Received: 16 March 2023 | Revised: 15 September 2023 | Accepted: 6 October 2023

DOI: 10.1002/qj.4595

Quarterly Journal of the Royal Meteorological Society RMetS

RESEARCH ARTICLE

A dynamical Gaussian, lognormal, and reverse lognormal Kalman filter

Senne Van Loon | **Steven J. Fletcher**

Cooperative Institute for Research in the Atmosphere, Colorado State University, Fort Collins, Colorado, USA

Correspondence
Senne Van Loon, Cooperative Institute for Research in the Atmosphere, Atmospheric Science Campus, Colorado State University, 1375 Campus Delivery, Fort Collins, CO 80523-1375, USA.
Email: Senne.Van_Loon@colostate.edu

Funding information
National Science Foundation, Grant/Award Number: AGS-2033405

Abstract

We derive a generalization of the Kalman filter that allows for non-Gaussian background and observation errors. The Gaussian assumption is replaced by considering that the errors come from a mixed distribution of Gaussian, lognormal, and reverse lognormal random variables. We detail the derivation for reverse lognormal errors and extend the results to mixed distributions, where the number of Gaussian, lognormal, and reverse lognormal state variables can change dynamically every analysis time. We test the dynamical mixed Kalman filter robustly on two different systems based on the Lorenz 1963 model, and demonstrate that non-Gaussian techniques generally improve the analysis skill if the observations are sparse and uncertain, compared with the Gaussian Kalman filter.

KEYWORDS

data assimilation, Kalman filter, lognormal, machine learning, non-Gaussian, reverse lognormal

1 | INTRODUCTION

The Kalman filter (Jazwinski, 1970; Kalman, 1960) has been very successful in data assimilation applications due to its simplicity. Its development offered an analytical approach to estimate the optimal state of a dynamical system by linearizing the model equations and measurement operators, and paved the way for ensemble data assimilation (Evensen, 1994). The major drawback of this technique is the fact that it assumes Gaussian-distributed background and observational errors. This Gaussian assumption lies at the basis of many data assimilation methods, while it generally does not hold (Bannister *et al.*, 2020; Foster *et al.*, 2006; Hu *et al.*, 2023; Perron & Sura, 2013; Poterjoy, 2022).

As numerical models and observing systems improve, this shortcoming is becoming more obvious, increasing interest in developing non-Gaussian methods in recent years. Although significant advances have been made in this regard, most available non-Gaussian data assimilation techniques still have challenges to overcome before they can be operationally viable. Approaches based on the particle filter (van Leeuwen, 2009) can in principle deal with any prior distribution and likelihood, but generally need large ensemble sizes to avoid ensemble degeneracy (Snyder *et al.*, 2015). Another avenue to take is to replace the Gaussian prior and likelihood distributions explicitly with a different probability density function (Bishop, 2016; Fletcher & Zupanski, 2006a), where one needs a way of deciding which is the optimal distribution to use. Similarly, Gaussian anamorphosis (see, e.g., Amezcua & Van Leeuwen, 2014; Bertino *et al.*, 2003; Bocquet *et al.*, 2010; Simon & Bertino, 2009; Simon & Bertino, 2012) can be used to transform ensembles into an approximately Gaussian

form. This procedure can handle non-Gaussian errors, but discards skewness information, which can lead to biases in the analysis (Fletcher & Zupanski, 2007). Finally, hybrid approaches have been proposed that combine a particle filter with standard data assimilation methods (see, e.g., Nerger, 2022).

Due to these challenges, operational numerical weather prediction systems are still based on Gaussian prior distributions (Bannister, 2017). Two main approaches are used: variational methods and ensemble techniques. In the former, the negative log-likelihood of the posterior distribution is minimized to find the optimal analysis state. The latter are based on the ensemble Kalman filter, which allows for a flow dependence of the background-error covariance matrix, but including enough ensemble members is computationally expensive. To improve the forecast skill, most operational numerical weather prediction centres use hybrid data assimilation methods, to try to combine the best of both worlds.

In the case of lognormal errors, variational data assimilation methods are already well established (Fletcher & Zupanski, 2006a, 2006b; Fletcher, 2010, 2022) and machine learning has been suggested as a promising way of selecting the underlying error distribution (Goodliff *et al.*, 2020; Goodliff *et al.*, 2022). Recently, Fletcher *et al.* (2023) showed that it is also possible to adapt the Kalman filter to allow for lognormal errors. This paves the way for hybrid techniques that can assimilate variables that are bounded from below better.

The lognormal distribution is just one of many possibilities for the error distribution of state variables. It is a good approximation for variables that are positively skewed and have a set minimum value, such as water-vapour mixing ratio (Kliewer *et al.*, 2016). However, atmospheric statistics can change over time and space (Foster *et al.*, 2006; Perron & Sura, 2013), and in particular the skewness can be negative. One way to deal with this is to introduce the reverse lognormal distribution, which looks like the mirror image of a lognormal distribution, with a given upper bound and negative skewness. The reverse lognormal distribution has been successfully incorporated into variational data assimilation (Goodliff *et al.*, 2023), allowing for a mixed distribution with Gaussian, lognormal, and reverse lognormal random variables.

In a sense, adding the reverse lognormal distribution to previous mixed data assimilation schemes completes the model, making it possible to account for variables with any skewness, positive or negative. Thus, in an effort to design operationally viable hybrid data assimilation methods, the mixed Kalman filter of Fletcher *et al.* (2023) should be appended with the reverse lognormal distribution, which is the motivation of this article.

We start by deriving the reverse lognormal Kalman filter in Section 2.1, where all errors are reverse lognormally distributed. Of course, such a filter is only marginally useful, as in any real application different distributions are present at the same time. In Section 2.2, we outline the fully mixed version of the Kalman filter, which incorporates Gaussian, lognormally, and reverse lognormally distributed background and observational errors. Moreover, we remark that the fully mixed Kalman filter is formulated in a dynamical way, where the underlying distribution of each state variable or observation can change every analysis time. The details of the implementation of the new Kalman filter are presented in Section 3, and used in Section 4 in conjunction with two models based on the Lorenz-63 system (Lorenz, 1963) to investigate its capabilities. Finally, we provide conclusions in Section 5.

2 | FORMULATION

The Kalman filter equations can be derived in multiple ways (Fletcher, 2022). Here, we present a formulation based on a least-squares approach, starting from a cost function describing the negative log likelihood of the analysis errors. To show the details of the derivations, before presenting the full dynamical model with mixed Gaussian, lognormal, and reverse lognormal distribution, we derive the reverse lognormal Kalman filter, where all errors follow a reverse lognormal distribution. The extension to the fully mixed error distribution is trivial, and will be summarized in Section 2.2.

The derivation in this section is similar to the one presented by Fletcher *et al.* (2023) in the context of lognormal errors, and the reader is referred to that work if more details are necessary. The main difference is the use of a different probability density function for the prior and likelihood distributions: the reverse lognormal distribution. Its multivariate probability density function is given by

$$\begin{aligned} &p_{\mathrm{rl}}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma},\boldsymbol{\xi}) \\ &= \frac{1}{\sqrt{(2\pi)^N|\boldsymbol{\Sigma}|}}\prod_{j=1}^{N}\frac{1}{\xi_j - x_j} \\ &\quad\times\exp\left\{-\frac{1}{2}\left[\ln(\boldsymbol{\xi}-\boldsymbol{x})-\boldsymbol{\mu}\right]\boldsymbol{\Sigma}^{-1}\left[\ln(\boldsymbol{\xi}-\boldsymbol{x})-\boldsymbol{\mu}\right]\right\}. \end{aligned} \tag{1}$$

The reverse lognormal distribution can be thought of as a mirror image of a lognormal distribution, and has a negative skewness. Reverse lognormal variables are semi-bounded, with a fixed upper bound ξ, in contrast with lognormal variables that have a lower bound.

2.1 | Reverse lognormal Kalman filter

The starting point for deriving the Kalman filter in a least-squares approach is the cost function that needs to be minimized to find the analysis state. If the background and observational errors follow a reverse lognormal distribution (Fletcher, 2022; Goodliff *et al.*, 2023), then the cost function is given by

$$\begin{aligned} J_{\mathrm{rl}}(\boldsymbol{x}) = &\frac{1}{2}[\ln(\xi_{\mathrm{f}} - \boldsymbol{x}) - \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}})]^{\mathrm{T}} P_{\mathrm{f,rl}}^{-1} \\ &\times [\ln(\xi_{\mathrm{f}} - \boldsymbol{x}) - \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}})] \\ &+ \frac{1}{2}[\ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}))]^{\mathrm{T}} R_{\mathrm{rl}}^{-1} \\ &\times [\ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}))]\,. \end{aligned} \tag{2}$$

Here, $\boldsymbol{x}_{\mathrm{b}}$ is the background state, $\boldsymbol{y}$ are the observations, $\boldsymbol{h}(\boldsymbol{x})$ is the nonlinear observation operator, transforming the state variables to observation space, $P_{\mathrm{f,rl}}$ is the forecast-error covariance matrix, and R_{rl} is the observation-error covariance matrix. The parameters ξ_{f} and ξ_{o} are the maximal values of the reverse lognormal distribution for the state variables and observations respectively. We furthermore denote the total number of state variables with N, such that $\boldsymbol{x}$ and $\boldsymbol{x}_{\mathrm{b}}$ are vectors of size N, while $P_{\mathrm{f,rl}}$ is an $N \times N$ matrix. The number of observed variables is N_{o}, so $\boldsymbol{y}$ and $\boldsymbol{h}(\boldsymbol{x})$ are vectors of size N_{o} and R_{rl} is a matrix of size $N_{\mathrm{o}} \times N_{\mathrm{o}}$.

Note that, similar to the lognormal Kalman filter (Fletcher *et al.*, 2023), we need to use the cost function for which the minimum is the median of the posterior distribution. Although this is not always the best statistic to use (Fletcher *et al.*, 2019), it is valid in the regime where the analysis errors are small. This is a necessary assumption to derive the Kalman filter equations: to find an analytical form of the minimizer of Equation 2, the gradient of the cost function needs to be linearized, as will be described below. Within this approximation (i.e., the analysis errors are small), the use of a median-based cost function is justified.

