Bayesian optimization for parameter estimation of a local particle filter

Shoichi AKAMI[1]

Graduate School of Life and Environmental Sciences, University of Tsukuba, Tsukuba, Japan
Meteorological Research Institute, Japan Meteorological Agency, Tsukuba, Japan

and

Keiichi KONDO

Meteorological Research Institute, Japan Meteorological Agency, Tsukuba, Japan

and

Hiroshi L. TANAKA

Organization of Volcanic Disaster Mitigation, Shinjuku, Japan

and

Mizuo KAJINO

Meteorological Research Institute, Japan Meteorological Agency, Tsukuba, Japan
Faculty of Life and Environmental Sciences, University of Tsukuba, Tsukuba, Japan

May 17th, 2025

---

1) Corresponding author: Shoichi AKAMI, Graduate School of Life and Environmental Sciences, University of Tsukuba, 1-1, Tennoudai, Tsukuba, Ibaraki, Japan, 305-8572.
Email: akami@mri-jma.go.jp
Tel: +81-29-853-8695

Abstract

The Particle filter (PF) is a powerful data assimilation method that does not assume the linearity in the time evolution of errors or Gaussian error distributions. However, the number of particles required increases exponentially with the dimensions of the dynamical system, which is a bottleneck when applying the PF to numerical weather prediction. Local particle filter (LPF) realizes the PF in high-dimensional systems by the localization, but it has high parameter sensitivity and is challenging to operate stably. On the other hand, when using a strong nonlinear observation operator, it is possible to estimate the analysis with higher accuracy than the local ensemble transform Kalman filter by setting the inflation factor τ and the localization scale r to the optima. Therefore, an efficient parameter estimation method is required.

Bayesian optimization (BO) is a method for efficiently solving optimization problems of black box functions with high computational costs, and is used for parameter optimization of neural networks. Therefore, we estimated τ and r that minimize the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in the LPF using the BO in the Lorenz-96 40-variable model. As a result, the BO estimated τ and r with higher accuracy than random sampling and was robust to changes in the observations to a certain extent. In addition, it was important to adopt the kernel functions and the acquisition functions tailored to the characteristics of the problem to improve the estimation accuracy of the BO.

This study clarified that the BO contributes to improving the practicality of the LPF and

suggested what approach should be adopted when the number of estimated parameters increases. By developing this technology, the prediction accuracy of heavy rainfall is expected to improve in the future. The usefulness of the BO will eventually be proven in atmospheric model experiments aimed at the practical application of the LPF.

Keywords: Local particle filter; Parameter estimation; Bayesian optimization; Gaussian process regression

1. Introduction

In chaotic dynamical systems such as numerical weather prediction (NWP) models, even small errors in the initial conditions can develop over time and become large errors. Data assimilation is a technique for estimating the analysis closer to the truth from the forecasts and the observations, and by using the high-precision analyses as the initial conditions, forecast errors can be improved. The ensemble Kalman filter (EnKF; Evensen, 1994) and 4D-Var (Dimet and Talagrand, 1986), which are currently the mainstream data assimilation methods, can estimate the optimum analysis when the errors develop linearly over time and the error distribution follows a Gaussian distribution. However, when these assumptions are not satisfied—around cumulus convection and storm tracks—it cannot estimate the optimum analyses (Kondo and Miyoshi, 2019).

On the other hand, the particle filter (PF; Gordon et al., 1993) does not assume linearity or Gaussianity, and therefore, it can be an appropriate data assimilation method for dynamical systems with strong nonlinearity. However, the PF estimates the analyses by resampling ensemble members (particles) based on weights obtained from the likelihood of observations, and therefore, “weight collapse” may occur in high-dimensional systems. The PF requires an exponential increase in the number of particles necessary for the dimensions of the dynamical system (Snyder et al., 2008), and this problem is a bottleneck when applying the PF to the NWP.

Local particle filter (LPF; Penny and Miyoshi, 2016) achieves the PF in high-dimensional

systems by reducing the dimensions of observations through localization. Spatial localization is justified because distant correlations are spurious or weak compared to nearby correlations. If well applied, the LPF can estimate a more accurate analysis than the EnKF with non-Gaussian observation errors, nonlinear observation operators, and sparse observation networks (Poterjoy and Anderson, 2016; Poterjoy, 2016; Penny and Miyoshi, 2016).

