_{b \in 1:\mathbb{B}} \phi_b(s) = 1 \quad \forall s \in \mathcal{S}. \quad (4.1)$$

The set of bump functions $\phi_{1:\mathbb{B}}$ is then termed a *partition of unity* (POU) of \mathcal{S} . POU's are typically used to allow local constructions to be extended globally across a space, for instance an atlas of local charts of a manifold. Generally in such applications the bump functions will be required to be infinitely differentiable. Here we will generally not require such stringent differentiability requirements, however we will informally refer to a *smooth* POU for the case where each bump function is of at least class \mathcal{C}^1 with continuous derivatives, and to a *hard* POU for the case where the cover $\hat{\mathcal{S}}_{1:\mathbb{B}}$ is exact, i.e. the patches are pairwise disjoint, and so the bump functions are indicators on the patches $\phi_b(s) = \mathbb{1}_{\hat{\mathcal{S}}_b}(s) \quad \forall b \in 1:\mathbb{B}$.

A useful method for constructing a POU with specified smoothness properties on an arbitrary spatial domain is via convolution. Specifically, if $\mathcal{S}_{1:\mathbb{B}}$ is a partition of \mathcal{S} and $\varphi : [0, \infty) \rightarrow [0, \infty)$ is a non-negative *mollifier* function satisfying

$$\int_{\mathcal{S}} \varphi \circ d(s, s') \, ds' = 1 \quad \forall s \in \mathcal{S} \quad (4.2)$$

then we can define a POU $\phi_{1:\mathbb{B}}$ on \mathcal{S} by convolving φ with the indicators on $\mathcal{S}_{1:\mathbb{B}}$

$$\phi_b(s) = \int_{\mathcal{S}} \mathbb{1}_{\mathcal{S}_b}(s) \varphi \circ d(s, s') \, ds' \quad \forall b \in 1:\mathbb{B}, s \in \mathcal{S}. \quad (4.3)$$

The bump functions will then inherit any smoothness properties of the mollifier. Figure 5 shows an example of a smooth POU constructed in this manner.

4.2 Constructing smooth local linear ensemble transform filters

We can use a POU to define a local LETF that uses transform coefficients computed for each patch rather than mesh node. We define the per-node transform coefficients $a_{t,m}^{1:\mathbb{P},1:\mathbb{P}}$ in Eq. (3.7) in terms of a set of per-patch coefficients $\hat{a}_{t,1:\mathbb{B}}^{1:\mathbb{P},1:\mathbb{P}}$ by

$$a_{t,m}^{p,q} = \sum_{b \in 1:\mathbb{B}} \hat{a}_{t,b}^{p,q} \phi_b(s_m) \quad \forall p \in 1:\mathbb{P}, q \in 1:\mathbb{P}, m \in 1:\mathbb{M}. \quad (4.4)$$

If the set of coefficients $\hat{a}_{t,b}^{1:\mathbb{P},1:\mathbb{P}}$ for each patch index $b \in 1:\mathbb{B}$ correspond to the elements of a left stochastic matrix such that

$$\hat{a}_{t,b}^{p,q} \in [0, 1] \quad \forall p \in 1:\mathbb{P}, q \in 1:\mathbb{P} \quad \text{and} \quad \sum_{q \in 1:\mathbb{P}} \hat{a}_{t,b}^{p,q} = 1 \quad \forall p \in 1:\mathbb{P}, \quad (4.5)$$

then due to the non-negativity and sum to unity properties of the POU we have that $a_{t,m}^{p,q} \in [0, 1] \quad \forall p \in 1:\mathbb{P}, q \in 1:\mathbb{P}, m \in 1:\mathbb{M}$ and

$$\sum_{q \in 1:\mathbb{P}} a_{t,m}^{p,q} = \sum_{b \in 1:\mathbb{B}} \sum_{q \in 1:\mathbb{P}} \hat{a}_{t,b}^{p,q} \phi_b(s_m) = \sum_{b \in 1:\mathbb{B}} \phi_b(s_m) = 1 \quad \forall p \in 1:\mathbb{P}, m \in 1:\mathbb{M}, \quad (4.6)$$

and so that $a_{t,1:\mathbb{M}}^{1:\mathbb{P},1:\mathbb{P}}$ also correspond to the elements of left stochastic matrices.

The resulting assimilation update in terms of the values of the predictive and filtering distribution state field particles, $\vec{z}_t^{1:P}$ and $z_t^{1:P}$, at the mesh nodes $s_{1:M}$ is

$$z_t^p(s_m) = \sum_{q \in 1:P} \sum_{b \in 1:B} \hat{a}_{t,b}^{p,q} \phi_b(s_m) \vec{z}_t^q(s_m) \quad \forall p \in 1:P, m \in 1:M. \quad (4.7)$$

For a smooth POU $\phi_{1:B}$ the $P \times B$ spatial fields defined by the pointwise products $\phi_b(s) \vec{z}_t^q(s) \forall b \in 1:B, q \in 1:P$ will be smooth functions of the spatial coordinate $s \in \mathcal{S}$ if the predictive distribution state field particles $\vec{z}_t^{1:P}$ are themselves smooth. Each filtering distribution state field particle $z_t^{1:P}$ is then formed as a convex combination of these pairwise product fields, and so will also be smooth if the POU and predictive distribution state fields are. This is illustrated for a one-dimensional example in Fig. C.1 in Appendix C.

4.3 Smooth local ensemble transform particle filtering

We now consider the specific application of the smooth local LETF scheme to define a smooth localisation of the ETPF, with in this case the coefficients $\hat{a}_{t,1:B}^{1:P,1:P}$ corresponding to OT maps computed for each patch. We first define the following notation for the distance between a subset of the spatial domain and a point.

$$\underline{d}(\mathcal{S}', s) = \inf_{s' \in \mathcal{S}'} d(s', s) \quad \forall s \in \mathcal{S}, \mathcal{S}' \subseteq \mathcal{S}. \quad (4.8)$$

Analogously to the per-node case in Eq. (3.10), the logarithms of the per-patch (unnormalised) particle weights can then be defined by

$$\log \tilde{w}_{t,b}^p = \sum_{l \in 1:L} \log g_{t,l}(y_{t,l} | \vec{z}_t^p(s_l^o)) \ell_r(\underline{d}(\hat{\mathcal{S}}_b, s_l^o)) \quad \forall b \in 1:B, p \in 1:P, \quad (4.9)$$

As $\underline{d}(\hat{\mathcal{S}}_b, s_l^o) = 0$ if $s_l^o \in \hat{\mathcal{S}}_b$ and $\ell_r(0) = 1$ the weighted summation of log observation density terms in Eq. (4.9) gives weight one to all the terms corresponding to observations located within a patch. Observations outside a patch but within a distance of less than r are given weights between zero and one, and all observations more than a distance of r from a patch are given zero weight.

Taking inspiration from the per-node case in Eq. (3.15) we could define per-patch transport costs directly in terms of the predictive state fields $\vec{z}_t^{1:P}$

$$c_{t,b}^{p,q} = \int_{\mathcal{S}} |\vec{z}_t^p(s) - \vec{z}_t^q(s)|_2^2 \ell_r'(\underline{d}(\hat{\mathcal{S}}_b, s_m)) ds \quad \forall b \in 1:B, p \in 1:P, q \in 1:P. \quad (4.10)$$

Although this is defined independently of the spatial discretisation used, evaluating the integrals exactly will often be intractable. Assuming the common case of equally spaced mesh nodes, we propose to define per-patch transport costs as

$$c_{t,b}^{p,q} = \sum_{m \in \mathcal{M}} \left| \vec{x}_{t,m}^p - \vec{x}_{t,m}^q \right|_2^2 \mathbb{1}_{\hat{\mathcal{S}}_b}(s_m) \quad \forall b \in 1:B, p \in 1:P, q \in 1:P, \quad (4.11)$$

where $\mathcal{M} \subseteq 1:M$ corresponds to a spatial subsampling of the mesh nodes, e.g. corresponding to every K^{th} node in each spatial dimension, such that $|\mathcal{M}| \approx \frac{M}{K^D}$. This spatial subsampling is motivated by the observation that if the state fields are spatially smooth then the values at immediately adjacent mesh nodes will typically be very similar and there is therefore minimal loss of information in computing pointwise differences over a subset of, rather than all, mesh nodes. In

addition to spatial subsampling we also define the transport costs in Eq. (4.11) with the fixed choice of a uniform localisation function ℓ'_r , with $r' = 0$. Empirically we found varying the choice of ℓ'_r , and r' for the transport costs had little discernable effect on filtering performance.

Given per-patch weights $w_{t,1:B}^{1:P}$ and transport costs $c_{t,1:B}^{1:P,1:P}$ computed as described above, the per-patch linear transform coefficients $\hat{a}_{t,1:B}^{1:P,1:P}$ are then computed as solutions to the B corresponding OT problems

$$\hat{a}_{t,b}^{1:P,1:P} = \operatorname{argmin}_{\varrho^{1:P,1:P} \in \mathcal{R}(w_{t,b}^{1:P})} \sum_{p \in 1:P} \sum_{q \in 1:P} \varrho^{p,q} c_{t,b}^{p,q} \quad \forall b \in 1:B. \quad (4.12)$$

We will subsequently refer to instances of this framework as *smooth local ensemble transform particle filters* (SLETPFs). To define a SLETPF method for a given spatial SSM, we need to specify: a localisation function and radius ℓ_r and r to compute the local weights; the set of mesh nodes \mathcal{M} to use in computing the local transport costs; a POU of the spatial domain.

For the SLETPF local weight calculation in Eq. (4.9), the number of non-zero log observation density terms in the sum is dependent on both the localisation function and the size of the patches $\hat{\mathcal{S}}_{1:B}$ used to define the POU. We can define an effective number of observations considered per patch as

$$n_b = \sum_{l \in 1:L} \ell_r(d(\hat{\mathcal{S}}_b, s_l^o)) \quad \forall b \in 1:B. \quad (4.13)$$

To avoid weight degeneracy we will typically need to control the $n_{1:B}$ values through the choice of POU and localisation radius r , with the results of [Rebeschini and van Handel \(2015\)](#) suggesting $n_{1:B}$ should roughly scale with $\log P$. To approximately minimise $\max(n_{1:B})$ for a given number of patches B , as a heuristic we suggest the patches should be chosen such that each contains a roughly equal number of observations. We discuss approaches for defining a partition of the spatial domain based on the observation locations to achieve this in Appendix D.

The choice of the number of patches B to use will typically be based on a tradeoff between several factors. Reducing computational cost favours using fewer patches, while the need to control $\max(n_{1:B})$ and so the tendency for weight degeneracy favours using a greater number of smaller patches. More complex is the dependency of the approximation error introduced by localisation. Using larger patches and a greater number of observations to update the state variables within each patch should reduce the approximation error for the updates within each patch. However for a fixed r using larger patches will also lead to great disparities in the local weights calculated for each patch using Eq. (4.9) and so the transform coefficients for adjacent patches. If using a hard POU this will typically lead to spatial discontinuities in the state particles across patch boundaries after applying the assimilation update, with the downstream effect of such discontinuities potentially negating any reduction in the approximation error within the patches.

If using a smooth POU the mesh nodes in the overlaps between patches will be updated using a interpolation of the transform coefficients for each of the patches, allowing smaller numbers of patches B to be used while still retaining smoothness. In the numerical experiments in Section 5 we show that using a smooth POU allows use of a number of patches B less than the number of mesh nodes M while still retaining accurate filtering distribution estimates.

4.4 Computational cost

The computational cost of the per-node local ETPF assimilation updates proposed in [Cheng and Reich \(2015\)](#) is dominated by solving the M OT problems leading to an overall $\tilde{\mathcal{O}}(MP^3)$ scaling for the computational cost. For the SLETPF, the number of OT problems is determined by the number of patches B and so the cost of solving the OT problems is $\tilde{\mathcal{O}}(BP^3)$. When $B \ll M$ the relative cost of the other computations in the overall assimilation update can become significant however. To derive a relationship for the overall scaling of the computational cost of the proposed SLETPF we make the following assumptions.

ASSUMPTION 1. *The maximum number of patches covering any mesh node is independent of and much smaller than B and so the sum across all patches of the number of mesh nodes within each patch scales independently of B , i.e.*

$$\sum_{b \in 1:B} \sum_{m \in 1:M} \mathbb{1}_{\hat{\mathcal{S}}_b}(s_m) = \mathcal{O}(M). \quad (4.14)$$

For POUs in which each patch overlaps with only a fixed number of ‘neighbour’ patches this will hold. If a uniform subsampling scheme is used to define the set of mesh node indices \mathcal{M} used in computing the transport costs, then as a corollary we will also have that the total number of subsampled mesh nodes contained within all patches scales independently of B , i.e.

$$\sum_{b \in 1:B} \sum_{m \in \mathcal{M}} \mathbb{1}_{\hat{\mathcal{S}}_b}(s_m) = \mathcal{O}(|\mathcal{M}|). \quad (4.15)$$

ASSUMPTION 2. *The sum across all patches of the number observations within a distance r from a patch is less than the number of mesh nodes $M > L$, i.e.*

$$\sum_{b \in 1:B} \sum_{l \in 1:L} \mathbb{1}_{[0,r]}(\underline{d}(\hat{\mathcal{S}}_b, s_l^o)) < M. \quad (4.16)$$

We will typically have that the number of observations locations L is small compared to the number of mesh nodes M and the localisation radius r will be set to limit the number of observations considered per patch to a small subset of all observations so this will usually hold.

Under Assumption 1 the cost of calculating the BP^2 transport costs using Eq. (4.11) is $\mathcal{O}(|\mathcal{M}|P^2)$ as we need to evaluate the distance between the $P(P - 1)$ pairs of particles at $|\mathcal{M}|$ mesh nodes and from Eq. (4.15) only $\mathcal{O}(|\mathcal{M}|)$ terms in the summations for each of the particle pairs need to be evaluated.

The update to the particles in Eq. (4.7) for a general set of per-patch linear transform coefficients $\hat{a}_{t,1:B}^{1:P,1:P}$ will have a cost of $\mathcal{O}(MP^2)$ under Assumption 1. However for transform coefficients computed as the solution to discrete OT problems, at most $2P - 1$ of the P^2 coefficients for each patch are non-zero ([Reich, 2013](#)). In this case the assimilation update in Eq. (4.7) therefore has a $\mathcal{O}(MP)$ cost.

Under Assumption 2, the computation using Eq. (4.9) of the BP per-patch weights will cost less than $\mathcal{O}(MP)$ as we need to evaluate $LP < MP$ log observation density factors, and from Eq. (4.16) less than M terms in the summations for each of the P particles will be non-zero and so need to be evaluated.

Under these assumptions, the overall computational cost of each SLETPF assimilation step therefore scales as $\tilde{\mathcal{O}}(BP^3 + |\mathcal{M}|P^2 + MP)$.

5. NUMERICAL EXPERIMENTS

To evaluate the performance of the proposed approach, we perform filtering in two SPDE test models, comparing our proposed scheme to the local ETPF (Cheng and Reich, 2015) and local ETKF (Hunt, Kostelich and Szunyogh, 2007). Rather than measure performance in terms of the distance between the estimated mean of the filtering distribution and the true state used to generate the observations, as is common in similar work e.g. Farchi and Bocquet (2018), here we measure the errors in the ensemble estimates of expectations with respect to the true filtering distributions. This gives more directly interpretable results as a filter which exactly computes the expectations would give a zero error, unlike the difference between the mean and true state which will in general be non-zero even if the mean is computed exactly. We are also able to assess the accuracy of a broader range of features of the filtering distribution estimates, for example their quantification of uncertainty via measures of dispersion.

To allow such comparisons, we require models for which ground truth values for expectations with respect to the filtering distributions can be computed. To this end our first model is based on a linear-Gaussian SPDE model for which the true filtering distribution can be exactly computed using a Kalman filter. For the second model, we use a more challenging SPDE model with non-linear state dynamics. Here our ‘ground-truth’ for the filtering distributions is based on long runs of a *Markov chain Monte Carlo* (MCMC) method.

5.1 Evaluating the accuracy of filtering estimates

For both models we consider several metrics for evaluating the accuracy of the different local ensemble filters’ estimates of the filtering distributions.

The first two metrics we consider are the time- and space-averaged *root mean squared errors* (RMSEs) of the ensemble estimates of the filtering distributions means and standard deviations, to reflect respectively the filters’ accuracy in estimating the central tendencies and dispersions of the filtering distributions. Denote $\mu_{1:T}$ and $\sigma_{1:T}$ as the true means and standard deviations under $\pi_{1:T}$

$$\mu_t = \int_{\mathcal{X}} x \pi_t(dx) \quad \text{and} \quad \sigma_t^2 = \int_{\mathcal{X}} (x - \mu_t) \odot (x - \mu_t) \pi_t(dx) \quad \forall t \in 1:T, \quad (5.1)$$

and $\hat{\mu}_{1:T}$ and $\hat{\sigma}_{1:T}$ as the corresponding means and standard deviations under the empirical ensemble estimates to the filtering distributions $\hat{\pi}_t(dx) = \sum_{p=1}^P \delta_{x_t^p}(dx)$,

$$\hat{\mu}_t = \int_{\mathcal{X}} x \hat{\pi}_t(dx) \quad \text{and} \quad \hat{\sigma}_t^2 = \int_{\mathcal{X}} (x - \hat{\mu}_t) \odot (x - \hat{\mu}_t) \hat{\pi}_t(dx) \quad \forall t \in 1:T. \quad (5.2)$$

We then define the time- and space-averaged RMSEs of the estimates as

$$\text{RMSE}(\hat{\mu}_{1:T}, \mu_{1:T}) = \sqrt{\frac{1}{TM} \sum_{t \in 1:T} \sum_{m \in 1:M} (\hat{\mu}_{t,m} - \mu_{t,m})^2}, \quad (5.3)$$

$$\text{and} \quad \text{RMSE}(\hat{\sigma}_{1:T}, \sigma_{1:T}) = \sqrt{\frac{1}{TM} \sum_{t \in 1:T} \sum_{m \in 1:M} (\hat{\sigma}_{t,m} - \sigma_{t,m})^2}. \quad (5.4)$$

In both cases lower values of these metrics are better, with a value of zero indicating the mean or standard deviation estimates exactly match the true values.