In contrast to the Gaussian Kalman filter, we do not assume the forecast model to be linear. Instead, we can define the analysis errors $\boldsymbol{e}_{\mathrm{a}}$ and forecast errors $\boldsymbol{e}_{\mathrm{f}}$ at time $t = t_k$ as

$$\ln\left(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{a}}^k\right) = \ln\left(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}}^k\right) - \ln\left(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{t}}^k\right), \tag{3}$$

$$\ln\left(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{f}}^k\right) = \ln\left(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{f}}^k\right) - \ln\left(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{t}}^k\right), \tag{4}$$

with $\boldsymbol{x}_{\mathrm{a}}$ the analysis state (the state that minimizes the cost function in Equation 2), $\boldsymbol{x}_{\mathrm{f}}$ the forecast state, and $\boldsymbol{x}_{\mathrm{t}}$ the true state. The forecast state at time $t = t_k$ can be found by letting the forecast model act upon the perturbed analysis state at $t = t_k$, that is,

$$\boldsymbol{x}_{\mathrm{f}}^k = \mathcal{M}\left[\xi_{\mathrm{f}} - (\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}}^{k-1}) \odot (\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{a}}^{k-1})\right]. \tag{5}$$

The operator $\odot$ denotes the Hadamard (elementwise) product, and $\mathcal{M}$ is the nonlinear model bringing the state variables from time $t = t_{k-1}$ to $t = t_k$. The seemingly complicated form of the perturbed analysis state (the argument of the model $\mathcal{M}$ in Equation 5) emerges from the fact that the errors are multiplicative with an upper bound, and can be found by solving Equation 3 for $\boldsymbol{x}_{\mathrm{t}}$. More details about this transformation can be found in Section 3.2. With these definitions, the forecast-error covariance matrix $P_{\mathrm{f,rl}}$ can be calculated as

$$P_{\mathrm{f,rl}}^k = \left[\ln\left(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{f}}^k\right)\right]\left[\ln\left(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{f}}^k\right)\right]^{\mathrm{T}} + Q^k, \tag{6}$$

where Q is the model-error covariance matrix, disregarding correlations between analysis and model errors. In contrast to the Gaussian Kalman filter, where the forecast-error covariance matrix is calculated by having the linearized forecast model act directly on the analysis-error covariance matrix, the definition of $P_{\mathrm{f,rl}}$ relies on the nonlinear forecast model $\mathcal{M}$, implicit in the calculation of the forecast errors defined in Equation 4. For non-Gaussian errors, there is no general way of defining a forecast-error covariance matrix equivalent to the Gaussian definition, because the model operator does not commute with the logarithms (Fletcher *et al.*, 2023). This has the added benefit of removing errors introduced by linearizing the model. This also implies that the true state $\boldsymbol{x}_{\mathrm{t}}$ is still present in the equations for the forecast error, although it is generally unknown. Therefore, we have to approximate the true state by running the analysis state of the previous assimilation step through the forecast model: $\boldsymbol{x}_{\mathrm{t}}^k \approx \boldsymbol{x}_{\mathrm{b}}^k = \mathcal{M}(\boldsymbol{x}_{\mathrm{a}}^{k-1})$. Finally, one should be aware that the forecast-error covariance matrix defined in Equation 6 is only an approximation to the covariance of the prior, as the true state and exact model errors are not known.

At each assimilation time, the cost function J_{rl} needs to be minimized, or, equivalently, the root of the gradient of the cost function has to be found. Differentiating the cost function of Equation 2 with respect to $\boldsymbol{x}$, one finds

$$\begin{aligned} \nabla_{\boldsymbol{x}} J_{\mathrm{rl}}(\boldsymbol{x}) = &W_{\mathrm{f,rl}}^{-\mathrm{T}}(\boldsymbol{x}) P_{\mathrm{f,rl}}^{-1} [\ln(\xi_{\mathrm{f}} - \boldsymbol{x}) - \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}})] \\ &- H_{\boldsymbol{x}}^{\mathrm{T}} W_{\mathrm{o,rl}}^{-\mathrm{T}}(\boldsymbol{x}) R_{\mathrm{rl}}^{-1} [\ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}))]\,. \end{aligned} \tag{7}$$

Here, we have introduced the Jacobian of the observation operator $H_{\boldsymbol{x}} = \nabla_{\boldsymbol{x}} \boldsymbol{h}(\boldsymbol{x})$ (a matrix of size $N_{\mathrm{o}} \times N$) and the diagonal matrices $W_{\mathrm{f,rl}}(\boldsymbol{x}) = \mathrm{diag}[x_1 - \xi_{\mathrm{f}}, \ldots, x_N - \xi_{\mathrm{f}}]$ and $W_{\mathrm{o,rl}}(\boldsymbol{x}) = \mathrm{diag}[h_1(\boldsymbol{x}) - \xi_{\mathrm{o}}, \ldots, h_{N_{\mathrm{o}}}(\boldsymbol{x}) - \xi_{\mathrm{o}}]$, arising from the derivatives of the logarithms in the cost function.

To find the best estimate of the analysis state, one is interested in the solution to $\nabla_{\boldsymbol{x}} J_{\mathrm{rl}}(\boldsymbol{x}_{\mathrm{a}}) = 0$ of the form

$$\begin{aligned}\ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}}) = \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}}) \\ + K_{\mathrm{rl}}\left[\ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{b}}))\right],\end{aligned} \tag{8}$$

with K_{rl}, the Kalman gain matrix, of size $N \times N_{\mathrm{o}}$. To do this, one needs a way of writing $\ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{a}}))$ in terms of $\ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}})$ and the background state, by making use of the geometric tangent linear model introduced by Fletcher and Jones (2014). Specifically, introducing the small parameter $\boldsymbol{r} = \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}}) - \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}})$, one can perform a Taylor expansion for small $\boldsymbol{r}$ on the function $f(\boldsymbol{r}) \equiv \ln\{\xi_{\mathrm{o}} - \boldsymbol{h}[\boldsymbol{x}_{\mathrm{a}}(\boldsymbol{r})]\}$, that is,

$$\begin{aligned}f(\boldsymbol{r}) &= \ln\left\{\xi_{\mathrm{o}} - \boldsymbol{h}\left[\xi_{\mathrm{f}} - (\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}})\mathrm{e}^{\boldsymbol{r}}\right]\right\} \\ &= \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{b}})) + W_{\mathrm{o,rl}}^{-1}(\boldsymbol{x}_{\mathrm{b}}) H_{\boldsymbol{x}_{\mathrm{b}}} W_{\mathrm{f,rl}}(\boldsymbol{x}_{\mathrm{b}}) \\ &\quad \times \left[\ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}}) - \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}})\right] + \mathrm{O}(\boldsymbol{r}^2).\end{aligned} \tag{9}$$

Using this Taylor expansion in the gradient of the cost function (Equation 7), one can rewrite $\nabla_{\boldsymbol{x}} J_{\mathrm{rl}}(\boldsymbol{x}_{\mathrm{a}}) = 0$ as

$$\begin{aligned}P_{\mathrm{f,rl}}^{-1}\boldsymbol{r} = W_{\mathrm{f,rl}}^{\mathrm{T}}(\boldsymbol{x}_{\mathrm{a}}) H_{\boldsymbol{x}_{\mathrm{a}}}^{\mathrm{T}} W_{\mathrm{o,rl}}^{-\mathrm{T}}(\boldsymbol{x}_{\mathrm{a}}) R_{\mathrm{rl}}^{-1} \\ \times \left[\ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{b}})) \right. \\ \left. - W_{\mathrm{o,rl}}^{-1}(\boldsymbol{x}_{\mathrm{b}}) H_{\boldsymbol{x}_{\mathrm{b}}} W_{\mathrm{f,rl}}(\boldsymbol{x}_{\mathrm{b}})\boldsymbol{r} + \mathcal{O}(\boldsymbol{r}^2)\right].\end{aligned} \tag{10}$$

To find this form, the small parameter $\boldsymbol{r}$ was recognized in the first term of Equation 7, and the whole expression was multiplied by $W_{\mathrm{f,rl}}^{\mathrm{T}}$ from the left. Finally, note that $\boldsymbol{r}_{\mathrm{o}} = \ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{b}}))$ should be of a similar order to $\boldsymbol{r}$, assuming the observational innovations are small. In order to keep the entire right hand side of Equation 10 linear in $\boldsymbol{r}$ and $\boldsymbol{r}_{\mathrm{o}}$, the prefactor should be expanded as well. In other words, we want to omit all terms of order $\boldsymbol{r}^2$, $\boldsymbol{r}_{\mathrm{o}}^2$, $\boldsymbol{r}\boldsymbol{r}_{\mathrm{o}}$, or higher in the full expression of Equation 10. As the terms between brackets are already of the order $\boldsymbol{r}$ or $\boldsymbol{r}_{\mathrm{o}}$, the prefactor can be expanded up to zeroth order in $\boldsymbol{r}$:

$$\begin{aligned}W_{\mathrm{f,rl}}^{\mathrm{T}}[\boldsymbol{x}_{\mathrm{a}}(\boldsymbol{r})] H_{\boldsymbol{x}_{\mathrm{a}}(\boldsymbol{r})}^{\mathrm{T}} W_{\mathrm{o,rl}}^{-\mathrm{T}}[\boldsymbol{x}_{\mathrm{a}}(\boldsymbol{r})] \\ = W_{\mathrm{f,rl}}^{\mathrm{T}}(\boldsymbol{x}_{\mathrm{b}}) H_{\boldsymbol{x}_{\mathrm{b}}}^{\mathrm{T}} W_{\mathrm{o,rl}}^{-\mathrm{T}}(\boldsymbol{x}_{\mathrm{b}}) + \mathrm{O}(\boldsymbol{r}),\end{aligned} \tag{11}$$

because $\boldsymbol{x}_{\mathrm{a}}(\boldsymbol{r} = \boldsymbol{0}) = \boldsymbol{x}_{\mathrm{b}}$. Defining $\tilde{H}_{\mathrm{rl}} = W_{\mathrm{o,rl}}^{-1}(\boldsymbol{x}_{\mathrm{b}}) H_{\boldsymbol{x}_{\mathrm{b}}} W_{\mathrm{f,rl}}(\boldsymbol{x}_{\mathrm{b}})$, Equation 10 can be written as

$$P_{\mathrm{f,rl}}^{-1}\boldsymbol{r} = \tilde{H}_{\mathrm{rl}} R_{\mathrm{rl}}^{-1}\left(\boldsymbol{r}_{\mathrm{o}} - \tilde{H}_{\mathrm{rl}}\boldsymbol{r}\right) + \mathrm{O}(\boldsymbol{r}_{\mathrm{o}}, \boldsymbol{r})^2. \tag{12}$$