However, the localization scale and inflation factor—smoothing weights among particles—are the parameters should be optimized in the LPF. In addition, because excessive resampling causes “weight collapse,” adjusting the resampling frequency based on an effective sample size is critical. Furthermore, a method for implementing the PF, which approximates the prior distribution using a combination of Gaussian kernels centered at the value of each particle, has been suggested. In this approach, the amplitude of the Gaussian kernel is a parameter that should be optimized (Stordal et al., 2011). If the LPF does not optimize these parameters, it will diverge (Kotsuki et al., 2022).

The parameters to be optimized and the computational cost of data assimilation experiments are expected to increase with the improvement of LPF methods and the advancement of systems for applying the LPF to the operational NWP. The simplest way to optimize the parameters is using the grid search (also known as manual tuning or brute-force). However, this method requires data assimilation experiments that increase exponentially with the number of parameters. In addition, random sampling (RS) is a method

that works more efficiently than grid search in high-dimensional spaces. Still, this method may be unable to explore the optimum if the number of samples is insufficient. Therefore, an efficient optimization method is needed.

One way to reduce computational cost is to replace the system response to the parameters with a surrogate model (e.g., Sawada, 2020). Bayesian optimization (BO; Mockus, 1989) is a method for estimating the parameters that minimize (or maximize) an objective function and is used for the parameter optimization of neural networks (Snoek et al., 2012). As the BO uses Gaussian process regression (GPR) to emulate the objective function, it can efficiently explore a globally optimal parameter even when the shape of the response surface for the input and output data is unknown or when the function is a multi-peaked function that cannot be differentiated. In addition, it is easy to implement because the BO works independently of other systems.

The effectiveness of using the BO within the EnKF framework has already been demonstrated (Lunderman et al., 2021), so we continued this line of study to investigate whether the BO can improve the practicality of LPF. In addition, since the BO has been used as a tool in previous studies, we verified how the estimation accuracy of the BO changes with the increase in the dimension of response surfaces and changes in settings of the BO, with a view to future technological developments. This study was conducted using a data assimilation experiment with the Lorenz-96 40-variable model (L96: Lorenz and Emanuel, 1998).

This paper is organized as follows: Section 2 introduces the methodology, while Section 3 describes the experimental setup. In Section 4, we compared the estimation accuracy of the RS and the BO. In addition, we investigated the estimation results of the BO in detail from the perspective of the GPR prediction distribution. Section 5 presents future prospect and conclusion.

2. Method

a. Local particle filter

The PF estimates the posterior distribution using the Monte Carlo method and Bayes' theorem:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)\, p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}, \tag{1}$$

where p represents the probability distribution; $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ denotes the posterior distribution of state variable $\boldsymbol{x}$ at time step t $(t = 1, \dots, T)$ given all observations $\boldsymbol{y}$ up to time t; $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ is the likelihood of $\boldsymbol{x}$ given $\boldsymbol{y}$; $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ is the prior distribution given all $\boldsymbol{y}$ up to one time step before analysis time step; and $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:y-1})$ denotes the marginal likelihood of $\boldsymbol{y}$, which can be expressed as a constant computed by climate data in the NWP. The prior distribution can be approximated using particles (or ensemble members) of the numerical forecast:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) \approx \frac{1}{M}\sum_{m=1}^{M} \delta\big(x_t - \mathcal{F}(x_{t-1}^m)\big), \tag{2}$$

where the subscripts m $(m = 1, \dots, M)$ denote the indices of the particle, δ is the Dirac delta function, and $\mathcal{F}$ is the numerical model. In this study assumes a Gaussian likelihood function, given by

$$p(\boldsymbol{y}_t|\boldsymbol{x}_t) = \frac{1}{\sqrt{(2\pi)^o|\boldsymbol{R}|}} \exp\left[-\frac{1}{2}\big(\boldsymbol{y}_t - h(\boldsymbol{x}_t)\big)^{\mathsf{T}}\boldsymbol{R}^{-1}\big(\boldsymbol{y}_t - h(\boldsymbol{x}_t)\big)\right]. \tag{3}$$

where o represents the dimension of $\boldsymbol{y}$. In addition, $\boldsymbol{R}$ denotes the observation error covariance matrix, and $|\boldsymbol{R}|$ is its determinant. h denotes the observation operator. The weight of each particle is the normalized likelihood, computed for all particles as follows:

$$w_t^m = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t^m)}{\sum_{m'=1}^{M} p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^{m'}\right)}, \tag{4}$$

where the subscripts m' denote the indices of the particles for summation. The posterior distribution is obtained by resampling each particle of the prior distribution in proportion to its weight:

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) \approx \sum_{m=1}^{M} w_t^m \delta\left(x_t - \mathcal{F}(x_{t-1}^m)\right). \tag{5}$$

The resampling method is also arbitrary. This study defined the analysis particles as the sum of the transformation for perturbations of forecast particles and the mean of the forecast particles:

$$\boldsymbol{X}^a = \overline{\boldsymbol{X}}^f + \delta \boldsymbol{X}^f \boldsymbol{T}, \tag{6}$$

where $\boldsymbol{X}^a$ denotes the analysis particles; $\overline{\boldsymbol{X}}^f$ represent the mean of forecast particles; and $\delta \boldsymbol{X}^f$ denotes the perturbation of forecast particles, where the row and column of $\boldsymbol{X}^a$, $\overline{\boldsymbol{X}}^f$, and $\delta \boldsymbol{X}^f$ indicate the particle size and dimension of the numerical model, respectively. $\boldsymbol{T}$ denotes the ensemble transform matrix, defined as a square matrix of order M. As resampling is performed using the ensemble transform matrix in the LPF, the matrix markedly affects filter performance (Farchi and Bocquet, 2018; Kotsuki et al., 2022). When the particle size is sufficiently large, the ratio of resampled particle sizes will closely match the ratio of weights; otherwise, the sampling error may become substantial.

In addition, the weights among grid points differ because varying observations are assimilated at each grid point through localization. As the pronounced weight difference

causes spatial discontinuity, the ensemble transform matrix should satisfy a spatially smooth transition. Addressing the smoothing issue presents an interesting challenge. For example, Kotsuki et al. (2022) addressed this problem by sorting the particles and creating an ensemble transform matrix close to an identity matrix (see also Potthast et al., 2019). Our resampling method is based on Algorithm 1 of Kotsuki et al. (2022) and uses stochastic universal resampling (SUR) instead of probabilistic resampling to reduce sampling error. The SUR is implemented as follows. Create a normalized cumulative probability distribution divided by the weight of each particle, and select a random starting point in the range $[0, 1/M]$. Set M pointers at equal intervals between the starting point and $1/M$, and sample the particles corresponding to the cumulative probabilities pointed to by each pointer.

Furthermore, we used localization to limit the impact of observations within the local domain to avoid "weight collapse" (Penny and Miyoshi, 2016; Kotsuki et al., 2022). This localization method is applied by independently computing the analysis at every grid point, similar to the local ensemble transform Kalman filter (LETKF; Hunt et al., 2007). Specifically, it is implemented by computing the product of the inverse of observation error covariance matrix $\boldsymbol{R}$ in Eq. (3) and the inverse of localization function $\boldsymbol{L}(r)$:

$$\exp\left[-\frac{1}{2}\left(\boldsymbol{y}_t - h(\boldsymbol{x}_t)\right)^{\top}\boldsymbol{R}^{-1}\{\boldsymbol{L}(r)\}^{-1}\left(\boldsymbol{y}_t - h(\boldsymbol{x}_t)\right)\right]. \tag{7}$$

Here, the localization function approximates a Gaussian function (Gaspari and Cohn, 1999):

$$L(r) = \begin{cases} \exp\left(-\frac{q^2}{2r^2}\right) \text{ if } q < 2\sqrt{\frac{10}{3}}\, r \\ 0 \text{ else} \end{cases}, \tag{8}$$

where q denotes the distance between the analysis grid point and the observation point and r represents the standard deviation of the Gaussian function, defining the localization scale. Observations beyond this scale, including its boundary, are not assimilated, while those within the localization scale are weighted based on the localization function. Therefore, r is the parameter that determines the localization scale, and it is necessary to set the appropriate value.