The two metrics discussed so far concentrate on the accuracy of estimates of local properties of the states, but do not reflect more global properties such as whether the ensemble filters correctly estimate the smoothness of the state fields. As a proxy measure for smoothness we use the expectation under the true filtering distributions of a finite-difference approximation of the integral across space of the magnitude of the spatial gradients of the state fields:

$$\gamma_t = \int_{\mathcal{X}} \sum_{m \in 1:\mathbb{M}} |x_{t,m} - x_{t,m \oplus 1}| \pi_t(dx) \approx \mathbb{E} \left[\int_{\mathcal{S}} |\partial_s z_t(s)| ds \right] \quad \forall t \in 1:T, \quad (5.5)$$

with $m \oplus 1$ here indicating $m + 1 \bmod \mathbb{M}$, with one-dimensional periodic spatial domains being used in both models considered. Defining the estimates $\hat{\gamma}_{1:T}$ of these *smoothness coefficients* under the ensemble filtering distributions equivalently as

$$\hat{\gamma}_t = \int_{\mathcal{X}} \sum_{m \in 1:\mathbb{M}} |x_{t,m} - x_{t,m \oplus 1}| \hat{\pi}_t(dx) \quad \forall t \in 1:T, \quad (5.6)$$

we then define an overall measure of the accuracy of the ensemble estimates' spatial smoothness as the following time-averaged RMSE

$$\text{RMSE}(\hat{\gamma}_{1:T}, \gamma_{1:T}) = \sqrt{\frac{1}{T} \sum_{t \in 1:T} (\hat{\gamma}_t - \gamma_t)^2}. \quad (5.7)$$

5.2 Stochastic turbulence model

As our first example we use a linear-Gaussian SSM derived from a SPDE model for turbulent signals by [Majda and Harlim \(2012, Ch. 5\)](#). The governing SPDE is

$$d\zeta(s, \tau) = (\theta_1 \partial_s^2 + \theta_2 \partial_s - \theta_3) \zeta(s, \tau) d\tau + (\kappa * \eta)(s, \tau), \quad (5.8)$$

where $\zeta : \mathcal{S} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is a real-valued space-time varying process, $\theta_1 \in \mathbb{R}_{\geq 0}$ is a non-negative parameter controlling dissipation due to diffusion, $\theta_2 \in \mathbb{R}$ is a parameter governing the direction and magnitude of the constant advection, $\theta_3 \in \mathbb{R}_{\geq 0}$ is a non-negative parameter controlling dissipation due to damping, $\kappa : \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is a spatial kernel function which governs the spatial smoothness of the additive noise in the dynamics and $\eta : \mathcal{S} \times \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}$ is a space-time varying noise process. The spatial domain is a one-dimensional interval $\mathcal{S} = [0, 1]$ with periodic boundary conditions and a distance function $d(s, s') = \min(|s - s'|, 1 - |s - s'|)$, and $*_{\mathcal{S}}$ represents circular convolution in space.

We use a spectral approach to define basis function expansions of the processes ζ and η and kernel κ using $\mathbb{M} = 512$ mesh nodes. This results in a linear system of *stochastic differential equations* (SDEs) for which the Gaussian state transition and stationary distributions can be solved for exactly. We assume a linear-Gaussian observation model with the state noisily observed at $L = 64$ locations and $T = 200$ time points. Full details of the model are given in Appendix F.1.

The resulting *stochastic turbulence* (ST) SSM is linear-Gaussian. We consider two cases in our experiments: inference in the original linear-Gaussian SSM, and inference in a *transformed* SSM using this linear-Gaussian model as the base SSM. The specific definition we use for a transformed SSM is given in Appendix E however in brief, by applying a non-linear transformation to the state of a linear-Gaussian SSM we can construct a SSM with non-Gaussian filtering distributions

	RMSE($\hat{\mu}_{1:T}, \mu_{1:T}$)	RMSE($\hat{\sigma}_{1:T}, \sigma_{1:T}$)	RMSE($\hat{\gamma}_{1:T}, \gamma_{1:T}$)
Minimum	4.34×10^{-2}	1.37×10^{-2}	7.40×10^{-4}
Median	4.38×10^{-2}	1.38×10^{-2}	8.18×10^{-4}
Maximum	4.43×10^{-2}	1.40×10^{-2}	9.13×10^{-4}
Localisation radius r	0.030	0.034	0.024

TABLE 1

Values of metrics at optimal localisation radii for local ETKF on linear-Gaussian ST model.

for which we can tractably estimate expectations with respect to the true filtering distributions with arbitrary accuracy. Here the nonlinear transformation is chosen as $T(x) = \sinh^{-1}(\theta_4 x)$ (with \sinh^{-1} evaluated elementwise on vector arguments). As $|\sinh^{-1}(\theta_4 x)| \approx \log(2\theta_4|x|)$ for $|\theta_4 x| \gg 1$ this non-linearity has the effect of compressing the variation in large magnitude values, while expanding small magnitude values, and so for an appropriate choice of scaling factor θ_4 tends to induce bimodality in the marginals of the transformed filtering distributions.

For both the transformed and linear-Gaussian cases we use the model parameter settings give in Table F.1 and use simulated noisy observations $y_{1:T}$ generated from the models using a shared set of Gaussian state and observation noise variable samples generated using a pseudo-random number generator. The resulting observation sequence $y_{1:T}$ (which is the same for both models) is shown in Fig. F.1 along with the corresponding true state sequences $z_{1:T}$ and $z'_{1:T}$ used to generate the observations under the linear-Gaussian and transformed SSMS respectively.

We compare the performance of the local ETKF, local ETPF and our proposed SLETPF algorithm in estimating the filtering distributions for both the linear-Gaussian and transformed SSMS. The mesh size $M = 512$ and number of observations $L = 64$ are sufficiently large that non-local PF methods suffer from weight degeneracy even with large ensembles of up to $P = 10^4$ particles for both the linear-Gaussian and transformed SSMS. While non-local variants of the ENKF do not suffer from weight degeneracy and can give relatively accurate filtering distribution estimates for an ensemble size of $P \geq 10^3$, this is still much larger than the ensemble sizes typically used in for example NWP ensemble filter systems. For an ensemble size $P = 10^2$ we found the local ETKF significantly outperformed the non-local ETKF on all the metrics we consider in both the linear-Gaussian and transformed SSMS. We used $P = 10^2$ for all methods in the experiments here.

For the local ETKF we use the smooth compact Gaspari and Cohn localisation function ℓ_r defined in Eq. (3.12). We conducted a grid search over localisation radii $r \in \{0.010, 0.012, \dots, 0.160\}$, for each r performing five independent runs of the local ETKF and recording the performance on the three metrics described in Section 5.1. The results for the linear-Gaussian ST model are summarised in Table 1 and for the transformed ST model in Table 2. For each metric the minimum, median and maximum value recorded across the five runs is shown, for the value of r which gave the minimum median value of that particular metric. The results for all r values are shown in the Appendix in Fig. G.1.

The performance on all metrics for both models was relatively stable across the multiple runs. Unsurprisingly the local ETKF performs significantly better on the linear-Gaussian ST model than the transformed ST model. While for the linear-Gaussian ST model the optimal r for each metric are relatively similar, for the transformed ST model the optimal r differs significantly across the metrics meaning any choice of r will incur a performance penalty on some metrics.

	RMSE($\hat{\mu}_{1:T}, \mu_{1:T}$)	RMSE($\hat{\sigma}_{1:T}, \sigma_{1:T}$)	RMSE($\hat{\gamma}_{1:T}, \gamma_{1:T}$)
Minimum	1.71×10^{-1}	1.93×10^{-1}	1.04×10^{-2}
Median	1.72×10^{-1}	1.94×10^{-1}	1.04×10^{-2}
Maximum	1.74×10^{-1}	1.95×10^{-1}	1.05×10^{-2}
Localisation radius r	0.030	0.152	0.160

TABLE 2

Values of metrics at optimal localisation radii for local ETKF on transformed ST model.

For our proposed SLETPF framework we need to choose a POU. Here we construct the POUs by (discretely) convolving a mollifier function with the indicator functions on a partition of the spatial domain. As the observations are located on a regular grid, we partition the domain into B equally sized intervals $\mathcal{S}_b = [\frac{b-1}{B}, \frac{b}{B}) \forall b \in 1:B$. For the mollifier function φ we use a normalised variant of the compactly supported Gaspari and Cohn localisation function ℓ_r in Eq. (3.12), the bump functions then defined as

$$\phi_b(s_n) = \sum_{m \in 1:M} \mathbb{1}_{\mathcal{S}_b}(s_m) \frac{\ell_w \circ d(s_n, s_m)}{\sum_{m' \in 1:M} \ell_w \circ d(s_n, s_{m'})} \quad \forall n \in 1:M, b \in 1:B \quad (5.9)$$

with w a *kernel width* parameter determining how many mesh nodes the effective smoothing kernel being discretely convolved with the indicators has support on. For $w = M^{-1}$ the kernel is only non-zero at one mesh node, and no smoothing is applied, corresponding to a hard partition of the space. For $w > M^{-1}$, the amount of smoothing and overlap between the patches increases with w .

For the experiments with the ST models we performed runs with SLETPFs with POUs with five different numbers of patches $B \in \{2^5, 2^6, 2^7, 2^8, 2^9\}$ and four different kernel widths $w \in \{512^{-1}, 256^{-1}, 128^{-1}, 64^{-1}\}$. We used a Gaspari and Cohn localisation function for the local weight calculation, for each (B, w) pair performing five independent runs for all localisation radii $r \in \{0.001, 0.002, \dots, 0.030\}$ where $\text{median}(n_{1:B})$ was in the range $[1, 5]$. As noted previously the local ETPF of Cheng and Reich (2015) can be considered a particular instance of the SLETPF framework, here corresponding to the runs with a POU with $B = 512$ patches and $w = 512^{-1}$. The set of mesh nodes M used to calculate the per-patch transport costs as in Eq. (4.11) was constructed by subsampling $1:M$ by a factor $\min(4, p_n)$ with $p_n = M(B^{-1} + 2w) - 1$ the number of mesh nodes in each patch, ensuring that at least one node per patch was used to compute the transport costs.

The values of the three metrics recorded across all SLETPF runs for each of the (B, w, r) parameter combinations are shown for the linear-Gaussian ST model in Fig. 6 and for the transformed ST model in Fig. 7. In each figure, the rows of plots correspond to different kernel widths w and the three columns to different metrics. On each plot the value of the relevant metric on the vertical axis is plotted against the median number of effective observations per patch on the horizontal axis (we plot against $\text{median}(n_{1:B})$ rather than r as it is more directly comparable across different values of B and w). The median values across the five independent runs for each of the numbers of patches B are shown by the coloured curves (see colour key at top of figures) and the surrounding lighter coloured regions indicated minimum to maximum range of values recorded across the runs (in many cases the across-run variation is too small to be visible). For each metric the best value achieved by the local ETKF (as given in Tables 1 and 2) for the metric is indicated by the black horizontal dashed line.

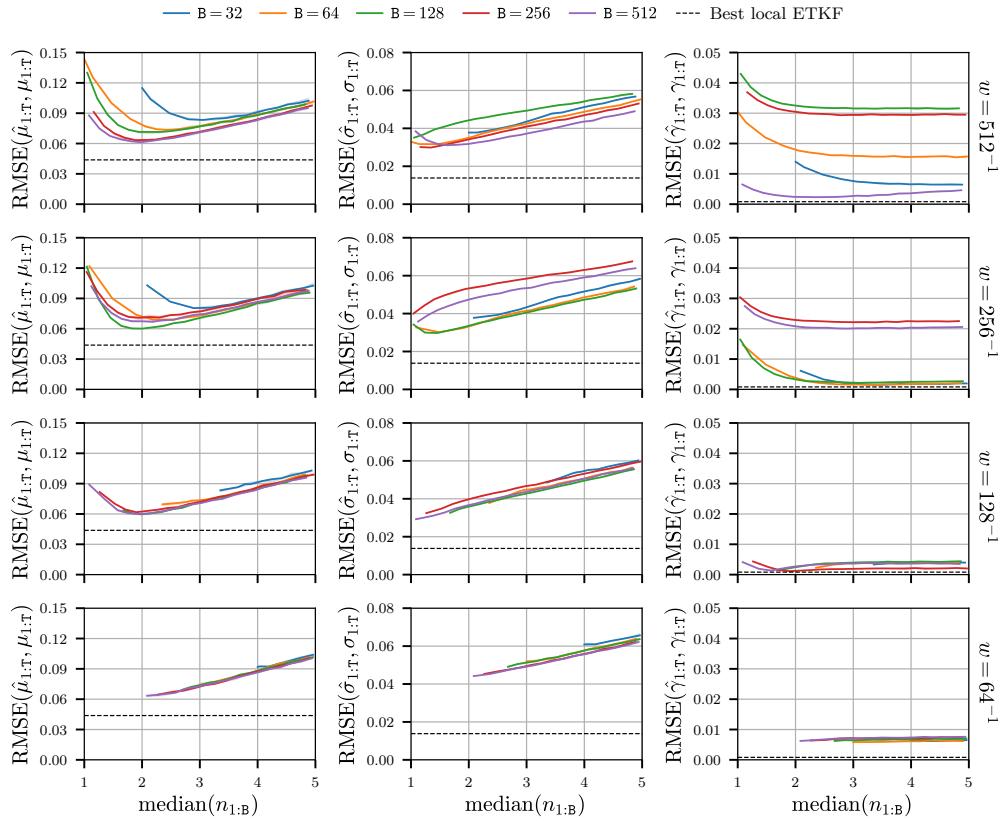


Fig 6: Comparison of accuracy of SLETPF estimates on linear-Gaussian ST SSM.

Considering first the linear-Gaussian ST model results, we see that across all parameter combinations and metrics the local ETPF methods are outperformed by the best local ETKF results. This is as expected as the linear-Gaussian assumptions made by the ETKF are correct in this case, and by better exploiting this model structure we expect the local ETKF to outperform the more generic local ETPF.

Concentrating on the results for filters with hard POU without smoothing in the first row ($w = 512^{-1}$), we see that the filters with $B = M = 512$ patches in the POU, corresponding to the [Cheng and Reich \(2015\)](#) scheme, outperform filters using POUs with smaller numbers of patches across virtually all $\text{median}(n_{1:B})$ values and metrics. This tallies with the findings of [Farchi and Bocquet \(2018\)](#) who found that for an equivalent ‘block’-based local ETPF scheme the best performance was always achieved with blocks of size one. Considering specifically the $\text{RMSE}(\hat{\mu}_{1:T}, \mu_{1:T})$ metric we see that as the number of patches B decreases the value of the metric across all values of $\text{median}(n_{1:B})$ monotonically increases (corresponding to poorer performance). The behaviours for the $\text{RMSE}(\hat{\sigma}_{1:T}, \sigma_{1:T})$ and $\text{RMSE}(\hat{\gamma}_{1:T}, \gamma_{1:T})$ metrics are more complex. For the smoothness coefficient we see that accuracy of the filter estimates initially decreases as the number of patches is increased from $B = 512$ to $B = 256$ and $B = 128$. The accuracy of the smoothness estimates however then increases on decreasing the number of patches further to $B = 64$ and again the accuracy increases on decreasing the number of patches to $B = 32$. We believe this non-monotonic relationship between the accuracy of the

smoothness estimates and the number of patches in the POU may be explained by the spatial averaging in the computation of the smoothness coefficient: while using fewer larger patches in the POU would be expected to introduce stronger discontinuities at the patch boundaries due to larger differences in the local weights assigned to each patch, there is a competing effect that as fewer patches are used there are fewer boundaries and so the spatially averaged error becomes lower despite the individual discontinuities at each block boundary being larger.

Now comparing the results as the kernel width w and so smoothness of the POU is increased, there are two main trends apparent. Most prominently the variation in performance across different numbers of patches B decreases as the smoothness of the POU increases, with many of the curves overlapping over much of their ranges for $w = 128^{-1}$ and $w = 64^{-1}$, while the optimal performance on each metric remains similar. This suggests using smooth POU allows fewer number of patches to be used (and thus a lower computational cost of the assimilation update) while maintaining performance, contrary to what was observed for the hard POU case where using fewer patches always decreased performance.

A second less obvious effect is that as the kernel width w is increased the lower limit for median($n_{1:B}$) is increased (similarly using fewer larger patches also increases the lower limit for median($n_{1:B}$)). This is the reason for the curves starting at higher median($n_{1:B}$) as the kernel width increases, corresponding to the values achieved with the smallest r tested ($r = 0.001$). In the case of the largest kernel width tested $w = 64^{-1}$ we see that all the curves start to the right of the point at which the optimal performance is reached for the other smaller w . This suggests there is a drawback to making w too large as it limits how far the number of observations per patch and so tendency to local weight degeneracy can be controlled; in this case it seems the best tradeoff is reached for either $w = 256^{-1}$ or $w = 128^{-1}$. Interestingly we also see that the accuracy of the smoothness and standard deviation estimates are poorer for $w = 64^{-1}$ compared to $w = 128^{-1}$ even when comparing at the same median($n_{1:B}$). This could be due to the greater overlap between the patches in this case, with the averaging of the particle values at the overlaps potentially acting to artificially oversmooth and reduce variation in the particles, again suggesting that the appropriate level of smoothing is a tradeoff between several factors.