Performing some additional algebra (including the use of the Sherman–Morrison–Woodby formula; see Fletcher (2022) and Fletcher *et al.* (2023) for a detailed derivation), the analysis state can be written as

$$\begin{aligned}\ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}}) = \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}}) + P_{\mathrm{f,rl}}\tilde{H}_{\mathrm{rl}}\left(\tilde{H}_{\mathrm{rl}} P_{\mathrm{f,rl}} \tilde{H}_{\mathrm{rl}}^{\mathrm{T}} + R_{\mathrm{rl}}\right)^{-1} \\ \times \left[\ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{b}}))\right],\end{aligned} \tag{13}$$

such that the gain matrix of the reverse lognormal Kalman filter is given by

$$K_{\mathrm{rl}} = P_{\mathrm{f,rl}}\tilde{H}_{\mathrm{rl}}^{\mathrm{T}}\left(\tilde{H}_{\mathrm{rl}} P_{\mathrm{f,rl}} \tilde{H}_{\mathrm{rl}}^{\mathrm{T}} + R_{\mathrm{rl}}\right)^{-1}. \tag{14}$$

This result is very similar to the gain matrix in a Gaussian (Kalman, 1960) or lognormal (Fletcher *et al.*, 2023) framework, with the only difference being the scaling of the Jacobian of the observation operator.

Lastly, in the Kalman filter scheme, the analysis-error covariance matrix should be updated, in order to estimate the forecast errors in the next assimilation step. The analysis errors are defined in Equation 3, and can be used to form the analysis-error covariance matrix by replacing $\ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{a}})$ in favour of the Kalman gain matrix through Equation 13. Using a similar Taylor expansion to Equation 9, $\ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{b}}))$ can be written as a function of the true state $\boldsymbol{x}_{\mathrm{t}}$ and the background error defined through $\ln(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{b}}) = \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{b}}) - \ln(\xi_{\mathrm{f}} - \boldsymbol{x}_{\mathrm{t}})$. In particular, Equation 9 can be used after replacing $\boldsymbol{r} \rightarrow -\ln(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{b}})$ and $\boldsymbol{x}_{\mathrm{a}} \rightarrow \boldsymbol{x}_{\mathrm{t}}$ to find the correct Taylor expansion. With this, the analysis error can be written as

$$\begin{aligned}\ln(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{a}}) = (\boldsymbol{I} - K_{\mathrm{rl}}\tilde{H}_{\mathrm{rl}})\ln(\xi_{\mathrm{f}} - \boldsymbol{e}_{\mathrm{b}}) \\ + K_{\mathrm{rl}}\ln(\xi_{\mathrm{o}} - \boldsymbol{e}_{\mathrm{o}}),\end{aligned} \tag{15}$$

with $\ln(\xi_{\mathrm{o}} - \boldsymbol{e}_{\mathrm{o}}) = \ln(\xi_{\mathrm{o}} - \boldsymbol{y}) - \ln(\xi_{\mathrm{o}} - \boldsymbol{h}(\boldsymbol{x}_{\mathrm{t}}))$. Assuming the background and observational errors are uncorrelated, the analysis-error covariance matrix becomes

$$P_{\mathrm{a,rl}} = (\boldsymbol{I} - K_{\mathrm{rl}}\tilde{H}_{\mathrm{rl}})P_{\mathrm{f,rl}}(\boldsymbol{I} - K_{\mathrm{rl}}\tilde{H}_{\mathrm{rl}})^{\mathrm{T}} + K_{\mathrm{rl}}R_{\mathrm{rl}}K_{\mathrm{rl}}^{\mathrm{T}}, \tag{16}$$

which is the inverse of the scaled Hessian matrix of the cost function, that is, $P_{\mathrm{a,rl}} = (W_{\mathrm{f,rl}}^{\mathrm{T}}(\boldsymbol{x}_{\mathrm{b}})\nabla_{\boldsymbol{x}}^2 J_{rl}(\boldsymbol{x}_{\mathrm{b}}) W_{\mathrm{f,rl}})^{-1}$.

2.2 | Dynamical mixed Kalman filter

Using the results presented in the last section and combining them with the existing literature on the Gaussian and lognormal Kalman filters (Fletcher *et al.*, 2023; Jazwinski, 1970; Kalman, 1960), it is straightforward to extend the equations to mixed distributions, where now the error of a specific state variable is a random variate from a Gaussian, lognormal, or reverse lognormal distribution. The starting point for the dynamical mixed Kalman filter is again the

cost function, now defined as

$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{X}-\boldsymbol{X}_\mathrm{b})^\mathrm{T} P_\mathrm{f}^{-1}(\boldsymbol{X}-\boldsymbol{X}_\mathrm{b}) + \frac{1}{2}(\boldsymbol{Y}-\boldsymbol{H}(\boldsymbol{x}))^\mathrm{T} R^{-1}(\boldsymbol{Y}-\boldsymbol{H}(\boldsymbol{x})). \tag{17}$$

The capitalized letters represent the mixed variables

$$\boldsymbol{X} = \begin{pmatrix} \boldsymbol{x}^{(g)} \\ \ln \boldsymbol{x}^{(l)} \\ \ln(\xi_\mathrm{f} - \boldsymbol{x}^{(r)}) \end{pmatrix}, \qquad \boldsymbol{X}_\mathrm{b} = \begin{pmatrix} \boldsymbol{x}_\mathrm{b}^{(g)} \\ \ln \boldsymbol{x}_\mathrm{b}^{(l)} \\ \ln(\xi_\mathrm{f} - \boldsymbol{x}_\mathrm{b}^{(r)}) \end{pmatrix}, \tag{18}$$

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{y}^{(g)} \\ \ln \boldsymbol{y}^{(l)} \\ \ln(\xi_\mathrm{o} - \boldsymbol{y}^{(r)}) \end{pmatrix}, \qquad \boldsymbol{H}(\boldsymbol{x}) = \begin{pmatrix} \boldsymbol{h}^{(g)}(\boldsymbol{x}) \\ \ln \boldsymbol{h}^{(l)}(\boldsymbol{x}) \\ \ln(\xi_\mathrm{o} - \boldsymbol{h}^{(r)}(\boldsymbol{x})) \end{pmatrix}, \tag{19}$$

where the superscripts (g), (l), and (r) depict all Gaussian, lognormal, and reverse lognormal variables respectively, which were sorted without loss of generality. The total number of state variables is still N, containing n_g Gaussian, n_l lognormal, and n_r reverse lognormal components. We allow the underlying distributions to change over time, such that $n_\mathrm{g} = n_\mathrm{g}(t)$, $n_\mathrm{l} = n_\mathrm{l}(t)$, and $n_\mathrm{r} = n_\mathrm{r}(t)$, while always keeping $N = n_\mathrm{g} + n_\mathrm{l} + n_\mathrm{r}$ fixed. Similarly, for the observed variables we denote $N_\mathrm{o} = n_\mathrm{g}^\mathrm{o} + n_\mathrm{l}^\mathrm{o} + n_\mathrm{r}^\mathrm{o}$, where the amount of observations within a certain distribution can change over time.

The derivation of the mixed Kalman filter equations is very similar to the reverse lognormal filter described above. By solving for the root of the gradient of the cost function $\nabla_{\boldsymbol{x}} J(\boldsymbol{x}_\mathrm{a}) = 0$ and performing an appropriate Taylor expansion to linearize the equation, one finds the mixed Kalman gain matrix as

$$K = P_\mathrm{f}\tilde{H}^\mathrm{T}\left(\tilde{H}P_\mathrm{f}\tilde{H}^\mathrm{T} + R\right)^{-1}, \tag{20}$$

from which the analysis state can be calculated:

$$\boldsymbol{X}_\mathrm{a} = \boldsymbol{X}_\mathrm{b} + K\left(\boldsymbol{Y} - \boldsymbol{H}(\boldsymbol{x}_\mathrm{b})\right). \tag{21}$$

Here, the scaled Jacobian of the observation operator is defined as $\tilde{H} = W_\mathrm{o}^{-1}(\boldsymbol{x}_\mathrm{b}) H_{\boldsymbol{x}_\mathrm{b}} W_\mathrm{f}(\boldsymbol{x}_\mathrm{b})$, with

$$\begin{aligned} W_\mathrm{f}(\boldsymbol{x}) &= \mathrm{diag}\left[\mathbf{1}_{n_\mathrm{g}}, \boldsymbol{x}^{(l)}, \boldsymbol{x}^{(r)} - \xi_\mathrm{f}\right] \qquad \text{and} \\ W_\mathrm{o}(\boldsymbol{x}) &= \mathrm{diag}\left[\mathbf{1}_{n_\mathrm{g}^\mathrm{o}}, \boldsymbol{h}^{(l)}(\boldsymbol{x}), \boldsymbol{h}^{(r)}(\boldsymbol{x}) - \xi_\mathrm{o}\right], \end{aligned} \tag{22}$$

where $\mathbf{1}_{n_\mathrm{g}}$ denotes a vector of size n_g with ones for each element. The analysis-error covariance matrix is updated as

$$P_\mathrm{a} = (\boldsymbol{I} - K\tilde{H})P_\mathrm{f}(\boldsymbol{I} - K\tilde{H})^\mathrm{T} + KRK^\mathrm{T}. \tag{23}$$

The formulation of the mixed Kalman filter is again analogous to the Gaussian, lognormal, or reverse lognormal filters. The main difference is that it is expressed in terms of the mixed variables, and thus also the errors are in a mixed representation. This needs to be remembered whenever the forecast model is used. Note that the mixed Kalman filter described here is a generalization of Kalman filters with a certain assumption for prior distributions. For example, taking $N = n_\mathrm{g}$ and $N_\mathrm{o} = n_\mathrm{g}^\mathrm{o}$ (such that $n_\mathrm{l} = n_\mathrm{r} = n_\mathrm{l}^\mathrm{o} = n_\mathrm{r}^\mathrm{o} = 0$), a Gaussian Kalman filter is recovered. However, it should be kept in mind that the fully nonlinear model is used, and thus this situation coincides with a nonlinear Gaussian Kalman filter. The details of the implementation of the mixed Kalman filter are reported in the next section.