The results on the transformed ST model shown in Fig. 7 show for the most part very similar trends as for the linear-Gaussian ST model. The most significant difference is the relative performance of the local ETKF and local ETPF methods, with in this case the local ETPF approaches outperforming the best local ETKF results across all parameter values for the $\text{RMSE}(\hat{\sigma}_{1:T}, \sigma_{1:T})$ and across a majority of the parameter values tested for the $\text{RMSE}(\hat{\mu}_{1:T}, \mu_{1:T})$ metric. As the only difference between these two models is the non-Gaussianity in the filtering distributions introduced by the transformation, these results support the earlier claims that PF-based methods such as the local ETPF and SLETPF proposed in this article, are more robust to non-Gaussianity than ENKF methods such as the local ETKF. Interestingly the relative performance loss in the local ETKF on introducing non-Gaussianity seems to be most severe in the $\text{RMSE}(\hat{\sigma}_{1:T}, \sigma_{1:T})$ metric, suggesting that uncertainty estimates provided by local ETKF methods on non-linear-Gaussian models should be particularly treated with caution.

In addition to the accuracy of the filter estimates, we are also interested in the

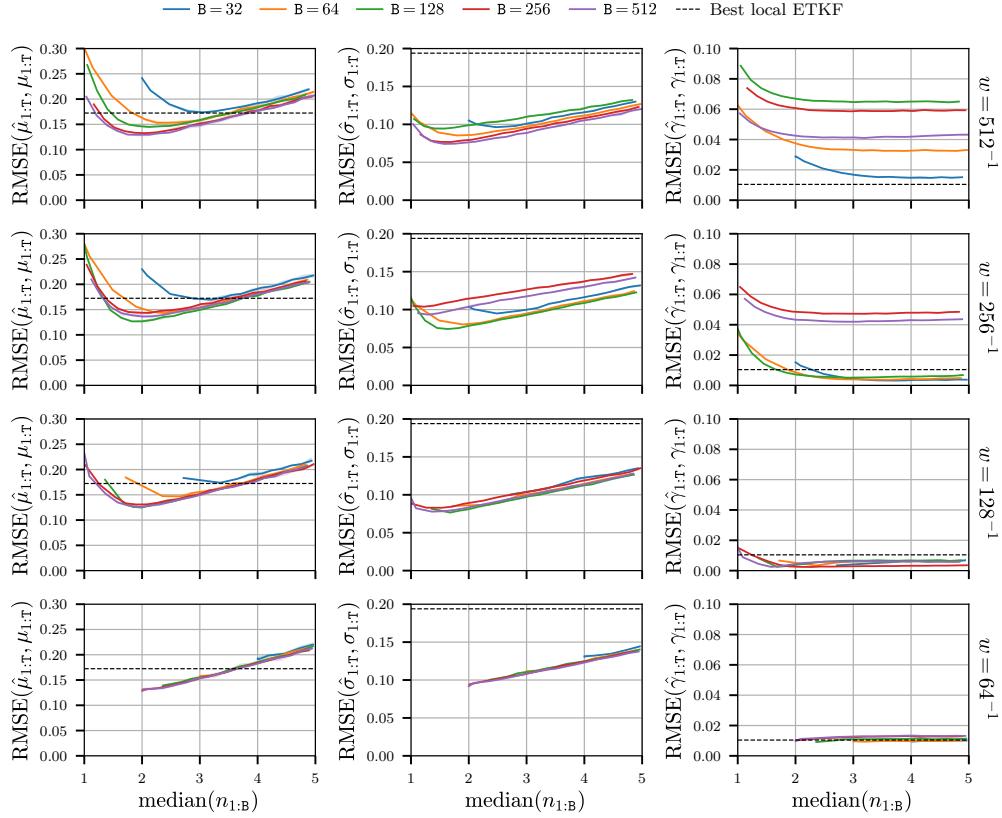


Fig 7: Comparison of accuracy of SLETPF estimates on transformed ST SSM.

relative computational cost of the different methods. Fig. 8 shows the values of the performance metrics achieved by the different SLETPF configurations tested, against the corresponding assimilation time (i.e. total filtering time minus the time taken to integrate the model dynamics in the prediction updates) for the transformed ST SSM. Each of the three plots corresponds to one of the performance metrics, the vertical coordinate of each marker indicates the minimum value of the metric achieved across all localisation radii r for a particular (B, w) combination, with the marker colour indicating the number of patches B , and the marker symbol the kernel width w . The horizontal coordinate of each marker indicates the median assimilation time across the five independent runs for the corresponding (B, w, r) values. For the POUS with $B = 512$ patches, only the case without smoothing ($w = 512^{-1}$), corresponding to the Cheng and Reich (2015) local ETPF, is shown, with the smoother POUS in this case substantially increasing the assimilation times without any gain in accuracy.

As would be expected due to the lower number of OT problems that need to be solved, in general the assimilation time decreases as the number of patches B in the POU is decreased *for a fixed smoothing kernel width w* . Note however that the assimilation time increases with the smoothing kernel width w (primarily due to the increased number of non-zero terms in the summation in Eq. (4.7)), which results for example in the assimilation time for the scheme with $B = 256$ and $w = 64^{-1}$ (\times) being slightly larger than for the runs under the Cheng and

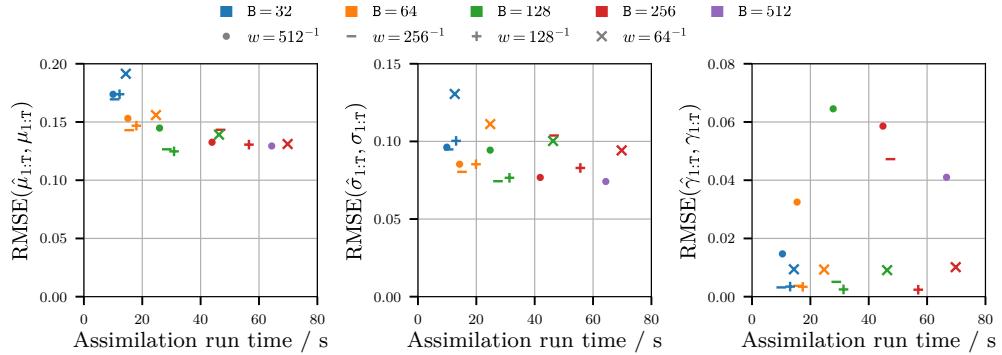


Fig 8: Accuracy versus run time for SLETPF in transformed ST SSM.

Reich (2015) settings of $B = 512$ and $w = 512^{-1}$ (\bullet). Although there is therefore a tradeoff in assimilation time between decreasing the number of patches B and increasing the kernel width w , we still find that there are combinations of (B, w) values which maintain the accuracy of the Cheng and Reich (2015) scheme while giving substantial reductions in assimilation time. In particular the runs with $B = 128$ and $w = 256^{-1}$ ($-$) and $B = 128$ and $w = 128^{-1}$ ($+$) achieve nearly identical accuracies on the mean and standard deviation RMSE metrics as $B = 512$ and $w = 512^{-1}$ (and a substantially improved smoothness coefficient RMSE) while reducing the assimilation time by slightly more than a factor of two. At the cost of around a 10% increase in the mean and standard deviation RMSEs, a more substantial reduction in the assimilation time by a factor of four can be achieved by using a POU with $B = 64$ patches and $w \in \{512^{-1}, 256^{-1}, 128^{-1}\}$.

Although the absolute values of the assimilation times in Fig. 8 are dependent on the computational environment used to run the experiments, the relative timings should still be informative as the same SLETPF implementation was used to run all the experiments. We purposefully did not include the local ETKF runs on the plots as any differences in the assimilation times for the local ETKF versus SLETPF approaches are likely to be as much due to the particulars of the software implementations and hardware used as any fundamental differences in performance. In particular more time was spent optimising the implementation of the SLETPF algorithm than our local ETKF implementation so the relative timings are likely to unfairly favour the SLETPF runs. The computational complexity for the local ETKF however is $\mathcal{O}(MP^3)$ which is the same as for the local ETPF scheme of Cheng and Reich (2015), so it would be expected that there are regimes in which the SLETPF assimilation updates (with complexity $\tilde{\mathcal{O}}(BP^3 + |\mathcal{M}|P^2 + MP)$) will have a computational advantage over the local ETKF updates.

5.3 Damped stochastic Kuramoto-Sivashinsky model

As our second test model we consider a stochastic variant of a fourth-order non-linear *partial differential equation* (PDE), often termed the *Kuramoto–Sivashinsky* (KS) equation, which has been independently derived as a model of various physical phenomena (Kuramoto and Tsuzuki, 1976; Sivashinsky, 1977) and studied as an example of a relatively simple PDE system exhibiting spatio-temporal chaos (Hyman and Nicolaenko, 1986). On a spatial domain $\mathcal{S} = [0, 1]$ with a distance function $d(s, s') = \min(|s - s'|, 1 - |s - s'|)$ and periodic boundary conditions, the

deterministic dynamics of the KS PDE model can be described by

$$\partial_\tau \zeta(s, \tau) = -\left(\frac{\partial_s^2}{\theta_1^2} + \frac{\partial_s^4}{\theta_1^4}\right) \zeta(s, \tau) - \frac{\partial_s}{2\theta_1} (\zeta^2) \quad (5.10)$$

where θ_1 is a length-scale parameter, with the system dynamics becoming chaotic for large values of θ_1 (Hyman and Nicolaenko, 1986).

As our focus is on filtering in models with stochastic dynamics, we use a related SPDE model on the same spatial domain, described by

$$d\zeta(s, \tau) = \left(-\left(\frac{\partial_s^2}{\theta_1^2} + \frac{\partial_s^4}{\theta_1^4} + \theta_2\right) \zeta(s, \tau) - \frac{\partial_s}{2\theta_1} (\zeta^2)\right) d\tau + (\kappa \circledast_s d\eta)(s, \tau) \quad (5.11)$$

where $\zeta : \mathcal{S} \times \mathcal{T} \rightarrow \mathbb{R}$ is a real-valued space-time varying process, $\theta_1 \in \mathbb{R}_{\geq 0}$ is the non-negative length-scale parameter, $\theta_2 \in \mathbb{R}_{\geq 0}$ is a non-negative parameter controlling dissipation due to damping, $\kappa : \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$ is a spatial kernel function and $\eta : \mathcal{S} \times \mathcal{T} \rightarrow \mathbb{R}$ is a space-time varying noise process. In addition to the introduction of the additive noise process, we also introduce a linear damping component controlled in magnitude by θ_2 . This is motivated by our empirical observation in simulations that the stochastic system can become unstable when numerically integrating over long time periods without additional dampening.

We use a similar spectral approach to define the spatial basis function expansions of the state and noise processes ζ and η and kernel κ as for the ST model, again using $M = 512$ mesh nodes. Full details of the discretisation used are given in Appendix F.2 and the values of all the parameters used in Table F.2. This results in a coupled non-linear system of SDES which governs the evolution of the state Fourier coefficients; unlike the linear-Gaussian dynamics of the ST model these SDES do not have an analytic solution and so need to be numerically integrated. We assume the state is observed at $T = 200$ time points, with $S = 10$ integrator steps performed between each observation time; the resulting state transition operators $F_{1:T}$ are non-linear and do not admit closed form transition densities.

We consider SSMS in which these KS state dynamics are noisily observed via both linear and non-linear observation operators. In both cases the state is assumed to be observed at $L = 64$ equispaced points in the spatial domain, with direct observations of the state values at these points in the linear case and via a hyperbolic tangent (\tanh) function in the non-linear case. The simulated state and observation sequences used in the experiments for both the linearly and non-linearly observed KS SSMSs are shown in Fig. F.2 (with the same simulated state sequence being used in both cases, with only the generated observations differing). Compared to ST model, the KS model exhibits more complex and unpredictable state dynamics and thus can be seen as more challenging test case for the local ensemble filtering methods.

Both the linearly and non-linearly observed KS SSMSs have non-Gaussian filtering distributions which cannot be exactly inferred unlike the linear-Gaussian ST model. We therefore used a MCMC method to generate proxy ground-truths for the filtering distributions, constructing Markov chains which left invariant the joint distribution across the $M = 512$ dimensional state vectors at all $T = 200$ time points given the observed sequence, i.e. $\mathbb{P}(x_{1:T} \in dx | y_{1:T} = y_{1:T})$, with the filtering distributions corresponding to marginals of this joint *smoothing distribution*. Due to the large overall state dimension $MT \approx 10^5$ we use a gradient-based

	RMSE($\hat{\mu}_{1:T}, \mu_{1:T}$)	RMSE($\hat{\sigma}_{1:T}, \sigma_{1:T}$)	RMSE($\hat{\gamma}_{1:T}, \gamma_{1:T}$)
Minimum	1.41×10^{-1}	3.39×10^{-2}	2.34×10^{-3}
Median	1.41×10^{-1}	3.40×10^{-2}	2.36×10^{-3}
Maximum	1.42×10^{-1}	3.40×10^{-2}	2.48×10^{-3}
Localisation radius r	0.068	0.160	0.092

TABLE 3

Values of metrics at optimal localisation radii for local ETKF on linearly observed KS model.

	RMSE($\hat{\mu}_{1:T}, \mu_{1:T}$)	RMSE($\hat{\sigma}_{1:T}, \sigma_{1:T}$)	RMSE($\hat{\gamma}_{1:T}, \gamma_{1:T}$)
Minimum	2.87×10^{-1}	1.04×10^{-1}	4.44×10^{-3}
Median	2.88×10^{-1}	1.04×10^{-1}	4.49×10^{-3}
Maximum	2.91×10^{-1}	1.05×10^{-1}	4.64×10^{-3}
Localisation radius r	0.064	0.156	0.020

TABLE 4

Values of metrics at optimal localisation radii for local ETKF on non-linearly observed KS model.

Hamiltonian Monte Carlo algorithm (Duane et al., 1987) to generate the chains. For each of the linear and non-linearly observed cases we ran five parallel chains of 200 samples each, with each chain using an independently seeded pseudo-random number generator. Details of the set up used for the MCMC runs are given in Appendix H. The ‘ground-truth’ values for the filtering distributions means $\mu_{1:T}$, standard deviations $\sigma_{1:T}$ and smoothness coefficients $\gamma_{1:T}$ were estimated using the combination of the final 100 $x_{1:T}$ samples of each of the five chains for each SSM, i.e. a total of 500 samples per SSM.

As for the ST SSMs, we used $P = 10^2$ particles for all the local ensemble filters runs on the KS SSMs. For the local ETKF we performed an equivalent grid search as for the ST models, performing five independent runs for each localisation radius $r \in \{0.010, 0.012, \dots, 0.160\}$ for both the linearly and non-linearly observed KS SSMs. The results are summarised in Tables 3 and 4, with plots of the full grid search results shown in Fig. G.1 in Appendix G.

Although the absolute values of the RMSE metrics in Table 3 are higher than for the local ETKF runs on the linear-Gaussian ST model, given the non-linear state dynamics in the KS model mean the filtering distributions are no longer constrained to remain Gaussian, the local ETKF performs remarkably well on the linearly-observed KS SSM, recovering relatively accurate estimates of the filtering distribution means, standard deviations and smoothness coefficients. This is concordant with the widespread empirical success of local ENKF approaches even when applied to models with non-linear state dynamics (Evensen, 2009), but also suggests that the filtering distributions in this case may have remained close to Gaussian despite the non-linear dynamics.

Swapping the linear observations for a non-linear observation operator however can be seen to have a detrimental effect on the accuracy of the local ETKF estimates of the filtering distributions. The optimal values achieved for each of the three RMSE metrics shown for the non-linearly observed case in Table 4 show significant increases over the corresponding figures for the linearly observed case in Table 3, with the errors in the standard deviation estimates showing the largest increase. This highlights that although local ENKF methods are robust to some degree of non-linearity in the dynamics or observation model, performance is still sensitive to strong departures from Gaussianity.

The effect of the non-Gaussianity induced by the non-linear observation opera-

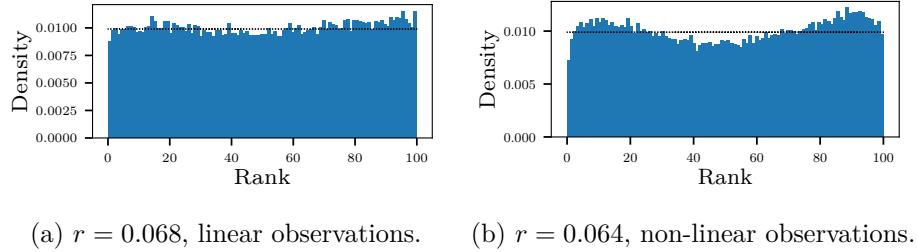


Fig 9: Rank histograms for single local ETKF runs on KS SSMS.

tor can also be seen by comparing rank histograms (i.e. the ranks of the true state values within the ensemble across all time and spatial indices) for single runs of the local ETKF on the linearly and non-linearly observed KS SSMS, as shown in Fig. 9a and Fig. 9b respectively. The localisation radius r was set to the value found in the grid searches to give the lowest mean estimate RMSE. For a well calibrated ensemble the rank histograms should be close to uniform (indicated by the dashed black line on the plots). While for the linearly observed case the minor departures from uniformity can be plausibly attributed to sampling noise, the histogram for the non-linearly observed case has a clear ‘double-humped’ non-uniform shape, with this suggesting the ensemble estimates of the filter distributions have greater kurtosis than the true filtering distributions.

For the SLETPF runs we use the same method to construct the Pous as described in the preceding section for the ST model experiments. We again considered Pous with $B \in \{32, 64, 128, 265, 512\}$ number of patches and smoothing kernel widths of $w \in \{512^{-1}, 256^{-1}, 128^{-1}, 64^{-1}\}$. For each (B, w) pair we tested all localisation radii $r \in \{0.001, 0.002, \dots, 0.030\}$ where $\text{median}(n_{1:B})$ was in the range [1, 5] for the linearly observed KS SSM and in the range [2, 6] for the non-linearly observed KS SSM. For each (B, w, r) parameter triple tested, we performed five independent filtering runs, with the median values recorded for the three metrics shown by the coloured curves in Fig. 10 for the linearly observed SSM and in Fig. 11 for the non-linearly observed SSM, along with the best values achieved by local ETKF on each metric by the dashed horizontal lines. The plots in Figs. 10 and 11 have the same format as Figs. 6 and 7 for the ST model experiments.

From the linearly observed KS SSM results in Fig. 10 we see that the SLETPF was outperformed across all parameter settings and metrics by the best local ETKF results. This reinforces the point that local ENKF methods are a strongly performant approach and can often be the best choice even in models with non-linear dynamics, where the Gaussianity assumptions are not valid, due to their robust performance when using small ensemble sizes. A further advantage of local ENKF methods over local PFs is that they naturally maintain smoothness properties of the state field particles as evidenced by the low smoothness coefficient errors achieved by the local ETKF across all model configurations. Local PF type approaches such as the SLETPF algorithm proposed here should generally therefore be considered as a fallback solution for cases where local ENKF methods are known, or at least suspected, to give poor accuracy.

Considering the performance of the SLETPF on the linearly observed KS SSM for different (B, w) parameter settings we see similar trends as observed for the ST

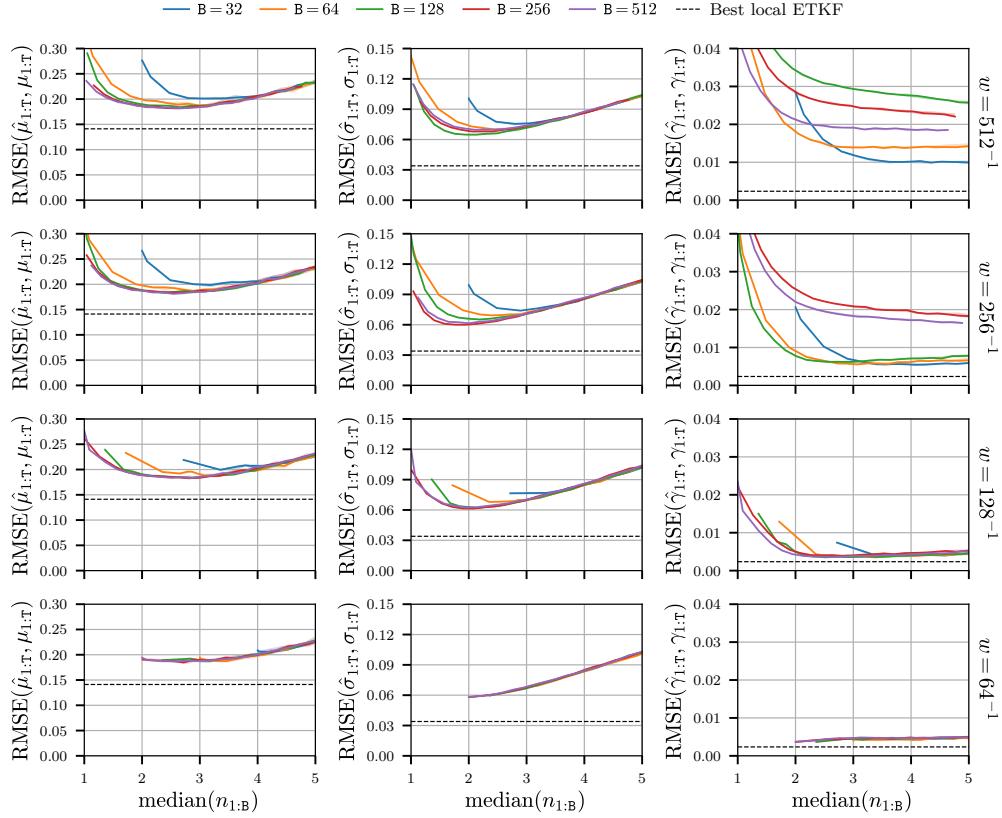


Fig 10: Comparison of accuracy of SLETPF estimates on linearly observed KS SSM.

model experiments though with some difference in the details. The differences in performances on the mean and standard deviation RMSE metrics for POU with different numbers of patches B for a fixed smoothing kernel width w show less variation than seen in the ST model experiments. Even for the hard POU case without smoothing ($w = 512^{-1}$, top-row of Fig. 10), only the runs with a POU with $B = 32$ patches show a significant drop in mean and standard deviation estimate accuracies across most $\text{median}(n_{1:B})$ values, and for the $B = 32$ case the relative drops in accuracies are still quite minor. The most obvious effect of increasing the kernel width w in this model is therefore in the improved accuracy of the smoothness coefficient estimates for larger w values. This suggests that in the KS model, although using a smoother POU does reduce the introduction of artificial discontinuities into the state field particles, these discontinuities have less of a negative effect on filtering performance than for the ST model, perhaps due to a stronger diffusive smoothing element to the model dynamics.

The results for the non-linearly observed KS SSM in Fig. 11 show similar relative performances for the different SLETPF configurations as for the linearly observed case, with a general increase in the absolute RMSE values across the board. The corresponding increase in the RMSE values for optimal tunings of the local ETKF are however significantly larger, meaning that for this model the SLETPF approaches show a minor improvement in the accuracy of the mean estimates compared to the local ETKF across virtually all configurations and performs

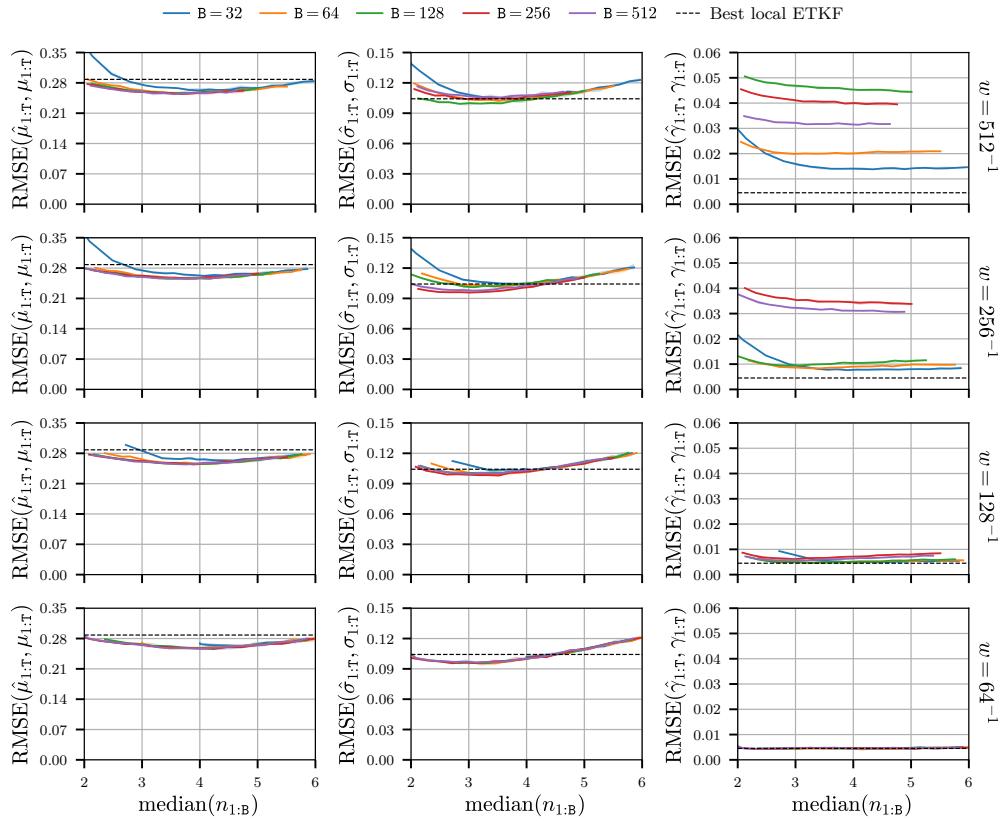


Fig 11: Comparison of accuracy of SLETPF estimates on non-linearly obs. KS SSM.

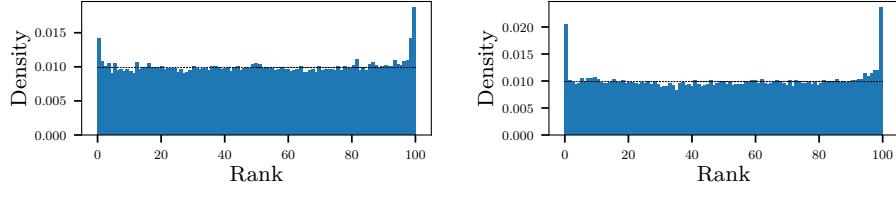
(a) $B = 512, w = 512^{-1}, r = 0.040$. (b) $B = 64, w = 128^{-1}, r = 0.022$.

Fig 12: Rank histograms for single SLETPF runs on non-linearly observed KS SSM.

comparably in terms of the accuracy of the standard deviations estimates, having slightly better performance for some configurations and slightly poorer for others. Again the smoothness of the POU used does not seem to have a strong effect on performance in terms of the mean and standard deviation estimates here, with the main change as the smoothing kernel width w is increased the improved accuracy of the smoothness coefficient estimates corresponding to improved reproduction of the smoothness of the fields under the true filtering distributions.

As for the local ETKF ensemble estimates of the KS SSM filtering distributions, we can also use rank histograms for the SLETPF ensembles as an alternative check of the calibration of the filtering distribution estimates. The rank histogram for an ensemble generated for the non-linearly observed KS SSM by a SLETPF with

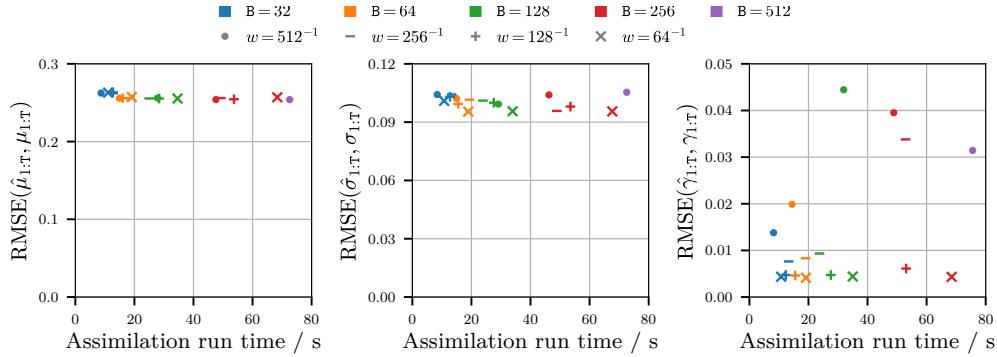


Fig 13: Accuracy versus run time for SLETPF in non-linearly observed KS SSM.

a POU with $B = 512$ patches and kernel width $w = 512^{-1}$ (i.e. corresponding to the per-node local ETPF) is shown in Fig. 12a, and for an ensemble generated for the non-linearly observed KS SSM by a SLETPF with a POU with $B = 64$ patches and kernel width $w = 128^{-1}$ in Fig. 12b. In both cases the localisation radius r was set to the value from the grid search giving the lowest mean estimate RMSE. Compared to the corresponding rank histogram for the local ETKF in Fig. 9b, the histograms for both SLETPF configurations are much closer to uniform. The peaks at the extreme ranks in both histograms are characteristic of the ensembles underestimating the dispersion of the filtering distribution in the tails, with this discrepancy appearing to be stronger in the SLETPF using fewer patches here.

As in Fig. 8 for the transformed ST model runs, it is instructive to also compare the relative computational cost of the different SLETPF configurations versus their performance on the three filtering accuracy metrics. Fig. 13 shows the time taken to perform the assimilation updates (horizontal axes) versus the value recorded for each of the three RMSE metrics (vertical axes), for each of the (B, w) POU configurations. The markers show the median values across the five runs for the localisation radius r which achieved the minimum value for that particular metric for the (B, w) values in question. Due to the decreased drop-off in filtering accuracy for POUs with fewer number of patches B compared to ST models, here we see we are able to achieve even larger improvements in computational efficiency compared to the local ETPF scheme of Cheng and Reich (2015) (corresponding to $B = 512, w = 51^{-1}, \bullet$) while retaining the same filtering accuracy. In particular the SLETPFs with $B = 64$ patches in the POU ($\bullet + - \times$) are able to achieve the same mean estimate accuracy, a slight improvement in the accuracy of the standard deviation estimates, and a substantial improvement in the accuracy of the smoothness coefficient estimates, while having an assimilation time that is around a quarter of the SLETPF which computes separate OT transport maps for each mesh node ($B = 512$). Further in this case the filtering accuracy is largely unaffected by the choice of smoothing kernel width w , other than an improvement in the smoothness coefficient estimates for larger w values. At the cost of a slight increase in all three RMSEs, the SLETPFs with POUs with $B = 32$ patches give a further approximate factor two decrease in assimilation time, leading to around a eight times decrease in assimilation time compared to the per-node local ETPF.

6. DISCUSSION

In this article we have proposed a new scheme for constructing local particle filters for state inference in SPDE models of spatially-extended dynamical systems. The local ETPF ([Cheng and Reich, 2015](#)) although having the desirable property of improved robustness to non-Gaussianity in the filtering distributions compared to local ENKF approaches has two key shortcomings: (i) the state fields produced by the assimilation step fail to maintain the smoothness properties of the predictive ensemble members, potentially leading to numerical instabilities when used to filter SPDE models and (ii) as an OT problem must be solved for every node in the spatial mesh, the assimilation updates can be costly for dense meshes.

Our approach to solving both issues is to softly partition the spatial domain using a *partition of unity*: a finite set of non-negative bump functions which tile the domain and sum to unity at all points. By computing an OT map for the patch of the spatial domain associated with each bump function and then using the bump functions to smoothly interpolate these maps across the domain, we are able to smoothly combine different regions of the predictive ensemble particles.

As well as allowing the smoothness of the spatial fields to be maintained during the assimilation step, the proposed approach reduces the $\tilde{\mathcal{O}}(MP^3)$ cost of the per-node local ETPF assimilation updates to $\tilde{\mathcal{O}}(BP^3 + |\mathcal{M}|P^2 + MP)$. If we increase the mesh resolution by using a larger number of nodes M , while keeping the number of patches B and number of subsampled nodes $|\mathcal{M}|$ fixed, the computational cost of the assimilation update only need to scale at rate $\tilde{\mathcal{O}}(MP)$ with M , which could be considered as the lower bound for an update to P particles of $\tilde{\mathcal{O}}(M)$ dimension.

We demonstrated in the numerical experiments that the resulting scheme is able to produce, at often significantly reduced computational cost, ensemble estimates of the filtering distributions for state space models with equivalent accuracy and improved smoothness compared to the local ETPF of [Cheng and Reich \(2015\)](#). Although the experiments were restricted to models on one-dimensional spatial domains, in most applications of interest the spatial domain will be two or three-dimensional. Our proposed scheme naturally carries over to this setting and as the mesh sizes in such models will tend to be significantly higher, the potential computational savings are even larger. Further, while we concentrated here on filtering in spatial models which are observed at point locations, our scheme could be extended to models with spatially distributed observations by partitioning the spatial domain according to the geometry of the observation processes.

The localisation approach to overcoming weight degeneracy when applying PFs to spatial models considered here could also be combined with other methods for improving PF performance in high-dimensional SSMs. In particular *tempering* approaches split the usual single prediction and assimilation update per observation time into multiple updates which target a sequence of distributions bridging between the filtering distributions at adjacent observation times ([Frei and Künsch, 2013; Johansen, 2015; Beskos et al., 2017; Svensson, Schön and Lindsten, 2018; Herbst and Schorfheide, 2019](#)). Tempering could be paired with our framework to further improve its robustness to high-dimensional and strongly informative observations, with the use of multiple assimilation updates per observation time when tempering making the reduced computational cost and improved smoothness preservation of our approach particularly important.