3 | IMPLEMENTATION

The actual use of the Kalman filter in a data assimilation setting can be partitioned into three main steps: forecasting, transforming, and updating. Here, we provide details of these different steps and how to implement the dynamical mixed Kalman filter.

3.1 | Forecast step

At a given time $t = t_k$ in a data assimilation cycle, one wants to find the optimal analysis state $\boldsymbol{x}_\mathrm{a}^k$, based on observations $\boldsymbol{y}^k$ and the knowledge of the previous states at $t < t_k$. To use the latter information, it is necessary to propagate the previous analysis state $\boldsymbol{x}_\mathrm{a}^{k-1}$ through the forecast model. This defines the background state $\boldsymbol{x}_\mathrm{b}^k = \mathcal{M}(\boldsymbol{x}_\mathrm{a}^{k-1})$. As a reminder, in contrast to the original formulation of the Kalman filter, we use the nonlinear model $\mathcal{M}$, not the linearized version.

In order to calculate the forecast-error covariance matrix, one also needs the forecast errors. We define the forecast state as

$$\boldsymbol{x}_\mathrm{f}^k = \mathcal{M}(\boldsymbol{x}_\mathrm{a}^{k-1} \star \boldsymbol{e}_\mathrm{a}^{k-1}), \tag{24}$$

where we introduced the short-hand notation

$$\boldsymbol{x} \star \boldsymbol{e} \equiv \begin{pmatrix} \boldsymbol{x}^{(g)} + \boldsymbol{e}^{(g)} \\ \boldsymbol{x}^{(l)} \odot \boldsymbol{e}^{(l)} \\ \xi_\mathrm{f} - (\xi_\mathrm{f} - \boldsymbol{x}^{(r)}) \odot (\xi_\mathrm{f} - \boldsymbol{e}^{(r)}) \end{pmatrix}. \tag{25}$$

This operation result from transforming the sum of two mixed variables $\boldsymbol{X} + \boldsymbol{E}$ back to the original variable representation. Details of this transformation and a description of the mixed variables can be found in Section 3.2. The

forecast-error covariance matrix P_{f}^k is defined through the mixed forecast errors $\boldsymbol{E}_{\mathrm{f}}^k = \boldsymbol{X}_{\mathrm{f}}^k - \boldsymbol{X}_{\mathrm{b}}^k$ as

$$P_{\mathrm{f}}^k = \overline{(\boldsymbol{E}_{\mathrm{f}}^k)(\boldsymbol{E}_{\mathrm{f}}^k)^{\mathrm{T}}} + Q^k, \tag{26}$$

where the capital letters $\boldsymbol{X}_{\mathrm{f}}$, $\boldsymbol{X}_{\mathrm{b}}$, and $\boldsymbol{E}_{\mathrm{f}}$ represent the mixed forms of $\boldsymbol{x}_{\mathrm{f}}$, $\boldsymbol{x}_{\mathrm{b}}$, and $\boldsymbol{e}_{\mathrm{f}}$, and the overline denotes the expected value. Note that the forecast-error covariance matrix as defined in Equation 26 is only an approximation, as it actually describes the errors of the forecast state with respect to the true state. As the true state is unknown in general, the approximation $\boldsymbol{x}_{\mathrm{t}}^k \approx \boldsymbol{x}_{\mathrm{b}}^k$ is made when calculating P_{f}^k.

In each assimilation window, both the previous analysis state $\boldsymbol{x}_{\mathrm{a}}^{k-1}$ and the perturbed state $\boldsymbol{x}_{\mathrm{a}}^{k-1} \star \boldsymbol{e}_{\mathrm{a}}^{k-1}$ have to be forecast, using the nonlinear model $\mathcal{M}$. If $k = 1$, namely at the beginning of the data assimilation cycle, one does not have access to the previous analysis state and error. One usually starts with an initial guess for the previous analysis state $\boldsymbol{x}_{\mathrm{a}}^0$. Multiple choices can be made, however, for the initial perturbations $\boldsymbol{e}_{\mathrm{a}}^0$. One possibility is to start with random noise, where the errors are drawn from an unbiased distribution. Alternatively, one can base the initial perturbations on a known climatology, such as in the National Meteorological Center (NMC) method (Parrish & Derber, 1992), where forecasts from different initial times are compared with each other to estimate the errors.

3.2 | Transform step

From the forecast step, we have access to the background state $\boldsymbol{x}_{\mathrm{b}}^k$ and the forecast state $\boldsymbol{x}_{\mathrm{f}}^k$. The mixed Kalman filter, however, is defined in terms of the mixed random variables. Some sort of decision function is necessary to determine which components need to be transformed. How to define this decision function is still an open question for most problems, although advances have been made with machine-learning techniques (Goodliff *et al.*, 2020, 2022, 2023).

Assuming the decision function is known, the transformation $\boldsymbol{X} \equiv \mathcal{T}(\boldsymbol{x})$ itself is straightforward: the Gaussian components do not change $\boldsymbol{X}^{(g)} = \boldsymbol{x}^{(g)}$, while the lognormal variables are calculated as $\boldsymbol{X}^{(l)} = \ln \boldsymbol{x}^{(l)}$, and the reverse lognormal variables as $\boldsymbol{X}^{(r)} = \ln(\xi_{\mathrm{f}} - \boldsymbol{x}^{(r)})$. Although the state variables in Equation 18 are sorted with respect to their distribution for notational simplicity, this is not a necessary condition of the mixed variables and is not required to use the mixed Kalman filter.

A similar transformation also needs to be done in observation space, to define the mixed random variables $\boldsymbol{Y} \equiv \mathcal{T}_{\mathrm{o}}(\boldsymbol{y})$ given in Equation 19. Note that the observation operator $\boldsymbol{h}$ acts on the original (lower-case) variables directly, and thus the transformation needs to be done after applying $\boldsymbol{h}$, that is, $\boldsymbol{H}(\boldsymbol{x}) = \mathcal{T}_{\mathrm{o}}[\boldsymbol{h}(\boldsymbol{x})]$. In general, the decision function in observation space can be different from the decision function in forecast space, that is, $\mathcal{T} \neq \mathcal{T}_{\mathrm{o}}$.

As we will specify in the next section, the output of the update step from the Kalman equation is always in the mixed representation, namely one obtains $\boldsymbol{X}_{\mathrm{a}} = \mathcal{T}(\boldsymbol{x}_{\mathrm{a}})$ and $\boldsymbol{E}_{\mathrm{a}} = \mathcal{T}(\boldsymbol{e}_{\mathrm{a}})$. In the next forecast step, one instead needs the state variables themselves. They can easily be computed through the inverse variable transform

$$\boldsymbol{x} = \mathcal{T}^{-1}(\boldsymbol{X}) = \begin{pmatrix} \boldsymbol{X}^{(g)} \\ \mathrm{e}^{\boldsymbol{X}^{(l)}} \\ \xi_{\mathrm{f}} - \mathrm{e}^{\boldsymbol{X}^{(r)}} \end{pmatrix}, \tag{27}$$

which can be used to find $\boldsymbol{x}_{\mathrm{a}} = \mathcal{T}^{-1}(\boldsymbol{X}_{\mathrm{a}})$ and $\boldsymbol{e}_{\mathrm{a}} = \mathcal{T}^{-1}(\boldsymbol{E}_{\mathrm{a}})$. The operation $\boldsymbol{x} \star \boldsymbol{e}$, introduced in Equation 25, is characterized through this inverse variable transform. Specifically, it is defined as $\boldsymbol{x} \star \boldsymbol{e} = \mathcal{T}^{-1}[\mathcal{T}(\boldsymbol{x}) + \mathcal{T}(\boldsymbol{e})]$, and thus represents a sum of mixed variables.

3.3 | Update step

Finally, the mixed distributed random variables can be used to calculate the Kalman gain matrix defined in Equation 20, and the analysis state $\boldsymbol{x}_{\mathrm{a}}^k$ can be determined through Equation 21. Moreover, the gain matrix can be used to update the analysis-error covariance matrix P_{a}^k, according to Equation 23.

In the original (Gaussian) formulation of the Kalman filter, the analysis-error covariance matrix is propagated directly to the next time step through the linearized forecast model, to define the forecast-error covariance matrix P_{f}^{k+1}. However, because the model operator $\mathcal{M}$ and the variable transform $\mathcal{T}$ do not commute, this is not possible here, and instead we define the forecast-error covariance matrix from the forecast errors $\boldsymbol{e}_{\mathrm{f}}$ directly, as described in Section 3.1. Thus, instead of calculating P_{a}, we calculate the analysis errors directly as

$$\boldsymbol{E}_{\mathrm{a}} = (\boldsymbol{I} - K\tilde{H})\boldsymbol{E}_{\mathrm{f}} + K\boldsymbol{E}_{\mathrm{o}}, \tag{28}$$

with $\boldsymbol{E}_{\mathrm{o}}$ the observation error. Note that the analysis-error covariance matrix (see Equation 23) can be calculated as $P_{\mathrm{a}} = \boldsymbol{E}_{\mathrm{a}}\boldsymbol{E}_{\mathrm{a}}^{\mathrm{T}}$, assuming the forecast and observation errors are uncorrelated. After performing the inverse variable transform, described in Section 3.2, the analysis state and errors can be used to compute the forecast state for the next time step, using Equation 24, and the cycle repeats.