APPENDIX A: ENSEMBLE TRANSFORM KALMAN FILTER

In this Appendix we describe the details of the ETKF assimilation update ([Bishop, Etherton and Majumdar, 2001](#)) and show how it can be expressed in the form of the LETF framework discussed in Section [2.6](#). We first introduce predictive and filtering *ensemble matrices* respectively defined as

$$\vec{X}_t = \begin{bmatrix} \vec{x}_t^1 & \vec{x}_t^2 & \cdots & \vec{x}_t^P \end{bmatrix}^\top \quad \text{and} \quad X_t = \begin{bmatrix} x_t^1 & x_t^2 & \cdots & x_t^P \end{bmatrix}^\top. \quad (\text{A.1})$$

Using the following linear operators

$$\varepsilon = \frac{1}{P} \mathbf{1}_P^\top, \quad \Delta = \frac{1}{\sqrt{P-1}} (\mathbf{I}_P - \mathbf{1}_P \varepsilon), \quad (\text{A.2})$$

the predictive and filtering ensemble means can then be compactly expressed

$$\vec{m}_t^\top = \varepsilon \vec{X}_t, \quad \text{and} \quad m_t^\top = \varepsilon X_t, \quad (\text{A.3})$$

and similarly the predictive and filtering ensemble covariances can be written

$$\vec{C}_t = (\Delta \vec{X}_t)^\top (\Delta \vec{X}_t) \quad \text{and} \quad C_t = (\Delta X_t)^\top (\Delta X_t). \quad (\text{A.4})$$

Assuming initially linear-Gaussian observations as in Eq. [\(2.12\)](#) then by substituting the expressions for the empirical covariances Eq. [\(A.4\)](#) into the Kalman filter covariance assimilation update in Eq. [\(2.13a\)](#) and applying the identity $\mathbf{I}_P - \Delta \vec{Y}_t (R_t + \vec{Y}_t^\top \Delta^2 \vec{Y}_t)^{-1} \vec{Y}_t^\top \Delta = (\mathbf{I}_P + \Delta \vec{Y}_t R_t^{-1} \vec{Y}_t^\top \Delta)^{-1}$ with $\vec{Y}_t = \vec{X}_t H_t^\top$ we have

$$(\Delta X_t)^\top (\Delta X_t) = (\Delta \vec{X}_t)^\top \left(\mathbf{I}_P + \Delta \vec{Y}_t R_t^{-1} \vec{Y}_t^\top \Delta \right)^{-1} (\Delta \vec{X}_t). \quad (\text{A.5})$$

Defining S_t as the symmetric matrix square-root of the central term in the right-hand-side of Eq. [\(A.5\)](#), i.e.

$$S_t^2 = S_t S_t = \left(\mathbf{I}_P + \Delta \vec{Y}_t R_t^{-1} \vec{Y}_t^\top \Delta \right)^{-1} \quad (\text{A.6})$$

then we can compute a family of solutions of Eq. [\(A.5\)](#) for the filtering ensemble projection ΔX_t in terms of the predictive ensemble projection $\Delta \vec{X}_t$ as

$$\Delta X_t = Q S_t \Delta \vec{X}_t \quad (\text{A.7})$$

where Q is an arbitrary $P \times P$ orthogonal matrix. For the ETKF generally $Q = \mathbf{I}_P$ is chosen, corresponding to directly transforming by the symmetric square-root.

Now considering the Kalman assimilation update for the mean in Eq. [\(2.13b\)](#), substituting the expressions for the ensemble empirical means and covariances in Eqs. [\(A.3\)](#) and [\(A.4\)](#) and using the definition of the square-root matrix S_t in Eq. [\(A.6\)](#) we have that

$$\varepsilon X_t = \varepsilon \vec{X}_t + (y_t^\top - \varepsilon \vec{Y}_t) R_t^{-1} \vec{Y}_t^\top \Delta S_t^2 \Delta \vec{X}_t. \quad (\text{A.8})$$

From the definition of Δ in Eq. [\(A.2\)](#) we have that $\mathbf{I}_P = \mathbf{1}_P \varepsilon + \sqrt{P-1} \Delta$ and so

$$X_t = \mathbf{1}_P \varepsilon X_t + \sqrt{P-1} \Delta X_t \quad (\text{A.9})$$

$$= \left(\mathbf{1}_P \varepsilon + \mathbf{1}_P (y_t^\top - \varepsilon \vec{Y}_t) R_t^{-1} \vec{Y}_t^\top \Delta S_t^2 \Delta + \sqrt{P-1} S_t \Delta \right) \vec{X}_t, \quad (\text{A.10})$$

with the matrix term in parentheses defining the coefficients $a_t^{1:P,1:P}$ of an LETF assimilation update as in Eq. (2.11).

In the above it was assumed the observation model is linear-Gaussian. In the case of a more general observation model of the form

$$G_t(x, v) = H_t(x) + v, \quad v_t \sim \mathcal{N}(0, R_t), \quad (\text{A.11})$$

where now H_t is a potentially non-linear operator, then by observing that all occurrences of H_t in Eqs. (A.6) and (A.10) are via $\vec{Y}_t = \vec{X}_t H_t^\top$, for non-linear H_t we can instead define the *predictive observation ensemble matrix* \vec{Y}_t as

$$\vec{Y}_t = [H_t(\vec{x}_t^1) \ H_t(\vec{x}_t^2) \ \cdots \ H_t(\vec{x}_t^P)]^\top. \quad (\text{A.12})$$

The ETKF formulation of a square-root ENKF has the advantage of only requiring computing cubic-cost matrix operations for matrices of size $P \times P$ (due to the conditional independence assumptions R_t is block diagonal and so the cost of computing R_t^{-1} is at worst $\mathcal{O}(LK^3)$ with in general $K \ll P$).

For all ENKF methods, the assimilation updates are only consistent with the analytic assimilation update in Eq. (2.7) as $P \rightarrow \infty$ for linear-Gaussian models. In models where the state update and observation operators are only weakly nonlinear, the filtering distribution at each time index π_t can remain ‘close’ to Gaussian and the ENKF updates will often give reasonable estimates of the filtering distribution (Evensen, 2009). For models with highly non-Gaussian filtering distributions ENKF methods will typically perform poorly however.

A local version of the ETKF algorithm was proposed in Hunt, Kostelich and Szunyogh (2007). In the global ETKF assimilation update summarised in Eq. (A.10) the linear transform coefficients depend on the current predictive state ensemble values $\vec{x}_t^{1:P}$ only via a $P \times KL$ observation ensemble matrix \vec{Y}_t . The local ETKF algorithm scales the dependence of the update coefficients at each mesh node on the columns of \vec{Y}_t via a *localisation function* $\ell_r : [0, \infty) \rightarrow [0, 1]$ satisfying the conditions in Eq. (3.9) for some localisation radius $r > 0$, such that observations at a distance more than r from the mesh node are ignored in the corresponding local assimilation update.

For each of the M mesh nodes a *localisation kernel* is then defined by applying ℓ_r to the distances between the mesh nodes and the observation locations

$$k_m^\top = [\ell_r(d(s_m, s_1^o))^{\frac{1}{2}} \ 1_K^\top, \ \cdots, \ \ell_r(d(s_m, s_L^o))^{\frac{1}{2}} \ 1_K^\top] \quad \forall m \in 1:M. \quad (\text{A.13})$$

We can then define local effective observation noise precision matrices $\tilde{R}_{t,1:M}^{-1}$

$$\tilde{R}_{t,m}^{-1} = R_t^{-1} \odot (k_m k_m^\top) \quad \forall m \in 1:M \quad (\text{A.14})$$

where \odot indicate the elementwise or Hadamard product between equal sized tensors. The local ETKF assimilation update is then

$$X_{t,m} = \left(I_P \varepsilon + I_P (y_t^\top - \varepsilon \vec{Y}_t) \tilde{R}_{t,m}^{-1} \vec{Y}_t^\top \Delta \tilde{S}_{t,m}^2 \Delta + \sqrt{P-1} \tilde{S}_{t,m} \Delta \right) \vec{X}_{t,m}, \quad (\text{A.15})$$

where the local square root matrix $\tilde{S}_{t,m}$ is defined

$$\tilde{S}_{t,m}^2 = \tilde{S}_{t,m} \tilde{S}_{t,m} = \left(I_P + \Delta \vec{Y}_t \tilde{R}_{t,m}^{-1} \vec{Y}_t^\top \Delta \right)^{-1}. \quad (\text{A.16})$$

This local assimilation update is equivalent to replacing each observation ensemble matrix term \tilde{Y}_t and observation vector term y_t in the global assimilation update in Eq. (A.10) with $\tilde{Y}_t \odot (1_P k_m^T)$ and $y_t \odot k_m$ respectively. As k_m has zero entries for all indices corresponding to observation locations more than r in distance from s_m , in practice when implementing the local ETKF assimilation update the computations can be performed with only the non-zero submatrices of $\tilde{Y}_t \odot (1_P k_m^T)$ and $y_t \odot k_m$ and corresponding submatrix of R_t^{-1} .

As separate assimilation updates need to be computed for each mesh node the computational cost of the local ETKF scales linearly with the number of mesh nodes M . The computation for each mesh node is of order $\mathcal{O}(P^3)$ due to requirement to calculate a matrix decomposition of the $P \times P$ matrix inside the parentheses on the right hand side of Eq. (A.16). On a sequential architecture the overall computation time will therefore have a $\mathcal{O}(MP^3)$ scaling. As each of the local assimilation updates can be independently computed in parallel, with a large number of parallel compute nodes the assimilation update can still be computed efficiently for models with large mesh sizes M however as shown in the numerical experiments in [Hunt, Kostelich and Szunyogh \(2007\)](#).

APPENDIX B: ALTERNATIVE PARTICLE FILTER PROPOSALS

Rather than propagating according to the forward dynamics of the generative model, it is possible to instead propose new particle values from different conditional distributions (which may depend on future observed values) and adjust the expression for the importance weights in Eq. (2.14) accordingly. Typically the resulting expression for the importance weights is given in terms of the transition density of the state updates, however as noted previously this density will often be intractable to compute. Alternative state proposals can however instead be formulated by changing the distribution the state noise variables are drawn from. If each state noise vector u_t^p is sampled from a distribution with a known density d_t^p with respect to μ_t and the predictive ensemble particles computed as in Eq. (2.9), then unnormalised importance weights for the propagated particles can be computed as

$$\tilde{w}_t^p = g_t(y_t | F_t(x_{t-1}^p, u_t^p)) d_t^p(u_t^p)^{-1} \quad \forall p \in 1:P. \quad (\text{B.1})$$

The corresponding normalised weights can then be used in the empirical filtering distribution approximation in Eq. (2.14) and resampling update in Eq. (2.15). If we restrict the state noise proposal density d_t^p to be dependent on only the previous particle x_{t-1}^p and current observation y_t in order to maintain the online nature of the algorithm, then the proposal distributions which minimise the variance of the importance weights have densities with respect to μ_t

$$d_t^p(u) = \frac{g_t(y_t | F_t(x_{t-1}^p, u))}{\int_{\mathcal{U}} g_t(y_t | F_t(x_{t-1}^p, u')) \mu_t(du')} \quad \forall p \in 1:P. \quad (\text{B.2})$$

In this case the unnormalised weights in Eq. (B.1) are independent of the state noise variables $u_t^{1:P}$. Although this ‘optimal’ proposal is more typically expressed as a conditional distribution on \tilde{x}_t^p given x_{t-1}^p this alternative formulation is equivalent. In general it will not be possible to generate samples from the optimal proposal, however it may be possible to for example find a tractable approximation to use as a proxy.

In cases where the optimal proposal is tractable or can be well approximated, the resulting PF algorithm can significantly outperform the basic bootstrap PF in terms of the ensemble size required for a given accuracy in the filtering distribution estimates.

APPENDIX C: VISUALISATION OF SMOOTH LOCAL LETF ASSIMILATION UPDATE

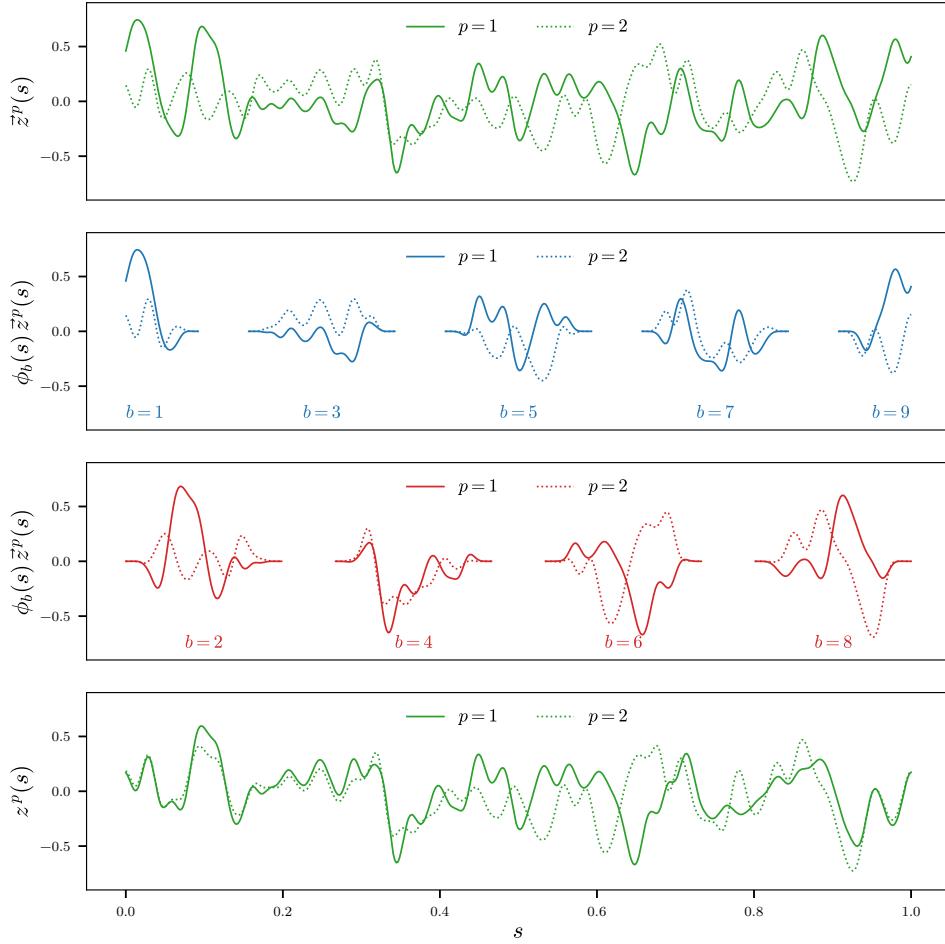


Fig C.1: Example of applying smooth local LETF assimilation update in Eq. (4.7) on a one-dimensional spatial domain $\mathcal{S} = [0, 1]$ using the POU from Fig. 5 with $P = 2$ particles.

Consider a spatial domain which is the same unit interval $\mathcal{S} = [0, 1]$ as used in Fig. 5 and a POU chosen as the smooth bump functions $\phi_{1:9}$ shown there. The top panel in Fig. C.1 shows two smooth predictive distribution particle realisations $\tilde{z}^{1:2}$. The central two³ panels show the products $\phi_b(s)\tilde{z}^p(s) \forall b \in 1:9, p \in 1:2$, which can also be seen to be smooth functions of the spatial coordinate s and compactly supported on the patches $\hat{\mathcal{S}}_{1:9}$. The bottom panel shows the filtering distribution particle realisations $z^{1:2}$ computed using the assimilation update in Eq. (4.7) for a randomly generated set of coefficients $\hat{a}_{1:9}^{1:2,1:2}$ satisfying the conditions in Eq. (4.5), with these post-assimilation fields maintaining the smoothness of the predictive fields.

³The separation of products with odd and even indexed bump functions on to separate panels in Fig. C.1 is simply for visual clarity.

APPENDIX D: PARTITIONING THE SPATIAL DOMAIN

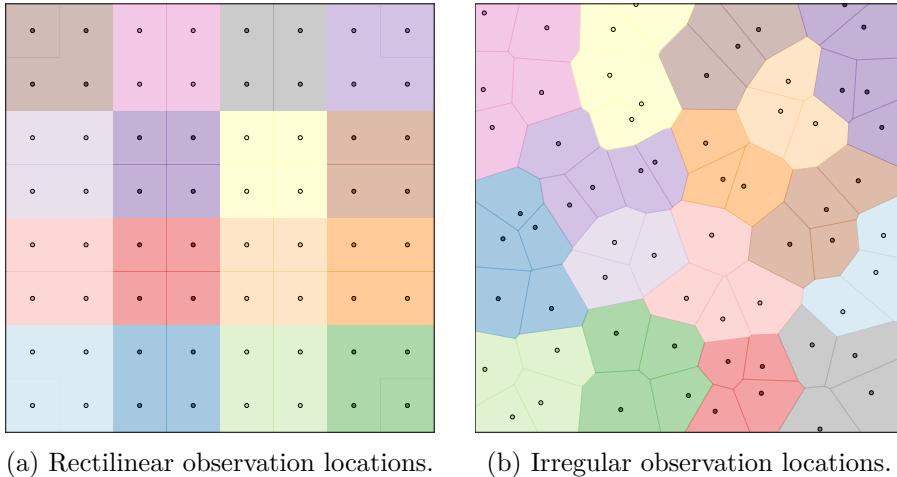


Fig D.1: Examples of partitioning a space based on observation locations for a two-dimensional spatial domain. Panel (a) shows a partition (indicated by coloured regions) for observations located on an equispaced rectilinear grid (shown by circular markers). Panel (b) shows a partition for an irregularly located set of observations, with the observation locations initially clustered (indicated by colours of markers) before partitioning based on the Voronoi cells associated with each cluster of observation locations (cells shown by bordered polygonal regions).

In order to control the number of observations used to compute each local weight in the SLETPF scheme, we recommend choosing the partition of the spatial domain used to define the POU such that each patch contains roughly the same number of observations. For observations located on a rectilinear grid this can easily be achieved by partitioning the space in to rectilinear blocks aligned with the observation grid and each containing the same number of observations (an example is shown in Fig. D.1a). For irregularly spaced observations, one option is to first group the observation locations in to similarly sized clusters using for example a k -means algorithm. The spatial domain can then be partitioned using a Voronoi diagram generated from the observation locations, with all the cells corresponding to observations in a single cluster then merged to form a single contiguous region. This leads to a partition of the spatial domain into a set of regions which each contain a roughly number of observations and such that the numbers of additional observations close to the region boundaries are minimised. A example of applying this scheme to a set of irregularly located observation points is shown in Fig. D.1b. In both the rectilinear and irregular spacing cases, a soft POU can then be generated from the resulting partition by convolving with a mollifier function as described in Section 4.1.