4 | RESULTS

To test the implementation of the dynamical mixed Kalman filter, we perform some experiments with a toy model. We choose the Lorenz-63 model (Lorenz, 1963), which has been shown to contain lognormal and reverse lognormal signals (Goodliff *et al.*, 2020; 2022; 2023). The governing differential equations are given by

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = \rho x - y - xz, \\ \frac{dz}{dt} = xy - \beta z, \end{cases} \tag{29}$$

with $\boldsymbol{x} = (x, y, z)$ the state variables and $(\sigma, \rho, \beta) = (10, 28, 8/3)$ the parameters of the model, which have been chosen to reflect chaotic behaviour. According to Goodliff *et al.* (2020), the x and y components of the Lorenz-63 system satisfy the Gaussian assumption, while z changes distributions over time, switching between Gaussian, lognormal, and reverse lognormal errors, making this an ideal system to test the dynamical mixed Kalman filter.

4.1 | Decision function

In order to assess the dynamical features of our mixed Kalman filter, a decision function is necessary to select the correct distribution to use for the z component at every assimilation step. We follow a technique similar to the one developed by Goodliff *et al.* (2022), based on machine-learning methods.

First, a long control run for $t \in [0, 1000]$ is generated from the initial conditions $\boldsymbol{x}_0 = (-3, -3, 20)$ with time step $dt = 0.01$. The first 100 time steps are removed to account for spinup. At a given time t_k, the skewness of the z variable is computed by performing a skewness test on the sample $\{z^{k-w}, \ldots, z^k, \ldots z^{k+w}\}$, with w the sample window radius determining the number of values used for the skewness test. In the results shown in this section, we use $w = 14$.

The skewness test is a two-sided hypothesis test, with the null hypothesis being that the skewness of the sample is zero (the skewness of a Gaussian distribution). It outputs a z-score expressing the deviation of the skewness of the sample from the skewness of a Gaussian distribution (no skew). We use a significance level of $\alpha \simeq 0.3$, rejecting the null hypothesis if the z-score is outside the domain $[-1, 1]$. If the z-score is larger than 1, we assume the errors of the z variable of the Lorenz model to be lognormally distributed; when it is smaller than -1, we assume a reverse lognormal distribution; if the null hypothesis is accepted, Gaussian errors are assumed. The significance level is chosen in order to produce enough cases where lognormal and reverse lognormal errors are predicted, and is reproduced from Goodliff *et al.* (2022). For example, for $\alpha = 0.3$, the z component behaves lognormally about 40% of the time and reverse lognormally about 15% of the time. This threshold could be optimized for better performance of the machine-learning model, but this goes beyond the scope of the current work.

Once the data are divided into these three bins (Gaussian, lognormal, and reverse lognormal, based on the skewness test), we train a k-nearest-neighbour algorithm to detect the underlying distribution without the need for the skewness test and the knowledge of z at multiple time steps. This is a classification algorithm that makes a decision boundary from a majority vote of the k nearest neighbours from the training data. In particular, we use the k-nearest-neighbour algorithm from `scikit-learn` (Pedregosa *et al.*, 2011), with $k = 15$ and inverse-distance weights. The training input data are the randomized and standardized time series of the x and y components of the Lorenz model, and the training output is the underlying distribution of the z variable, decided by the skewness test. We keep 30% of the original control run for testing, to find an accuracy of 98.7% in predicting the correct distribution.

Once the machine-learning algorithm is trained, it can be used to predict the underlying error distribution of the z variable, based on the input of x and y at a given time, and can thus be used as the decision function. The advantage of the k-nearest-neighbour algorithm is that it is incredibly fast, both in training and in predicting, such that the extra computational time for deciding the z-distribution is minimal.

4.2 | Twin experiments

Now that we have all the ingredients, it is possible to evaluate the accuracy of the dynamical mixed Kalman filter. To do so, we perform twin experiments with the Lorenz-63 model repeatedly to estimate the robustness of the proposed filter. For each assimilation cycle, the following steps are executed.

(i) Generate the true state $\boldsymbol{x}_t$ by integrating the Lorenz-63 model from $t = 0$ to $t = t_{\max}$ from initial condition $\boldsymbol{x}_0$.
(ii) Generate artificial observations by sampling from a distribution with mode $\boldsymbol{x}_t$ and standard deviation $\sigma_{\boldsymbol{x}}$.
(iii) Apply the Kalman filter to the observations from some initial state $\boldsymbol{x}_{0,\mathrm{DA}}$.

(iv) Calculate the root-mean-square error $\mathrm{RMSE} = \left[\frac{1}{N_a}\sum_{k=1}^{N_a}(\boldsymbol{x}_a^k - \boldsymbol{x}_t^k)^2\right]^{1/2}$.

Each run contains $N_a = 250$ assimilation times (and thus observations), and we repeat the experiment $N_c = 50$ times. In the end, the average root-mean-square error is taken over all runs

$$\overline{\mathrm{RMSE}} = \frac{1}{N_c}\sum_{j=1}^{N_c}\mathrm{RMSE}_j,$$

to conclude with a single value characterizing the accuracy of the filter. Note that, for lognormal and reverse lognormal errors, the root-mean-square error might not be the ideal statistic to use. However, because the error distributions are allowed to change over time, there is not one best performance indicator available, which is why we stick to the simple root-mean-square error here, providing an estimate of the accuracy.

To generate the true state, we start from the initial condition $\boldsymbol{x}_0 = (-5, -6, 22) + \boldsymbol{G}(\boldsymbol{0}, 1)$, where $\boldsymbol{G}(\mu, \sigma)$ denotes a vector of random Gaussian variables with mean μ and standard deviation σ. Similarly, we start the data assimilation algorithm by perturbing the true initial condition $\boldsymbol{x}_{0,\mathrm{DA}} = \boldsymbol{x}_0 + \boldsymbol{G}(\boldsymbol{0}, 1)$.

The observations are made by measuring all state variables directly, that is, $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{x}$ and $\boldsymbol{y} = (x_o, y_o, z_o)$. The x and y components of the Lorenz model are assumed to be Gaussian, such that their observed values are drawn from a Gaussian distribution around the true state $\boldsymbol{x}_t = (x_t, y_t, z_t)$ as $x_o = G(x_t, \sigma_x)$ and $y_o = G(y_t, \sigma_y)$. To decide which distribution to sample z_o from, we use the trained machine-learning model, with the true state as input. In that way, the chosen underlying distribution most closely follows the true distribution. Thus, the observed value of z is either $z_o = G(\mu_z^{(g)}, \sigma_z^{(g)})$, $z_o = L(\mu_z^{(l)}, \sigma_z^{(l)})$, or $z_o = R(\mu_z^{(r)}, \sigma_z^{(r)})$, with $L(\mu, \sigma)$ a random lognormal variable and $R(\mu, \sigma)$ a random reverse lognormal variable. The means and standard deviations are chosen in such a way that the true state z_t lies at the mode m of the distribution. For the Gaussian case, this is simply $\mu_z^{(g)} = z_t$ and $\sigma_z^{(g)} = \sigma_z$. In the case of lognormal errors, the mode of the distribution can be shown to be $m = e^{\mu-\sigma^2}$, while the variance is $\sigma_z^2 = e^{2\mu+\sigma^2}(e^{\sigma^2} - 1)$. Setting $m = z_t$, $\mu = \mu_z^{(l)}$, $\sigma = \sigma_z^{(l)}$, and rearranging the equations to find $\mu_z^{(l)}$and $\sigma_z^{(l)}$, one finds $\mu_z^{(l)} = \ln(z_t r)$ and $\sigma_z^{(l)} = \sqrt{\ln r}$, where is r is the single real root larger than 1 of the equation[1] $r^4 - r^3 - \sigma_z^2/z_t^2 = 0$. Similarly, for reverse lognormal errors, $\mu_z^{(r)} = \ln[(\xi_f - z_t)r]$ and $\sigma_z^{(r)} = \sqrt{\ln r}$, where is r is the single real root larger than 1 of the equation $r^4 - r^3 - \sigma_z^2/(\xi_f - z_t)^2 = 0$.

To test different cases, we vary two different variables: $\sigma \equiv \sigma_x = \sigma_y = \sigma_z$, the observation error of all variables (which also defines the observation-error covariance matrix as $R = \mathrm{diag}[\sigma^2, \sigma^2, \sigma^2]$) and the observation period τ_o, fixing the amount of time between observations. Moreover, six different Kalman filters are validated, based on the error distribution of the z component in the data assimilation method. The x and y components are always assumed to be Gaussian, while the z component is

a. Gaussian $\forall t$,
b. lognormal $\forall t$,
c. reverse lognormal $\forall t$,
d. Gaussian – lognormal,
e. Gaussian – reverse lognormal,
f. Gaussian – lognormal – reverse lognormal,

where, for the last three methods (d)–(f), the machine-learning decision function is used to switch dynamically between distributions. As we are interested in the performance of the mixed schemes, we compare the root-mean-square errors of the five mixed filters (b)–(f) with the all-Gaussian method. We use the fixed model-error covariance matrix derived by Evensen and Fario (1997) in all experiments. Note that in method (a) the forecast model is still considered to be nonlinear, in contrast to the original formulation of the Kalman filter (Jazwinski, 1970; Kalman, 1960), such that the effects of linearizing the model are avoided in comparing the different methods. For a comparison with the linearized (extended) Gaussian Kalman filter, the reader is referred to Fletcher *et al.* (2023).