APPENDIX E: TRANSFORMED STATE-SPACE MODELS

One of our primary motivations for considering PF-based methods was the claim that they are more robust to non-Gaussianity in the filtering distributions compared to ENKF methods. While this can be shown to be the case in the large ensemble limit for non-localised PF algorithms (including the ETPF) compared to ENKF methods, it does not necessarily follow that, when using small ensemble sizes, a local ETPF would be expected to outperform a local ENKF in models with non-Gaussian filtering distributions. Further even if there is a benefit to using the local ETPF compared to the local ENKF, this does not necessarily carry over to our proposed smooth and scalable local ETPF scheme.

Therefore to assess the affect on the relative performance of the local ensemble filters methods being considered of non-Gaussianity in the filtering distributions while controlling as far as possible other factors which might affect performance, we use a simple scheme to map a tractable linear-Gaussian SSM to a transformed SSM with non-Gaussian filtering distributions. In particular let $T : \mathcal{X} \rightarrow \mathcal{X}$ be a diffeomorphism on the state space, with T^{-1} denoting its inverse, which we assume we can also compute. If we define $x'_t = T(x_t) \forall t \in 1:T$ then the conditional distribution on x'_t given observations $y_{1:t} = y_{1:t}$ will be $T_\sharp \pi_t$ for any time index $t \in 1:T$, i.e. the *push-forward* of the filtering distribution π_t under the map T . If T is non-linear then if π_t is Gaussian $T_\sharp \pi_t$ will in general be non-Gaussian.

Importantly for our purposes we can construct a SSM acting directly on the transformed states $x'_{1:T}$. In particular for a *base* SSM with state update and observation operators $F_{1:T}$ and $G_{1:T}$, we can define a *T-transformed* SSM with state update and observation operators $F'_{1:T}$ and $G'_{1:T}$ given by

$$x'_1 = F'_1(u_1) = T \circ F_1(u_1), \quad u_1 \sim \mu_1, \quad (\text{E.1})$$

$$x'_t = F'_t(x'_{t-1}, u_t) = T \circ F_t(T^{-1}(x'_{t-1}), u_t), \quad u_t \sim \mu_t \quad \forall t \in 2:T, \quad (\text{E.2})$$

$$y_t = G'_t(x'_t, v_t) = G_t(T^{-1}(x'_t), v_t), \quad v_t \sim \nu_t \quad \forall t \in 1:T \quad (\text{E.3})$$

and with observation densities $g'_{1:T}$ defined by

$$g'_t(y_t | x'_t) = g_t(y_t | T^{-1}(x'_t)) \quad \forall t \in 1:T. \quad (\text{E.4})$$

We can therefore run ensemble filter algorithms on the *T*-transformed SSM to directly compute ensemble estimates of the transformed filtering distributions $\pi'_{1:T}$ with by construction $\pi'_t = T_\sharp \pi_t \forall t \in 1:T$. If the base SSM is linear-Gaussian and so a KF can be used to exactly compute the Gaussian filtering distributions $\pi_{1:T}$, we can compute accurate unbiased Monte Carlo estimates of expectations under the transformed filtering distributions $\pi'_{1:T}$ as we can generate N independent samples from each π'_t by generating N independent samples from the Gaussian filtering distribution π_t and pushing each of the samples through the map T .

This scheme therefore provides a method for constructing a non-Gaussian SSM for which we can easily compute accurate Monte Carlo estimates of the true filtering distribution means $\mu_{1:T}$, standard deviations $\sigma_{1:T}$ and smoothness coefficients $\gamma_{1:T}$ as defined in Eqs. (5.1) and (5.5) and so evaluate the RMSE accuracy metrics described in the preceding section for ensemble estimates of the filtering distributions. By using a large number of independent samples N in the Monte Carlo estimates we can ensure the $\mathcal{O}(N^{-\frac{1}{2}})$ Monte Carlo error is negligible compared to the error in the ensemble estimates.

APPENDIX F: MODEL DETAILS

F.1 Stochastic turbulence model

Number of mesh nodes	$M = 512$
Number of observation times	$T = 200$
Number of observation locations	$L = 64$
Time step	$\delta = 2.5$
Diffusion coefficient	$\theta_1 = 4 \times 10^{-5}$
Advection coefficient	$\theta_2 = 0.1$
Damping coefficient	$\theta_3 = 0.1$
Transformation scale factor	$\theta_4 = 5$
State noise kernel length scale	$\vartheta = 4 \times 10^{-3}$
State noise kernel amplitude	$\alpha = 0.1$
Observation noise standard deviation	$\varsigma = 0.5$

TABLE F.1
Stochastic turbulence model parameter settings

We define a regular mesh of nodes $s_{1:M}$ and basis functions $\beta_{1:M}$

$$s_m = \frac{m-1}{M} \quad \text{and} \quad \beta_m(s) = \frac{\text{sinc}(2\pi M(s - s_m)) \cos(\pi(s - s_m))}{\text{sinc}(\pi(s - s_m))} \quad \forall m \in 1:M \quad (\text{F.1})$$

with the space-time varying processes ζ and η and kernel function κ then being defined respectively in terms of the finite set of time-varying processes $\chi_{1:M}$ and $v_{1:M}$ and coefficients $\lambda_{1:M}$ as

$$\zeta(s, \tau) = \sum_{m \in 1:M} \chi_m(\tau) \beta_m(s), \quad (\text{F.2})$$

$$\eta(s, \tau) = \sum_{m \in 1:M} v_m(\tau) \beta_m(s), \quad (\text{F.3})$$

$$\text{and} \quad \kappa(s) = \sum_{m \in 1:M} \lambda_m \beta_m(s). \quad (\text{F.4})$$

The basis functions $\beta_{1:M}$ and nodes $s_{1:M}$ satisfy Eq. (3.2) such that $\chi_m(\tau)$, $v_m(\tau)$ and λ_m correspond to the values of respectively $\zeta(s_m, \tau)$, $\eta(s_m, \tau)$ and $\kappa(s_m)$ for any mesh node s_m . We define $\tilde{\chi}_{0:K}(\tau) = \text{DFT}(\chi_{1:M}(\tau))$, $\tilde{v}_{0:K}(\tau) = \text{DFT}(v_{1:M}(\tau))$ and $\tilde{\lambda}_{0:K} = \text{DFT}(\lambda_{1:M})$ with $K = \lfloor \frac{M}{2} \rfloor$ and DFT indicating the discrete Fourier transform, with the Fourier coefficient \tilde{x}_k for a real sequence $x_{1:M}$ being computed as

$$\tilde{x}_k = \text{DFT}_k(x_{1:M}) = \frac{1}{M} \sum_{m \in 1:M} x_m \exp\left(-\frac{i2\pi km}{M}\right) \in \begin{cases} \mathbb{R} & \text{if } k \in \{0, \frac{M}{2}\}, \\ \mathbb{C} & \text{if } k \in 1:\lceil \frac{M}{2} \rceil - 1. \end{cases} \quad (\text{F.5})$$

Then we have the following equivalent spectral expansions for ζ , η and κ

$$\zeta(s, \tau) = \sum_{k \in -K:K} \alpha_k \tilde{\chi}_k(\tau) \exp(i\omega_k s), \quad (\text{F.6})$$

$$\eta(s, \tau) = \sum_{k \in -K:K} \alpha_k \tilde{v}_k(\tau) \exp(i\omega_k s), \quad (\text{F.7})$$

$$\kappa(s) = \sum_{k \in -K:K} \alpha_k \tilde{\lambda}_k \exp(i\omega_k s), \quad (\text{F.8})$$

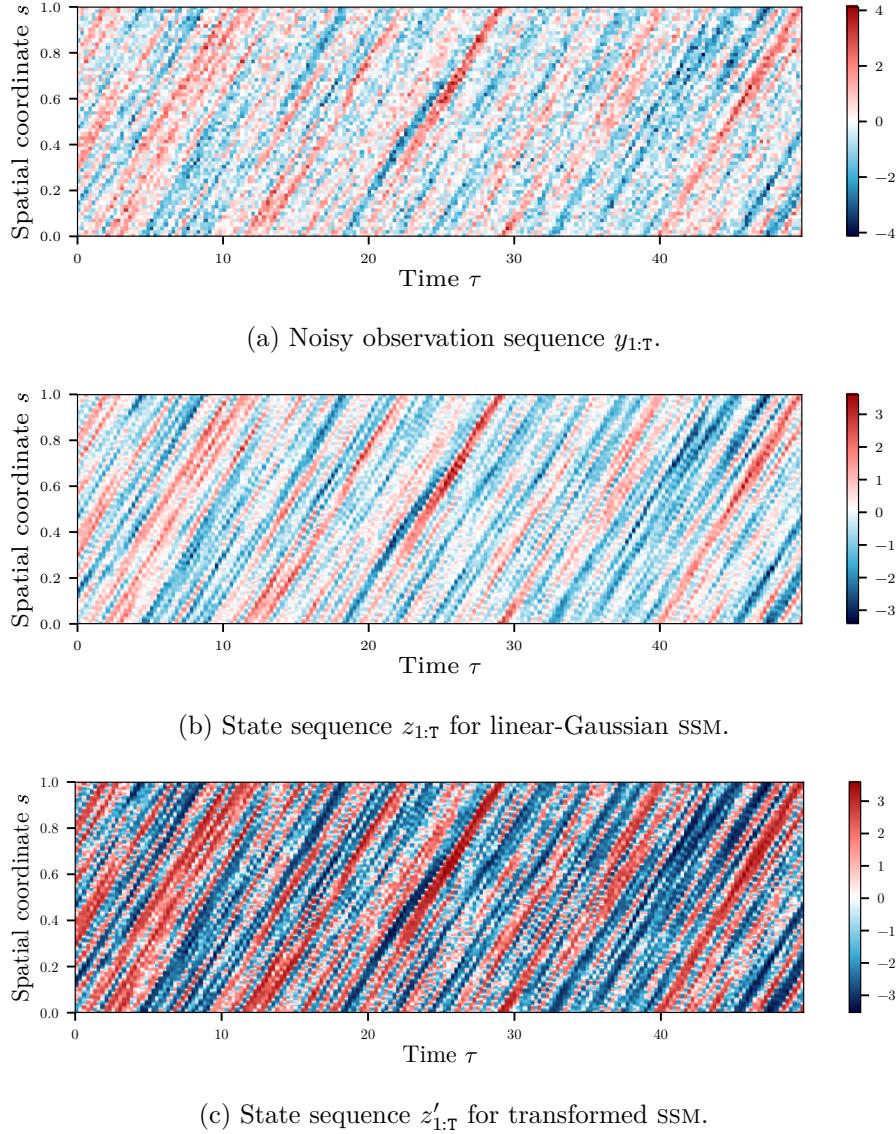


Fig F.1: Simulated sequences used in experiments for ST SSMS.

with the convention that negative indices to the Fourier coefficients indicate complex conjugation, e.g. $\tilde{\lambda}_{-k} = \tilde{\lambda}_k^*$, and $\alpha_{-\mathbf{K}:\mathbf{K}}$ and $\omega_{-\mathbf{K}:\mathbf{K}}$ are defined as

$$\alpha_k = \begin{cases} \frac{1}{2} & \text{if } k = 0, \\ 1 & \text{if } |k| \in 1 : \lceil \frac{M}{2} \rceil - 1, \quad \text{and} \quad \omega_k = 2\pi k \quad \forall k \in -\mathbf{K}:\mathbf{K}, \\ \frac{1}{4} & \text{if } |k| = \frac{M}{2}, \end{cases} \quad (\text{F.9})$$

Using Eq. (F.6) we then have that spatial derivatives of ζ can be computed as

$$\partial_s^n \zeta(s, \tau) = \sum_{k \in -\mathbf{K}:\mathbf{K}} \alpha_k (i\omega_k)^n \bar{\chi}_k(\tau) \exp(i\omega_k s) \quad \forall n \in \mathbb{N}. \quad (\text{F.10})$$

Substituting the expansions in Eqs. (F.6), (F.7) and (F.10) for the processes and

spatial derivatives into Eq. (5.8) and using the convolution theorem gives

$$\sum_{k \in -K:K} \alpha_k \left(d\tilde{\chi}_k - (-\theta_1 \omega_k^2 + i\theta_2 \omega_k - \theta_3) \tilde{\chi}_k d\tau - \tilde{\lambda}_k d\tilde{v}_k \right) \exp(i\omega_k s) = 0. \quad (\text{F.11})$$

Integrating both sides over \mathcal{S} against a suitable orthogonal set of test functions

$$h_j(s) = \exp(-i\omega_j s) \quad \forall j \in 0: (\lceil \frac{M}{2} \rceil - 1) \quad \text{and} \quad h_{\frac{M}{2}}(s) = \cos(M\pi s) \quad \text{if } M \text{ is even}, \quad (\text{F.12})$$

we arrive at the following system of SDES

$$d\tilde{\chi}_k(\tau) = (-\theta_1 \omega_k^2 + i\theta_2 \omega_k - \theta_3) \tilde{\chi}_k(\tau) d\tau + \tilde{\lambda}_k d\tilde{v}_k(\tau) \quad \forall k \in 0: (\lceil \frac{M}{2} \rceil - 1), \quad (\text{F.13})$$

$$d\tilde{\chi}_{\frac{M}{2}}(\tau) = (-\theta_1 \omega_k^2 - \theta_3) \tilde{\chi}_{\frac{M}{2}}(\tau) d\tau + \tilde{\lambda}_{\frac{M}{2}} d\tilde{v}_{\frac{M}{2}}(\tau) \quad \text{if } M \text{ is even}. \quad (\text{F.14})$$

Assuming that the noise Fourier coefficients $\tilde{v}_{0:K}$ are independent Wiener processes, real-valued for the zero- and Nyquist-frequency coefficients (\tilde{v}_0 and $\tilde{v}_{\frac{M}{2}}$) and complex-valued for the remaining coefficients, then the transition distributions for this system have analytic solutions

$$\begin{aligned} \tilde{\chi}_k(\tau) | \tilde{\chi}_k(0) &\sim \mathcal{N} \left(\exp(\xi_k \tau) \tilde{\chi}_k(0), \frac{\tilde{\lambda}_k^2}{2\psi_k} (1 - \exp(-2\psi_k \tau)) \right), \\ \text{with } \psi_k = \theta_1 \omega_k^2 + \theta_3 \text{ and } \xi_k &= \begin{cases} i\theta_2 \omega_k - \psi_k & \text{if } k \neq \frac{M}{2} \\ -\psi_k & \text{if } k = \frac{M}{2} \end{cases}, \end{aligned} \quad k \in 0:K. \quad (\text{F.15})$$

where we have overloaded the notation for a Gaussian distribution \mathcal{N} to extend to complex-valued variables with the convention that for a complex-valued random variable $z \in \mathbb{C}$, complex mean parameter $\mu \in \mathbb{C}$ and real variance $\sigma^2 \in \mathbb{R}_{>0}$, that

$$\begin{aligned} z &\sim \mathcal{N}(\mu, \sigma^2) \implies \\ \Re(z) &\sim \mathcal{N}\left(\Re(\mu), \frac{\sigma^2}{2}\right), \quad \Im(z) \sim \mathcal{N}\left(\Im(\mu), \frac{\sigma^2}{2}\right) \quad \text{and} \quad \Re(z) \perp \Im(z). \end{aligned} \quad (\text{F.16})$$

The Fourier coefficients $\tilde{\chi}_{0:K}$ then also have Gaussian stationary distributions

$$\tilde{\chi}_k(\infty) \sim \mathcal{N}\left(0, \frac{\tilde{\lambda}_k^2}{2\psi_k}\right) \quad \forall k \in 0:K. \quad (\text{F.17})$$

We assume the system is observed at T time points with $\tau_t = (t - 1)\delta \quad \forall t \in 1:T$ and that the Fourier coefficients of the initial state at time $\tau_1 = 0$ are generated from the stationary distributions in Eq. (F.17). Identifying

$$z_t(s) = \zeta(s, \tau_t) \quad \text{and} \quad x_{t,1:M} = \chi_{1:M}(\tau_t) \quad \forall t \in 1:T \quad (\text{F.18})$$

we have that the state update operators can be written

$$x_{1,1:M} = \text{DFT}^{-1}(a_{0:K} \odot u_{1,0:K}), \quad (\text{F.19})$$

$$x_{t,1:M} = \text{DFT}^{-1}(b_{0:K} \odot \text{DFT}(x_{t-1,1:M}) + c_{0:K} \odot u_{t,0:K}) \quad \forall t \in 2:T, \quad (\text{F.20})$$

where $a_{0:K}$, $b_{0:K}$ and $c_{0:K}$ are length $K + 1$ vectors with

$$a_k = \frac{\tilde{\lambda}_k}{\sqrt{2\psi_k}}, \quad b_k = \exp(\xi_k \delta), \quad c_k = a_k \sqrt{1 - \exp(-2\psi_k \delta)} \quad \forall k \in 0:K, \quad (\text{F.21})$$

Number of mesh nodes	$M = 512$
Number of observation times	$T = 200$
Number of observation locations	$L = 64$
Number of integrator steps between observations	$S = 10$
Integrator time step	$\delta = 0.25$
Length scale parameter	$\theta_1 = 32\pi$
Damping coefficient	$\theta_2 = \frac{1}{6}$
State noise kernel length scale	$\theta_3 = \theta_1^{-1}$
State noise kernel amplitude	$\theta_4 = \theta_1^{-\frac{1}{2}}$
Observation noise standard deviation	$\varsigma = 0.5$