In Figure 1, results are shown for the relative root-mean-square errors of the different Kalman filters, varying σ and τ_o. In Figure 1a, the root-mean-square error of the Gaussian Kalman filter is presented. As is expected, $\overline{\mathrm{RMSE}}$ grows when either σ or $\tau_o$increases. The Gaussian filter is used as a baseline to compare the other filters with, shown in Figure 1b–f according to the different distributions of z explained above. A two-sided F-test[2] is used to test the significance of the calculated root-mean-square errors relative to the Gaussian filter, with a null hypothesis that the root-mean-square errors are the same and a significance level of $\alpha = 10^{-4}$.

Overall, the mixed filters generally outperform the Gaussian filter, except for some specific cases. For small τ_o

[1] An analytic solution for r is available, but too lengthy to write down here.

[2] To use this test, we need to assume that absolute errors of the different filters are Gaussian with the same variance. This is not necessarily true, but, similarly to why we chose to compare root-mean-square errors, it gives an estimate of the performance of mixed Kalman filters.

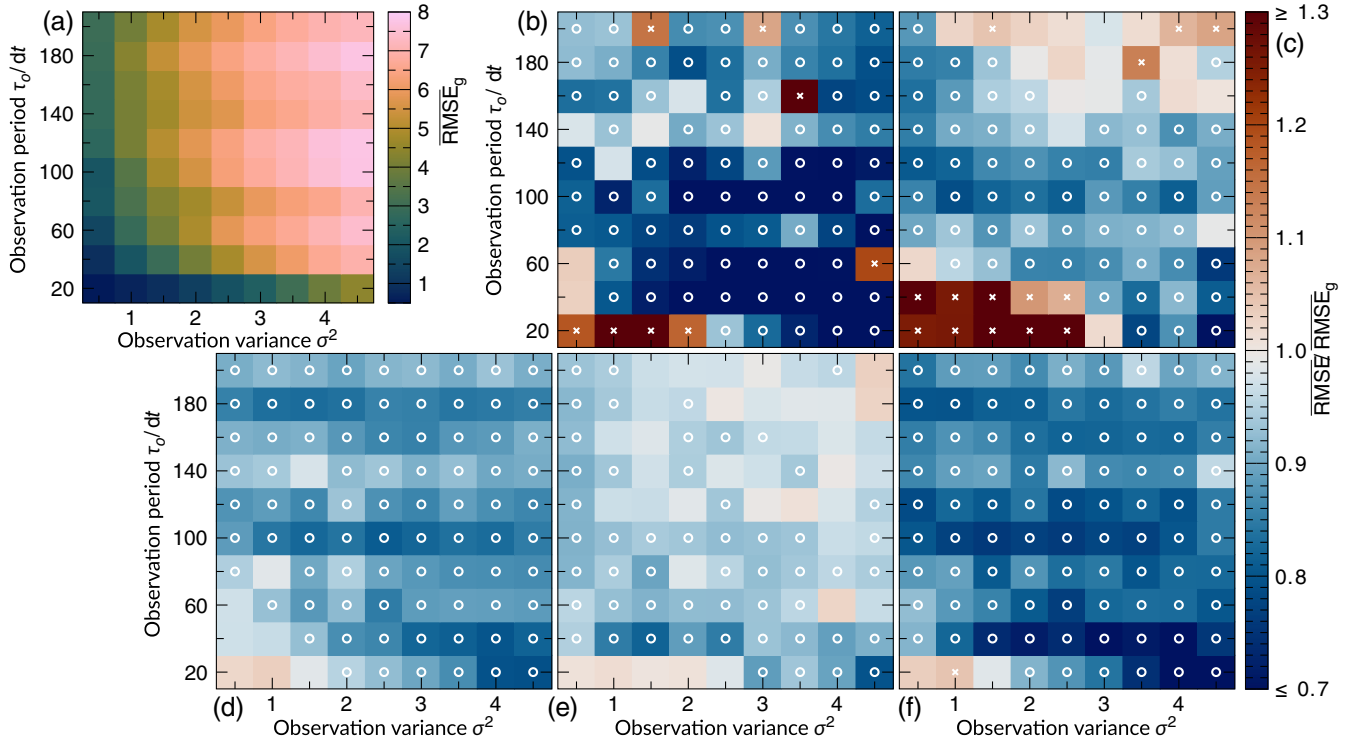

FIGURE 1 Root-mean-square error (RMSE) of different Kalman filters for different values of the observation error σ and the observation period τ_o for the Lorenz-63 model. In (a), the $\overline{\text{RMSE}}$ for the Gaussian Kalman filter is shown. The other panels show the $\overline{\text{RMSE}}$ relative to the Gaussian $\overline{\text{RMSE}}$ for different distributions of the z component: (b) lognormal $\forall t$; (c) reverse lognormal $\forall t$; (d) Gaussian – lognormal decided by machine-learning model; (e) Gaussian – reverse lognormal decided by machine-learning model; (f) Gaussian – lognormal – reverse lognormal decided by machine-learning model. The circles and crosses show the parameters that perform significantly better or worse than the Gaussian filter respectively, based on a two-sided F-test with $\alpha = 10^{-4}$. [Colour figure can be viewed at wileyonlinelibrary.com]

and σ, the lognormal (b) and reverse lognormal (c) filters have higher $\overline{\text{RMSE}}$, mainly because (reverse) lognormal signals in the background do not develop yet on these small timescales and a Gaussian approximation describes the evolution well. Moreover, for small observation errors the Gaussian Kalman filter performs very well, and not much improvement is possible. In isolated cases [see, for example, Figure 1b for $\tau_o = 160\ dt$ and $\sigma^2 = 3.5$], there is an instance where the filter failed to converge, resulting in a high $\overline{\text{RMSE}}$, as also reported in Fletcher *et al.* (2023).

Using the machine-learning model to switch between distributions (d)–(f) solves these problems by only using (reverse) lognormal updates when necessary. In all but one case, the dynamical Kalman filters (d)–(f) outperform the Gaussian filter, whenever the variation is significant. The dynamical filter switching between the three distributions (f) generally outperforms the filters where only two distributions are used, (d) and (e). It should be noted that the results are dependent on the choice of model-error covariance matrix, which we have taken to be the same for all filters. For the mixed filters, the model error dominates for small σ, resulting in large perturbations in the background state, which may affect the performance at small σ.

To show explicitly the improvement of using lognormal or reverse lognormal updates, Figure 2 presents an example of a specific twin experiment. There, we compare the fully Gaussian Kalman filter (a) with the fully dynamical Kalman filter (f). As before, the x and y components are always assumed to have Gaussian priors and likelihoods, while the z component switches between the three distributions considered. The observations are generated artificially from the true state and change distributions according to the machine-learning model applied on that true state. This can be seen in the bottom panel of Figure 2, where the filled circles show the distribution from which the observations are sampled. The circles are surrounded by an open circle, which shows the predicted distribution at the assimilation time, based on the observation itself. As can be seen, in three cases ($t = 2$, 2.4, and 7.2), the machine-learning model incorrectly predicts a Gaussian error, when the observation was actually generated from a lognormal or reverse lognormal distribution.

Still, the dynamical Kalman filter (red dashed lines) outperforms the Gaussian filter (blue dotted lines) in this specific scenario. In particular, after $t = 3.2$, both backgrounds start to diverge from the true solution. At $t = 3.6$, a

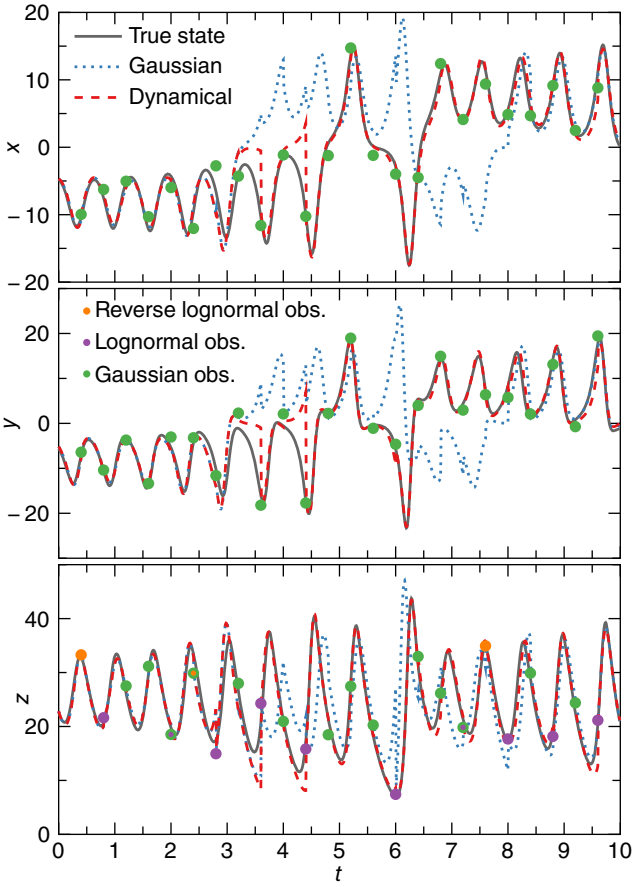

FIGURE 2 Example of a twin experiment with the dynamical Kalman filter with the Lorenz-63 model for $\sigma^2 = 3$ and $\tau_o = 40\ dt$. Grey lines show the true solution, blue dotted lines the background from the Gaussian Kalman filter, and red dashed lines the background from the fully mixed Kalman filter, with the machine-learning model selecting the underlying distribution. Filled circles show the artificial observations, sampled from a distribution selected by the machine-learning model. Open circles show the predicted distribution at the analysis time, which does not always match the correct underlying error distribution. [Colour figure can be viewed at wileyonlinelibrary.com]

lognormal observation is available, which is predicted correctly by the machine-learning model and brings the background back towards the true state. However, the Gaussian filter does not use the correct underlying statistics, such that it is unable to bring the background back towards the truth. The same phenomenon can be seen at later times, where the dynamical filter is able to assimilate lognormal errors correctly, while the Gaussian filter diverges from the true state. Only when enough Gaussian observations are available does the Gaussian filter converge back to the true solution.