TABLE F.2
Kuramoto-Sivashinsky model parameter settings

and the state noise variables $u_{1:T,0:K}$ are real-valued for the zero- and Nyquist-frequency components and complex otherwise and have Gaussian distributions

$$u_{t,k} \in \begin{cases} \mathbb{R} & \text{if } k \in \{0, \frac{M}{2}\}, \\ \mathbb{C} & \text{if } k \in 1 : \lceil \frac{M}{2} \rceil - 1, \end{cases} \quad u_{t,k} \sim \mathcal{N}(0, 1) \quad \forall t \in 1:T, k \in 0:K. \quad (\text{F.22})$$

The system is observed at L equispaced mesh nodes with $s_l^o = s_{\frac{M}{L}(l-\frac{1}{2})}$ $\forall l \in 1:L$ and a simple linear-Gaussian observation model assumed

$$y_{t,l} = z_t(s_l^o) + v_{t,l} = x_{\frac{M}{L}(l-\frac{1}{2})} + v_{t,l}, \quad v_{t,l} \sim \mathcal{N}(0, \varsigma^2) \quad \forall t \in 1:T, l \in 1:L. \quad (\text{F.23})$$

The state noise kernel Fourier coefficients $\tilde{\lambda}_{0:K}$ are chosen to represent a squared-exponential kernel with length-scale parameter ϑ and amplitude parameter α

$$\tilde{\lambda}_k = \alpha \exp(-\omega_k^2 \vartheta^2) \quad \forall k \in 0:K. \quad (\text{F.24})$$

F.2 Kuramoto–Sivashinsky model

We use the same spectral approach as in the ST model to define the basis function expansions of the processes ζ and η and kernel κ in terms of coefficients $\xi_{1:M}$, $v_{1:M}$ and $\lambda_{1:M}$ (see Eqs. (F.6) to (F.7)). The non-linear ζ^2 term in the drift component of the SPDE cannot be exactly expressed as a linear combination of the basis function $\beta_{1:M}$, and so we cannot directly form a system of SDEs to solve as in the ST model. We make the approximation that

$$\zeta(s, \tau)^2 = \sum_{m \in 1:M} \sum_{n \in 1:M} \chi_m(\tau) \chi_n(\tau) \beta_m(s) \beta_n(s) \approx \sum_{m \in 1:M} \chi_m(\tau)^2 \beta_m(s). \quad (\text{F.25})$$

At the mesh nodes $s_{1:M}$ this gives the correct values but gives a different interpolation at points between the nodes; for dense meshes however the error introduced is small. Using this approximation the following system of SDEs can be derived in the Fourier coefficients $\tilde{\xi}_{0:K}$, $\tilde{v}_{0:K}$ and $\tilde{\lambda}_{0:K}$

$$d\tilde{\chi}_k(\tau) = \left(\left(\frac{\omega_k^2}{\theta_1^2} - \frac{\omega_k^4}{\theta_1^4} - \theta_2 \right) \tilde{\chi}_k(\tau) + N_k(\tilde{\chi}_{0:K}) \right) d\tau + \tilde{\lambda}_k d\tilde{v}_k(\tau) \quad \forall k \in 0:K \quad (\text{F.26})$$

with the noise Fourier coefficients $\tilde{v}_{1:K}$ again assumed to be (complex-valued) Wiener processes and the non-linear N_k terms in the drift defined by

$$N_k(\tilde{\chi}_{0:K}) = \begin{cases} \frac{i\omega_k}{2\theta_1} \text{DFT}_k(\text{DFT}^{-1}(\tilde{\chi}_{0:K}(\tau))^2) & \text{if } k \in 0 : (\lceil \frac{M}{2} \rceil - 1), \\ 0 & \text{if } k = \frac{M}{2}. \end{cases} \quad (\text{F.27})$$

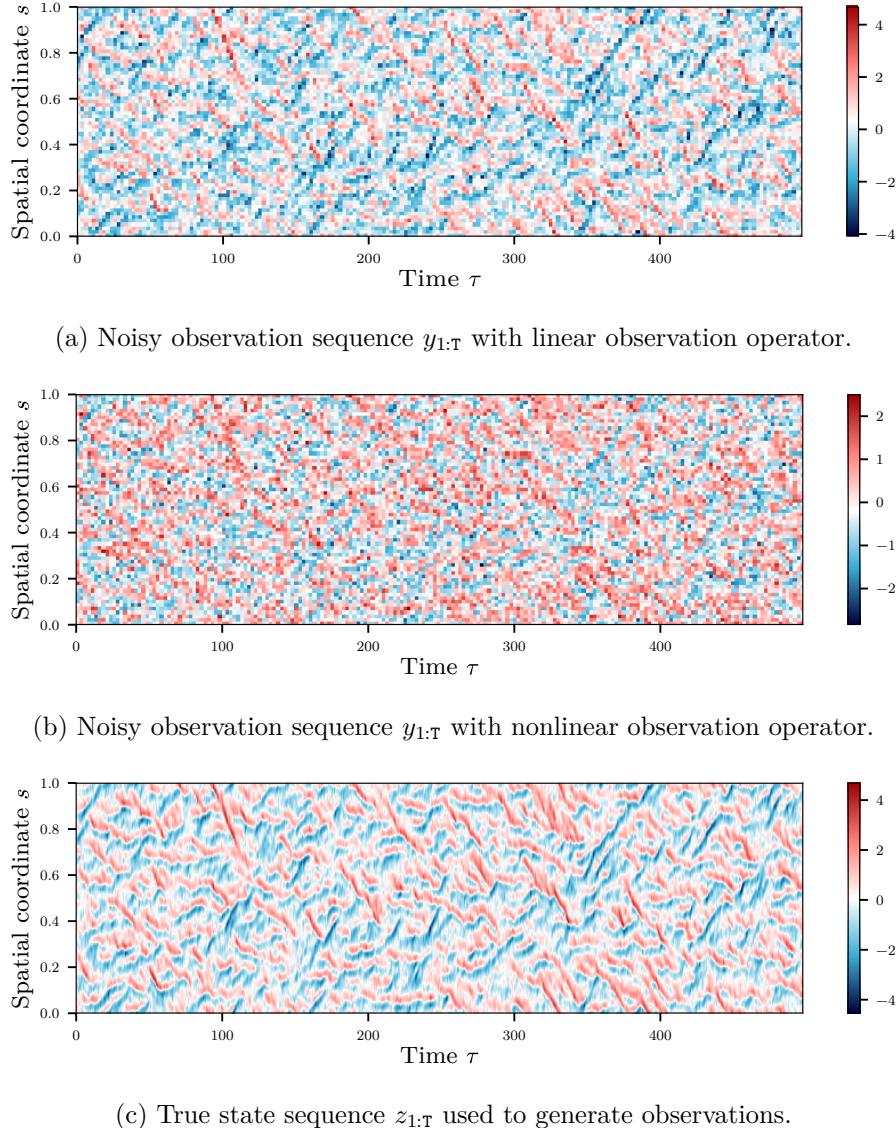


Fig F.2: Simulated sequences used in experiments with stochastic KS SSMS.

The state noise kernel Fourier coefficients $\tilde{\lambda}_{0:K}$ are as in the ST model chosen to represent a squared-exponential kernel as defined in Eq. (F.24).

Due to the non-linear terms, the system of SDEs in Eq. (F.26) does not have an analytic solution. Therefore we numerically integrate the system using a heuristic combination of a exponential-time differencing fourth-order Runge-Kutta scheme (Cox and Matthews, 2002) to time step forward according to the drift term and a Euler-Maruyama discretisation to account for the diffusion term. To reduce the time discretisation error we use S integrator steps with time step δ between each of the T observation times $\tau_t = (t - 1)\mathbf{S}\delta \forall t \in 1:T$. The state transition operator F_t then correspond to the map from a previous state vector x_{t-1} and state noise variable u_t (consisting of the concatenation of S simulated Wiener process increments) to the state vector x_t by performing S integrator steps. The

state transition operators are non-linear and the density of the corresponding state transition distribution does not have a closed form solution.

For the observation operators we considered two cases - a linear-Gaussian observation model and a non-linear observation operator. Although due to the non-linear state transition operators the filtering distributions are non-Gaussian irrespective of the observation operator used, in practice we found the local ENKF was able to generate accurate ensemble estimates of the filtering distributions when using a simple linear-Gaussian observation model, suggesting the filtering distributions remain close to Gaussian despite the non-linear state dynamics. As our focus is on inference in SSMS for which existing local ENKF approaches perform poorly in, we also considered an alternative model configuration in which a non-linear function of the model state is noisily observed.

In both the linear and non-linear cases system is assumed to be observed at L equispaced mesh nodes with $s_l^o = s_{\frac{L}{2}(l-\frac{1}{2})} \forall l \in 1:L$. For the linear case the observation model is assumed to be equivalent to that assumed for the ST model,

$$y_{t,l} = z_t(s_l^o) + v_{t,l} = x_{\frac{L}{2}(l-\frac{1}{2})} + v_{t,l}, \quad v_{t,l} \sim \mathcal{N}(0, \varsigma^2) \quad \forall t \in 1:T, l \in 1:L. \quad (\text{F.28})$$

The non-linear case is directly analogous other than the state values being observed via a hyperbolic tangent non-linearity:

$$y_{t,l} = \tanh(x_{\frac{L}{2}(l-\frac{1}{2})}) + v_{t,l}, \quad v_{t,l} \sim \mathcal{N}(0, \varsigma^2) \quad \forall t \in 1:T, l \in 1:L. \quad (\text{F.29})$$

Although seemingly minor change in the model, as illustrated in the experimental results, introducing this non-linearity was sufficient to significantly degrade the filtering performance of the local ETKF.

APPENDIX G: FULL GRID SEARCH RESULTS FOR LOCAL ETKF

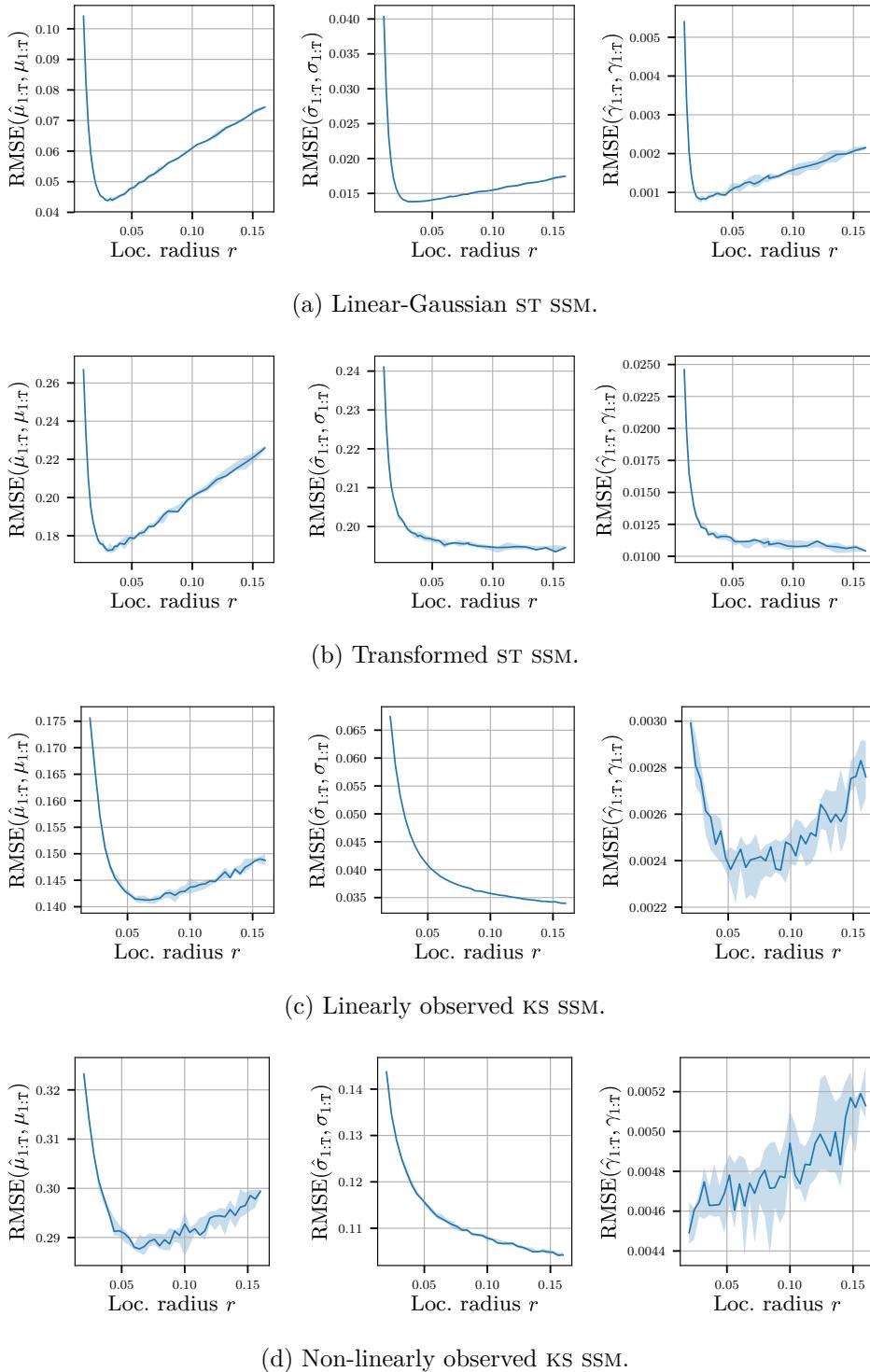


Fig G.1: Values of metrics for all localisation radii r for local ETKF on four SSMs considered in experiments. In all cases the curve shows the median value across five independent runs and the filled region the minimum to maximum range.

APPENDIX H: DETAILS OF MCMC RUNS FOR KS MODELS

A non-centred parametrisation was used for the Hamiltonian Monte Carlo chains for the two KS SSMs (Papaspiliopoulos, Roberts and Sköld, 2007), with the target smoothing distribution formulated in terms of the MTS $\approx 10^6$ dimensional set of state noise variables $u_{1:T}$ which are independently and identically distributed standard normal variables under the prior, with the observation sequence $y_{1:T}$ then having a Gaussian conditional distribution given $u_{1:T}$. The step-size for the integrator of the Hamiltonian dynamics was manually tuned once for each SSM using short pilot chains with a fixed number of integrator steps to achieve an average acceptance probability in the range [0.6, 0.9] (Betancourt, Byrne and Girolami, 2014), with in both SSMs a step size 2.5×10^{-3} found to give an acceptance rate is the target range. The integrator used was a variant of the standard leapfrog / Störmer-Verlet integrator which uses an alternative splitting of the Hamiltonian to leverage an exact analytic solution for the Hamiltonian dynamics under the quadratic potential energy component due to the Gaussian prior (Shahbaba et al., 2014). The number of integrator steps used to generate the Hamiltonian dynamics trajectory in each chain transition was dynamically set on each iteration using a variant of the *No-U-Turn sampler* scheme (Hoffman and Gelman, 2014; Betancourt, 2017), with the chains for both SSMs performing approximately 2×10^3 steps per transition on average. For each SSM the total wall clock time to run the five chains in parallel on a Intel Xeon E5-2620 v4 8-core CPU was around one week.

All chains were initialised from the true state noise sequence $u_{1:T}$ used to generate the observations, which corresponds to a single exact sample from the target distribution $\mathbb{P}(u_{1:T} \in du | y_{1:T} = y_{1:T})$ as the $(u_{1:T}, y_{1:T})$ pair was originally generated from the corresponding joint distribution $\mathbb{P}(u_{1:T} \in du, y_{1:T} = dy)$. Although typically it would be preferable for the robustness of convergence diagnostics based on comparisons between chains to initialise each of the chains independently from an over-dispersed distribution compared to the target such as the prior, here we found the step-size required to robustly achieve an average acceptance probability in the range [0.6, 0.9] for chains initialised from the prior to be much smaller than for chains initialised from the ‘true’ noise sequence $u_{1:T}$, likely due to the differing geometry of the target distribution in the tails (where initialisations from the prior are likely to fall) and typical set, which $u_{1:T}$ as an exact sample from the target should be within. Given the long chain run times even when using the larger step size, a pragmatic choice was therefore made to use a common initialisation. This initialisation scheme and relatively small number of samples in each chain means there is a risk that the chains therefore only explored a subset of the target distributions’ typical sets. As partial evidence against this being the case, visual checks of the estimates of the first and second moments of a subset of the filtering distributions $\pi_{1:T}$ using the final 100 samples from each of the chains suggest that the estimates from the different chains are consistent with each other (see examples in Figs. H.1 and H.2).