To elucidate the improvements of the non-Gaussian filters further, we also perform experiments where only the x and y components of the Lorenz-63 model are observed, that is, $\boldsymbol{h}(\boldsymbol{x}) = (x, y)$. To also test the robustness with respect to specific conditions of the machine-learning model, we use a significance level of $\alpha = 0.2$ for the skewness test (rejecting the null hypothesis if the z score is outside the domain $[-1.28, 1.28]$) and take a sample window size of $w = 12$. The rest of the experimental details are identical to the experiments outlined above.

The results are shown in Figure 3 for different values of τ_o and σ. As can be seen, the dynamical Kalman filter (solid brown bars) always outperforms the Gaussian Kalman filter (hatched red bars), with particularly significant improvement at larger observation variances.

4.3 | Coupled Lorenz system

Because the Lorenz-63 system is very simple, and only has a single variable that seems to follow non-Gaussian statistics, it is not possible to use it to assess the full strength of the mixed Kalman filter, where all three distributions are present at a single assimilation time. To overcome this, we build a higher-dimensional system by creating a chain of Lorenz-63 systems, with the possibility of coupling between the different attractors, similar to Akhmet and Fen (2015). These coupled differential equations then have the form

$$\begin{cases} \frac{dx_\ell}{dt} = \sigma(y_\ell - x_\ell) + c_x x_{\ell+1}, \\ \frac{dy_\ell}{dt} = \rho x_\ell - y_\ell - x_\ell z_\ell + c_y y_{\ell+1}, \\ \frac{dz_\ell}{dt} = x_\ell y_\ell - \beta z_\ell + c_z z_{\ell+1}, \end{cases} \tag{30}$$

with $\ell \in \{1, \ldots, n_p\}$ an index selecting the attractor and $n_p$the number of attractors. Moreover, we assume the total system to be cyclic, such that $x_{n_p+1} = x_1$, $y_{n_p+1} = y_1$, and $z_{n_p+1} = z_1$. The state vector $\boldsymbol{x}$ has $N = 3n_p$ elements, three variables for each attractor, and Equation 30 expresses a system of N coupled differential equations. We again choose $(\sigma, \rho, \beta) = (10, 28, 8/3)$, and perform experiments for $n_p = 10$ and weak coupling $c \equiv c_x = c_y = c_z = 0.1$.

The experiments performed with the coupled Lorenz system are very similar to the ones for the three-variable Lorenz system described above. Every separate Lorenz attractor in the chain starts from different initial conditions, chosen to be $x_{\ell,0} = -5 + G(0, 3)$, $y_{\ell,0} = -6 + G(0, 3)$, and $z_{\ell,0} = 22 + G(0, 3)$. The data assimilation run is initiated with $\boldsymbol{x}_{0,\mathrm{DA}} = \boldsymbol{x}_0 + \boldsymbol{G}(\boldsymbol{0}, 1)$.

A decision function is again defined through a machine-learning model based on the k-nearest-neighbour classifier. Because the different attractors have the same general behaviour, the machine-learning model can be trained on only one location $\ell = 1$ and

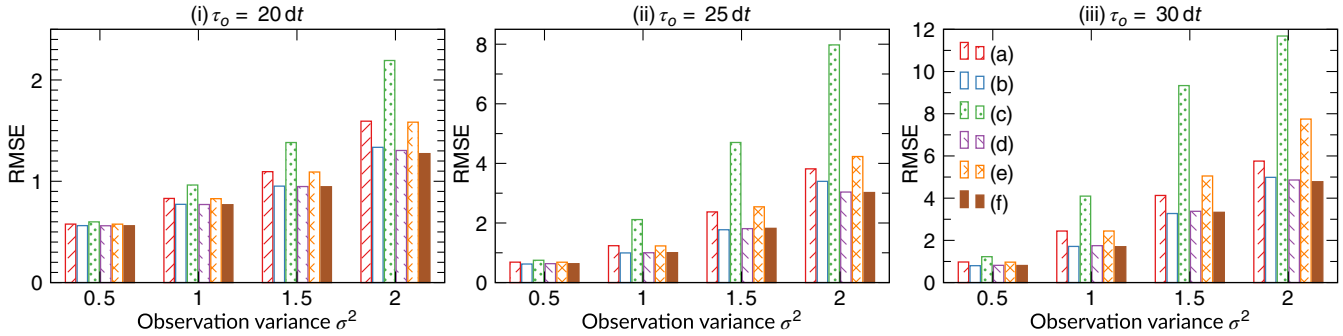

FIGURE 3 Root-mean-square errors of different Kalman filters for different values of the observation variance σ^2 and the observation period τ_o for the Lorenz-63 model, observing the x and y components only. The different bars depict different choices for the prior distribution of the z component: (a) Gaussian $\forall t$; (b) lognormal $\forall t$; (c) reverse lognormal $\forall t$; (d) Gaussian – lognormal decided by machine-learning model; (e) Gaussian – reverse lognormal decided by machine-learning model; (f) Gaussian – lognormal – reverse lognormal decided by machine-learning model. [Colour figure can be viewed at wileyonlinelibrary.com]

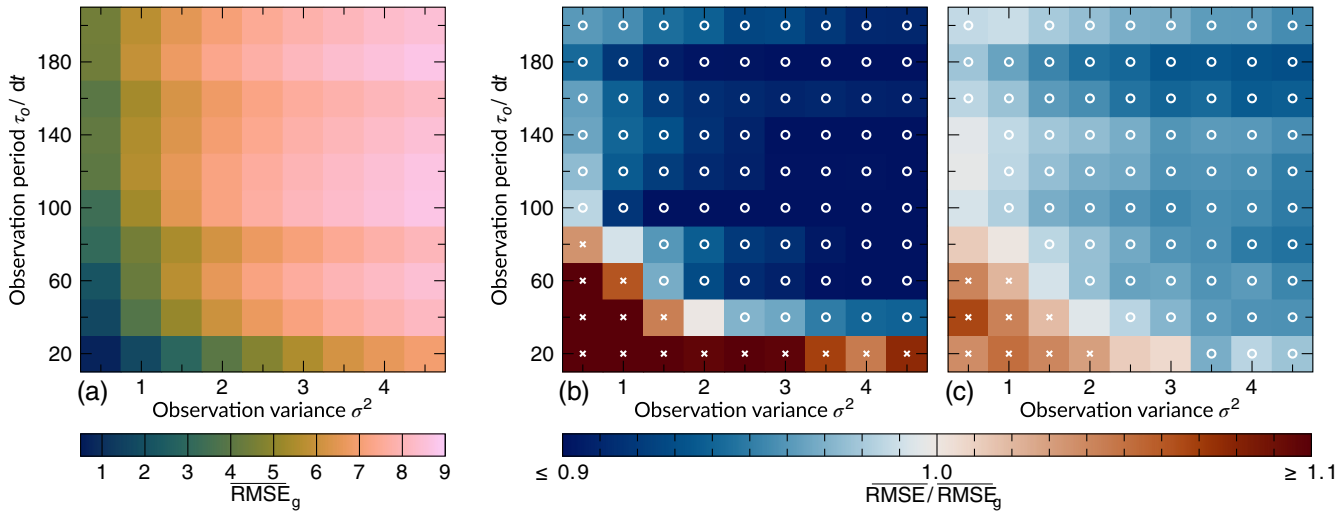

FIGURE 4 Root-mean-square error of different Kalman filters for different values of the observation error σ and the observation period τ_o for the coupled Lorenz-63 model with $n_p = 10$ and $c = 0.1$. In (a), the root-mean-square error for the Gaussian Kalman filter is shown. The other panels show the root-mean-square error relative to the Gaussian root-mean-square error for different distributions of the z_ℓ components: (b) lognormal $\{z_1, \ldots, z_5\}$ and reverse lognormal $\{z_6, \ldots, z_{10}\}$ $\forall t$; (c) Gaussian – lognormal – reverse lognormal decided by machine-learning model. The circles and crosses show the parameters that perform significantly better or worse than the Gaussian filter, respectively, based on a two-sided F-test with $\alpha = 10^{-4}$. [Colour figure can be viewed at wileyonlinelibrary.com]

used for all locations. Thus, the machine-learning model is trained on x_1 and y_1 as input and the skewness of z_1 as output, with the same parameters as before, i.e. time $t \in [0, 1000]$, time step $dt = 0.01$, skewness sample window $w = 14$, significance level $\alpha \simeq 0.3$, neighbours $k = 15$, and inverse-distance weights. The model predicts the correct skewness 97.7% of the time, tested on a randomized 30% sample of the original control run. The resulting decision function is used both to generate artificial observations [by deciding the distribution to sample $\boldsymbol{y}_{z_\ell}$ from, with $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{x}$] and to select the distribution to use at each assimilation time in the mixed Kalman filter.

We again compute the average root-mean-square error $\overline{\text{RMSE}}$ for $N_c = 50$ trials and $N_a = 250$ assimilation windows for different observation errors σ (assuming all state variables have the same standard deviation) and observation periods τ_o. We always assume x_ℓ and y_ℓ to be Gaussian, and test three different scenarios for the distributions of z_ℓ:

a. Gaussian $\{z_1, \ldots, z_{10}\}$ $\forall t$;
b. lognormal $\{z_1, \ldots, z_5\}$ and reverse lognormal $\{z_6, \ldots, z_{10}\}$ $\forall t$;
c. Gaussian – lognormal – reverse lognormal decided by machine-learning model.

The all-Gaussian method is used as a baseline to compare the other methods with. In scenario (b), we seemingly arbitrarily chose the first five attractors to behave lognormally and the last five to behave reverse lognormally. We have tested different variations of selecting five lognormal and five reverse lognormal $z_\ell$components, with similar results, so the conclusions are robust to this choice. In scenario (c), only information of x_ℓ and y_ℓ is needed to predict the distribution of z_ℓ, and thus the attractors are assumed to be independent when making this decision.

The results for the coupled Lorenz-63 model are summarized in Figure 4, where in Figure 4a the root-mean-square error of the Gaussian filter is shown; this is used as a comparison for the mixed filters in Figure 4b,c. Again, a two-sided F-test is used to investigate the significance of the variations between the Gaussian and mixed filters, with a significance level of $\alpha = 10^{-4}$. The mixed Kalman filters only perform worse than the Gaussian filter for small τ_0 and σ. The main reason for the decreased performance of the dynamical filter can be attributed to the machine-learning model, as it has not been optimized for the coupled system and may therefore predict incorrect distributions. Moreover, the model error remains fixed for the entire data assimilation run and dominates the errors at small values of σ. For larger τ_0 and σ, the mixed Kalman filters always perform better than the Gaussian one.

5 | CONCLUSION

We have derived a dynamical mixed Kalman filter that accounts for background and observation errors that originated from a mixed distribution of Gaussian, lognormal, and reverse lognormal random variables. This generalizes the earlier work of (Fletcher *et al.*, 2023), where the lognormal Kalman filter was introduced, now allowing for variables with negative skewness. An important aspect of the mixed Kalman filter is that it makes use of the nonlinear model, therefore eliminating any errors introduced by linearizing the model.

To test the performance of the dynamical mixed Kalman filter, we performed twin experiments with two different toy systems based on the Lorenz-63 model. If the time between observations is long and their errors are large, the dynamical mixed Kalman filter significantly improves the analysis skill over an all-Gaussian approach in the parameter space that was examined.

Our work makes an important step towards creating operationally viable data assimilation systems with skewed state variables that have upper and/or lower bounds. Specifically, the mixed Kalman filter equations can be used to develop non-Gaussian ensemble methods, for example by extending the maximum-likelihood ensemble filter (Zupanski, 2005) to accommodate non-Gaussian errors. In conjunction with similar non-Gaussian variational methods that use a mixed distribution of Gaussian, lognormal, and reverse lognormal random variables (Goodliff *et al.*, 2023), this moreover creates the possibility to design hybrid non-Gaussian data assimilation techniques in the future.

AUTHOR CONTRIBUTIONS

Senne Van Loon: formal analysis; investigation; methodology; software; visualization; writing – original draft; writing – review and editing. **Steven J. Fletcher:** funding acquisition; supervision; writing – review and editing.

ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation grant AGS-2033405 at the Cooperative Institute for Research in the Atmosphere (CIRA), Colorado State University (CSU). We thank two anonymous reviewers for constructive and helpful comments on the article.

CONFLICT OF INTEREST STATEMENT

The authors have no conflict of interest to declare.

DATA AVAILABILITY STATEMENT

The code used in this article is available at https://github.com/SnnVL/DynamicalGaussianLognormalReverseLognormalKalmanfilter.

ORCID

Senne Van Loon https://orcid.org/0000-0002-7868-3943
Steven J. Fletcher https://orcid.org/0000-0002-1662-7460

REFERENCES

Akhmet, M. & Fen, M.O. (2015) Extension of Lorenz unpredictability. *International Journal of Bifurcation and Chaos*, 25, 1550126.

Amezcua, J. & Van Leeuwen, P.J. (2014) Gaussian anamorphosis in the analysis step of the EnKF: a joint state-variable/observation approach. *Tellus A: Dynamic Meteorology and Oceanography*, 66, 23493.

Bannister, R. (2017) A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143, 607–633.

Bannister, R.N., Chipilski, H.G. & Martinez-Alvarado, O. (2020) Techniques and challenges in the assimilation of atmospheric water observations for numerical weather prediction towards convective scales. *Quarterly Journal of the Royal Meteorological Society*, 146, 1–48 https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3652

Bertino, L., Evensen, G. & Wackernagel, H. (2003) Sequential data assimilation techniques in oceanography. *International Statistical Review*, 71, 223–241.

Bishop, C.H. (2016) The GIGG-EnKF: ensemble Kalman filtering for highly skewed non-negative uncertainty distributions. *Quarterly Journal of the Royal Meteorological Society*, 142, 1395–1412.

Bocquet, M., Pires, C.A. & Wu, L. (2010) Beyond Gaussian statistical modeling in geophysical data assimilation. *Monthly Weather Review*, 138, 2997–3023.

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. & Fario, N. (1997) Solving for the generalized inverse of the Lorenz model. *Journal of the Meteorological Society of Japan Series II*, 75, 229–243.

Fletcher, S.J. (2010) Mixed Gaussian-lognormal four-dimensional data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 62, 266–287.

Fletcher, S.J. (2022) *Data assimilation for the geosciences: from theory to application*, 2nd edition. Amsterdam, Netherlands: Elsevier.

Fletcher, S.J. & Jones, A.S. (2014) Multiplicative and additive incremental variational data assimilation for mixed lognormal–Gaussian errors. *Monthly Weather Review*, 142, 2521–2544.

Fletcher, S.J., Kliewer, A.J. & Jones, A.S. (2019) Quantification of optimal choices of parameters in lognormal Variational data assimilation and their chaotic behavior. *Mathematical Geosciences*, 51, 187–207.

Fletcher, S.J. & Zupanski, M. (2006a) A data assimilation method for log-normally distributed observational errors. *Quarterly Journal of the Royal Meteorological Society: A Journal of the Atmospheric Sciences, Applied Meteorology and Physical Oceanography*, 132, 2505–2519.

Fletcher, S.J. & Zupanski, M. (2006b) A hybrid multivariate normal and lognormal distribution for data assimilation. *Atmospheric Science Letters*, 7, 43–46.

Fletcher, S.J. & Zupanski, M. (2007) Implications and impacts of transforming lognormal variables into normal variables in VAR. *Meteorologische Zeitschrift*, 16, 755–765. Available from: https://doi.org/10.1127/0941-2948/2007/0243

Fletcher, S.J., Zupanski, M., Goodliff, M.R., Kliewer, A.J., Jones, A.S., Forsythe, J.M. *et al.* (2023) Lognormal and mixed Gaussian-lognormal Kalman filters. *Monthly Weather Review*, 151, 761–774.

Foster, J., Bevis, M. & Raymond, W. (2006) Precipitable water and the lognormal distribution. *Journal of Geophysical Research: Atmospheres*, 111, D15102. https://doi.org/10.1029/2005JD006731

Goodliff, M., Fletcher, S., Kliewer, A., Forsythe, J. & Jones, A. (2020) Detection of non-Gaussian behavior using machine learning techniques: a case study on the Lorenz 63 model. *Journal of Geophysical Research: Atmospheres*, 125, e2019JD031551. Available from: https://doi.org/10.1029/2019JD031551

Goodliff, M.R., Fletcher, S.J., Kliewer, A.J., Jones, A.S. & Forsythe, J.M. (2022) Non-Gaussian detection using machine learning with data assimilation applications. *Earth and Space Science*, 9, e2021EA001908.

Goodliff, M.R., Hossen, M.J., Van Loon, S. & Fletcher, S.J. (2023) Reverse Lognormal Variational Data Assimilation. *In review*.

Hu, G., Dance, S.L., Bannister, R.N., Chipilski, H.G., Guillet, O., Macpherson, B. *et al.* (2023) Progress, challenges, and future steps in data assimilation for convection-permitting numerical weather prediction: report on the virtual meeting held on 10 and 12 November 2021. *Atmospheric Science Letters*, 24, e1130 https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/asl.1130

Jazwinski, A.H. (1970) *Stochastic processes and filtering theory*. New York: Academic Press.

Kalman, R.E. (1960) A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82, 35–45.

Kliewer, A.J., Fletcher, S.J., Jones, A.S. & Forsythe, J.M. (2016) Comparison of Gaussian, logarithmic transform and mixed Gaussian-log-normal distribution based 1DVAR microwave temperature-water-vapour mixing ratio retrievals. *Quarterly Journal of the Royal Meteorological Society*, 142, 274–286.

Lorenz, E.N. (1963) Deterministic nonperiodic flow. *Journal of Atmospheric Sciences*, 20, 130–141.

Nerger, L. (2022) Data assimilation for nonlinear systems with a hybrid nonlinear Kalman ensemble transform filter. *Quarterly Journal of the Royal Meteorological Society*, 148, 620–640.

Parrish, D.F. & Derber, J.C. (1992) The National Meteorological Center's spectral statistical-interpolation analysis system. *Monthly Weather Review*, 120, 1747–1763.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O. *et al.* (2011) Scikit-learn: machine learning in python. *Journal of Machine Learning Research*, 12, 2825–2830.

Perron, M. & Sura, P. (2013) Climatology of non-Gaussian atmospheric statistics. *Journal of Climate*, 26, 1063–1083.

Poterjoy, J. (2022) Implications of multivariate non-Gaussian data assimilation for multiscale weather prediction. *Monthly Weather Review*, 150, 1475–1493 https://journals.ametsoc.org/view/journals/mwre/150/6/MWR-D-21-0228.1.xml

Simon, E. & Bertino, L. (2009) Application of the Gaussian anamorphosis to assimilation in a 3-D coupled physical-ecosystem model of the North Atlantic with the EnKF: a twin experiment. *Ocean Science*, 5, 495–510.

Simon, E. & Bertino, L. (2012) Gaussian anamorphosis extension of the DEnKF for combined state parameter estimation: application to a 1D ocean ecosystem model. *Journal of Marine Systems*, 89, 1–18.

Snyder, C., Bengtsson, T. & Morzfeld, M. (2015) Performance bounds for particle filters using the optimal proposal. *Monthly Weather Review*, 143, 4750–4761.

van Leeuwen, P.J. (2009) Particle filtering in geophysical systems. *Monthly Weather Review*, 137, 4089–4114.

Zupanski, M. (2005) Maximum likelihood ensemble filter: theoretical aspects. *Monthly Weather Review*, 133, 1710–1726.

How to cite this article: Van Loon, S. & Fletcher, S.J. (2023) A dynamical Gaussian, lognormal, and reverse lognormal Kalman filter. *Quarterly Journal of the Royal Meteorological Society*, 1–13. Available from: https://doi.org/10.1002/qj.4595