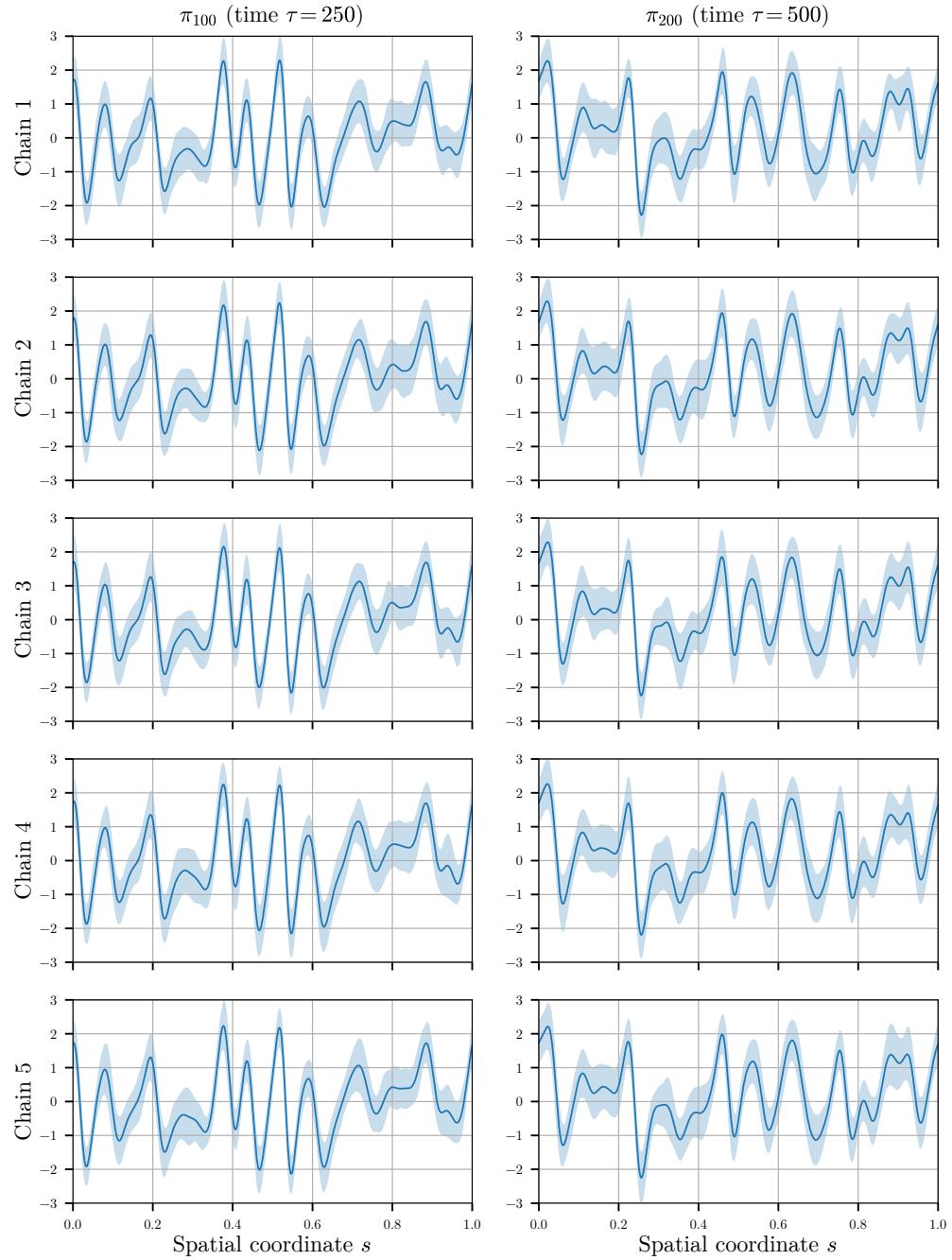


Fig H.1: Comparison of estimates of first and second moments of filtering distributions π_{100} and π_{200} for linearly observed KS SSM using final 100 samples from each of 5 chains (curves show the estimated mean and the filled region the mean \pm two standard deviations).

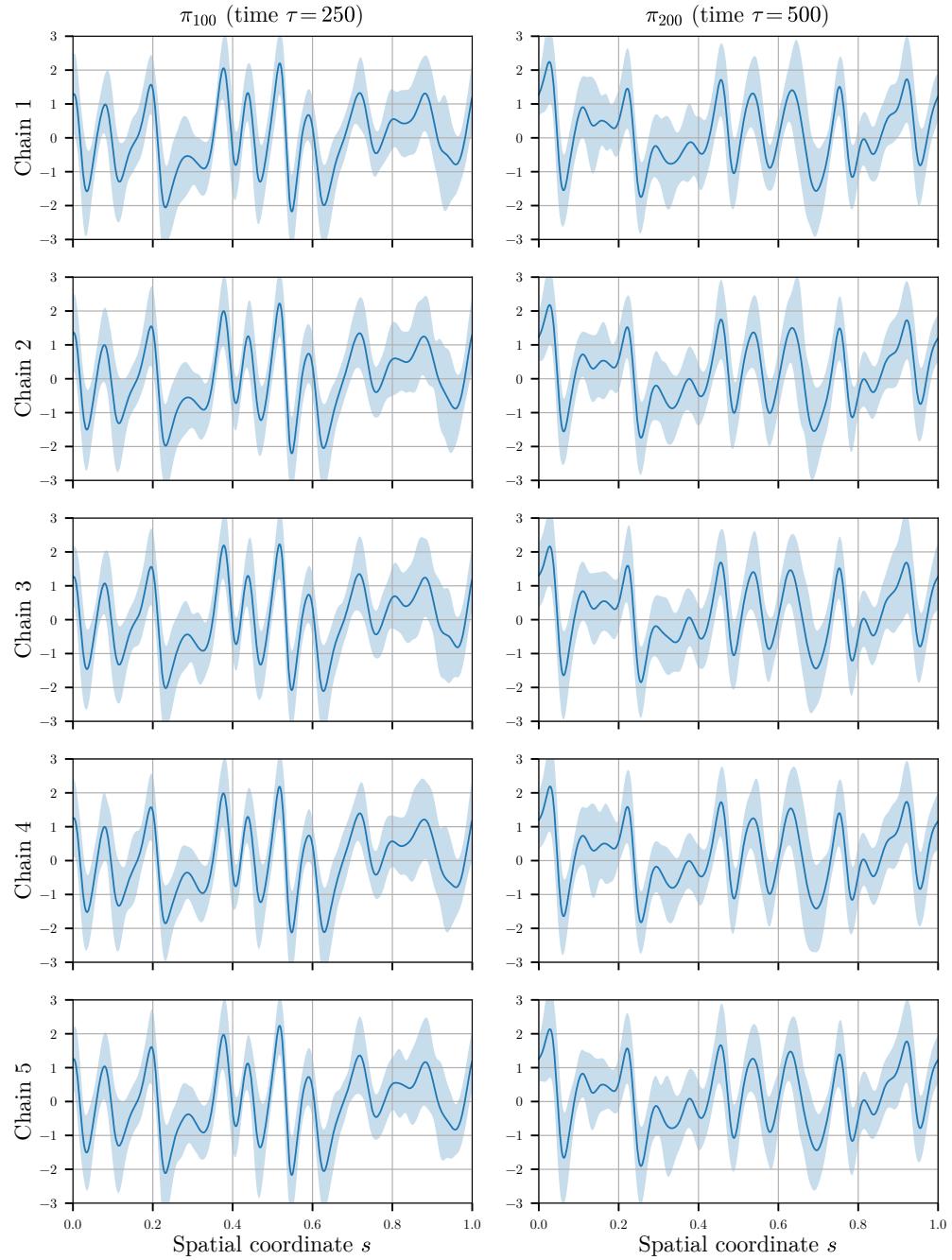


Fig H.2: Comparison of estimates of first and second moments of filtering distributions π_{100} and π_{200} for non-linearly observed KS SSM using final 100 samples from each of 5 chains (curves show the estimated mean and the filled region the mean \pm two standard deviations).

REFERENCES

- ACEVEDO, W., DE WILJES, J. and REICH, S. (2017). Second-order accurate ensemble transform particle filters. *SIAM Journal on Scientific Computing* **39** A1834–A1850.
- ALTSCHULER, J., WEED, J. and RIGOLLET, P. (2017). Near-linear time approximation algorithms for optimal transport via Sinkhorn iteration. In *Advances in Neural Information Processing Systems 30* 1964–1974.
- ANDERSON, J. L. (2001). An ensemble adjustment Kalman filter for data assimilation. *Monthly weather review* **129** 2884–2903.
- BAUER, P., THORPE, A. and BRUNET, G. (2015). The quiet revolution of numerical weather prediction. *Nature* **525** 47.
- BENGTSSON, T., BICKEL, P. and LI, B. (2008). Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman* 316–334. Institute of Mathematical Statistics.
- BERTOLI, F. and BISHOP, A. N. (2014). Adaptively Blocked Particle Filtering with Spatial Smoothing in Large-Scale Dynamic Random Fields. *arXiv:1406.0136*.
- BESKOS, A., CRISAN, D., JASRA, A., KAMATANI, K. and ZHOU, Y. (2017). A stable particle filter for a class of high-dimensional state-space models. *Advances in Applied Probability* **49** 24–48.
- BETANCOURT, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv:1701.02434*.
- BETANCOURT, M., BYRNE, S. and GIROLAMI, M. (2014). Optimizing the integrator step size for Hamiltonian Monte Carlo. *arXiv:1411.6669*.
- BISHOP, A. N. and DEL MORAL, P. (2018). On the Stability of Matrix-Valued Riccati Diffusions. *arXiv preprint arXiv:1808.00235*.
- BISHOP, C. H., ETHERTON, B. J. and MAJUMDAR, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly weather review* **129** 420–436.
- BOLIC, M., DJURIC, P. M. and HONG, S. (2005). Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing* **53** 2442–2450.
- BONAVITA, M., TORRISI, L. and MARCUCCI, F. (2008). The ensemble Kalman filter in an operational regional NWP system: Preliminary results with real observations. *Quarterly Journal of the Royal Meteorological Society* **134** 1733–1744.
- BOWLER, N. E., ARRIBAS, A., BEARE, S. E., MYLNE, K. R. and SHUTTS, G. J. (2009). The local ETKF and SKEB: Upgrades to the MOGREPS short-range ensemble prediction system. *Quarterly Journal of the Royal Meteorological Society* **135** 767–776.
- BUIZZA, R., HOUTEKAMER, P., PELLERIN, G., TOTH, Z., ZHU, Y. and WEI, M. (2005). A comparison of the ECMWF, MSC, and NCEP global ensemble prediction systems. *Monthly Weather Review* **133** 1076–1097.
- BURGERS, G., VAN LEEUWEN, P. J. and EVENSEN, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly weather review* **126** 1719–1724.
- CHENG, Y. and REICH, S. (2015). Assimilating data into scientific models: An optimal coupling perspective. In *Nonlinear Data Assimilation* 75–118. Springer.
- CLAYTON, A. M., LORENC, A. C. and BARKER, D. M. (2013). Operational implementation of a hybrid ensemble / 4D-Var global data assimilation system at the Met Office. *Quarterly Journal of the Royal Meteorological Society* **139** 1445–1461.
- COX, S. M. and MATTHEWS, P. C. (2002). Exponential time differencing for stiff systems. *Journal of Computational Physics* **176** 430–455.
- CUTURI, M. (2013). Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *Advances in Neural Information Processing Systems 26* 2292–2300.
- DEL MORAL, P. (1996). Non-linear filtering: interacting particle resolution. *Markov processes and related fields* **2** 555–581.
- DEL MORAL, P. and TUGAUT, J. (2018). On the stability and the uniform propagation of chaos properties of ensemble Kalman–Bucy filters. *The Annals of Applied Probability* **28** 790–850.
- DOUC, R. and CAPPÉ, O. (2005). Comparison of resampling schemes for particle filtering. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005*. 64–69. IEEE.
- DUANE, S., KENNEDY, A. D., PENDLETON, B. J. and ROWETH, D. (1987). Hybrid Monte Carlo. *Physics Letters B* **195** 216–222.
- EVENSEN, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model

- using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans* **99** 10143–10162.
- EVENSEN, G. (2009). *Data Assimilation: The Ensemble Kalman Filter*, 2nd ed. Springer.
- FARCHI, A. and BOCQUET, M. (2018). Comparison of local particle filters and new implementations. *Nonlinear Processes in Geophysics Discussions* **2018** 1–63.
- FEARNHEAD, P. and KÜNSCH, H. (2018). Particle Filters and Data Assimilation. *Annual Review of Statistics and Its Application* **5** 421–449.
- FREI, M. and KÜNSCH, H. R. (2013). Bridging the ensemble Kalman and particle filters. *Biometrika* **100** 781–800.
- FURRER, R. and BENGTSSON, T. (2007). Estimation of high-dimensional prior and posterior covariance matrices in Kalman filter variants. *Journal of Multivariate Analysis* **98** 227–255.
- GASPARI, G. and COHN, S. E. (1999). Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society* **125** 723–757.
- GERBER, M., CHOPIN, N. and WHITELEY, N. (2019). Negative association, ordering and convergence of resampling methods. *The Annals of Statistics* **47** 2236–2260.
- GORDON, N. J., SALMOND, D. J. and SMITH, A. F. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)* **140** 107–113. IET.
- HAMILL, T. M., WHITAKER, J. S. and SNYDER, C. (2001). Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review* **129** 2776–2790.
- HERBST, E. and SCHORFHEIDE, F. (2019). Tempered particle filtering. *Journal of Econometrics* **210** 26–44.
- HOFFMAN, M. D. and GELMAN, A. (2014). The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal of Machine Learning Research* **15** 1593–1623.
- HOL, J. D., SCHON, T. B. and GUSTAFSSON, F. (2006). On resampling algorithms for particle filters. In *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE* 79–82. IEEE.
- HOUTEKAMER, P. L. and MITCHELL, H. L. (1998). Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review* **126** 796–811.
- HUNT, B. R., KOSTELICH, E. J. and SZUNYOGH, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena* **230** 112–126.
- HYMAN, J. M. and NICOLAENKO, B. (1986). The Kuramoto–Sivashinsky equation: a bridge between PDEs and dynamical systems. *Physica D: Nonlinear Phenomena* **18** 113–126.
- JOHANSEN, A. M. (2015). On blocks, tempering and particle MCMC for systems identification. *IFAC-PapersOnLine* **48** 969–974.
- KALMAN, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering* **82** 35–45.
- KELLY, D. T., LAW, K. and STUART, A. M. (2014). Well-posedness and accuracy of the ensemble Kalman filter in discrete and continuous time. *Nonlinearity* **27** 2579.
- KURAMOTO, Y. and TSUZUKI, T. (1976). Persistent propagation of concentration waves in dissipative media far from thermal equilibrium. *Progress of theoretical physics* **55** 356–369.
- LE GLAND, F., MONBET, V. and TRAN, V.-D. (2011). Large sample asymptotics for the ensemble Kalman filter. In *The Oxford Handbook of Nonlinear Filtering* (D. Crisan and B. Rozovskii, eds.) 598–631. Oxford University Press.
- LEE, A. and WHITELEY, N. (2015). Forest resampling for distributed sequential Monte Carlo. *Statistical Analysis and Data Mining: The ASA Data Science Journal*.
- LEI, J., BICKEL, P. and SNYDER, C. (2010). Comparison of ensemble Kalman filters under non-Gaussianity. *Monthly Weather Review* **138** 1293–1306.
- MAJDA, A. J. and HARLIM, J. (2012). *Filtering complex turbulent systems*. Cambridge University Press.
- MORZFELD, M., HODYSS, D. and SNYDER, C. (2017). What the collapse of the ensemble Kalman filter tells us about particle filters. *Tellus A: Dynamic Meteorology and Oceanography* **69**.
- ORLIN, J. B. (1997). A polynomial time primal network simplex algorithm for minimum cost flows. *Mathematical Programming* **78** 109–129.
- PAPASILIOPOULOS, O., ROBERTS, G. O. and SKÖLD, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science* 59–73.
- PENNY, S. G. and MIYOSHI, T. (2015). A local particle filter for high dimensional geophysical systems. *Nonlinear Processes in Geophysics Discussions* **2** 1631–1658.

- PEYRÉ, G. and CUTURI, M. (2019). *Computational Optimal Transport*. Now Publishers.
- REBESCHINI, P. and VAN HANDEL, R. (2015). Can local particle filters beat the curse of dimensionality? *The Annals of Applied Probability* **25** 2809–2866.
- REICH, S. (2013). A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scientific Computing* **35** A2013–A2024.
- SEN, D. and THIERY, A. H. (2019). Particle filter efficiency under limited communication. *arXiv:1904.09623*.
- SHAHBABA, B., LAN, S., JOHNSON, W. O. and NEAL, R. M. (2014). Split Hamiltonian Monte Carlo. *Statistics and Computing* **24** 339–349.
- SINKHORN, R. and KNOPP, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics* **21** 343–348.
- SIVASHINSKY, G. (1977). Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations. *Acta Astronautica* **4** 1177–1206.
- SNYDER, C. (2011). Particle filters, the ‘optimal’ proposal and high-dimensional systems. In *Proceedings of the ECMWF Seminar on Data Assimilation for atmosphere and ocean* 1–10.
- SNYDER, C., BENGTSSON, T. and MORZFIELD, M. (2015). Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review* **143** 4750–4761.
- SNYDER, C., BENGTSSON, T., BICKEL, P. and ANDERSON, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review* **136** 4629–4640.
- SVENSSON, A., SCHÖN, T. B. and LINDSTEN, F. (2018). Learning of state-space models with highly informative observations: A tempered sequential Monte Carlo solution. *Mechanical Systems and Signal Processing* **104** 915–928.
- TONG, X. T., MAJDA, A. J. and KELLY, D. (2016). Nonlinear stability and ergodicity of ensemble based Kalman filters. *Nonlinearity* **29** 657.
- VAN LEEUWEN, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review* **137** 4089–4114.
- VERGÉ, C., DUBARRY, C., DEL MORAL, P. and MOULINES, E. (2015). On parallel implementation of sequential Monte Carlo methods: the island particle model. *Statistics and Computing* **25** 243–260.
- WHITAKER, J. S. and HAMILL, T. M. (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review* **130** 1913–1924.
- WHITELEY, N., LEE, A. and HEINE, K. (2016). On the role of interaction in sequential Monte Carlo algorithms. *Bernoulli* **22** 494–529.