In addition, to avoid filter divergence, it is necessary to maintain particle diversity. Therefore, we smoothed the weights among particles to prevent a few particles from occupying most of the weights. We refer to this approach as inflation in this study:

$$w_t^m \leftarrow \tau w_t^m + \frac{1-\tau}{M}, (0 \leq \tau \leq 1), \quad (9)$$

where τ represents the inflation factor. If τ is not 1, the weights w_t^m are smoothed, and all particles have equal weights when τ equals 0. On the other hand, if the original weights are used, the LPF tends to diverge due to "weight collapse." As τ becomes smaller, the LPF deviates from the PF but becomes more stable. Thus, the relationship between mathematical rigor and stability is a trade-off on the inflation factor τ. Note that this approach is mathematically equivalent to Eq. (23) in Kotsuki et al. (2022). However, while Kotsuki et al. (2022) smoothed the weights in the time direction, we smoothed the weights among particles.

b. Bayesian optimization

The BO estimates input data that minimizes the objective function by modeling response surface using the GPR and evaluating using an acquisition function. The GPR assumes that a joint distribution $p(\boldsymbol{g})$ of input data $\boldsymbol{z} = \{z_1, z_2, \dots, z_S\}$ and corresponding output data $\boldsymbol{g} = \{g(z_1), g(z_2), \dots, g(z_S)\}$ follow the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$. This assumption is written as follows:

$$\boldsymbol{g} \sim \mathcal{GP}\big(\boldsymbol{\mu}(\boldsymbol{z}), \boldsymbol{K}(\boldsymbol{z}, \boldsymbol{z}')\big), \tag{10}$$

where $\boldsymbol{z}$ are input data that summarizes the inflation factor τ and the localization scale r into a single vector. The subscripts s $(s = 1, \dots, S)$ denote the indices of the data and the superscript $'$ denotes the another data within the data set. In addition, $\mathcal{GP}$ denotes the Gaussian process with the mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{K}$ defined as a square matrix of order S. The elements of covariance matrix $\boldsymbol{K_{ij}}$ is defined as $k(z, z' \mid \boldsymbol{\theta})$. In this study, we used the Gaussian kernel with added white noise as a general choice:

$$k(z, z' \mid \boldsymbol{\theta}) = \theta_1 \exp\left(-\frac{(\tau - \tau')^2}{\theta_2} - \frac{(r - r')^2}{\theta_3}\right) + \theta_4 \delta(z, z'). \tag{11}$$

Here, the kernel function $k(z, z')$ defines the correlation between any two data z and z' in the input data $\boldsymbol{z}$. In addition, $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4)$ denotes the positive hyper-parameters that define the kernel function, while δ represents the Dirac delta function.

When the amplitude parameter θ_1 is small, the variation in the GPR prediction distribution is slight. The GPR prediction distribution becomes smoother when the length scale parameters θ_2 and θ_3 are large. When the noise parameter θ_4 is small, the uncertainties in the GPR prediction distribution near the input data are reduced. Note that when there are

two types of input data, using two length scale parameters, θ_2 and θ_3, allows for more flexible modeling tailored to the characteristics of each input data.

In addition, since τ and r have different scales by a factor of 10, we normalized them to the same scale. Since the Gaussian kernel performs distance-based calculations, the normalization prevents the influence of specific input data from becoming dominant. In our system, this approach markedly contributed to improving the performance of the BO.

When new input data $\boldsymbol{z}^*$ is given, the GPR is updated, and the new joint distribution of output data $\boldsymbol{g}^*$ is expressed as:

$$p(\boldsymbol{g}^*|\boldsymbol{z}^*, \mathcal{D}) = \mathcal{N}(\boldsymbol{k}_*^\top \boldsymbol{K}^{-1}\boldsymbol{g}, k_{**} - \boldsymbol{k}_*^\top \boldsymbol{K}^{-1}\boldsymbol{k}_*)\,, \tag{12}$$

where $\mathcal{D} = (\boldsymbol{z}, \boldsymbol{g})$ denotes the accumulated input data, $\boldsymbol{k}_*$ is the similarity between the new input data $\boldsymbol{z}^*$ and the accumulated input data $\mathcal{D}$. k_{**} represents the similarity of the new input data z^* to themselves.

$$\boldsymbol{k}_* = \big(k(z^*, z_1), k(z^*, z_2), \dots, k(z^*, z_S)\big)^\top, \tag{13}$$

$$k_{**} = k(z^*, z^*). \tag{14}$$

Eq. (12), (15), and (16) are derived under the assumption that $\boldsymbol{\mu}(\boldsymbol{z})$ in Eq. (10) is zero, but in practice, mathematical rigor can be achieved by subtracting the average from the input data.

In addition, when the covariance matrix $\boldsymbol{K}$ becomes close to a singular matrix due to redundant exploration of the same input data, it may become impossible to calculate the inverse matrix stably (Rasmussen and Nickisch, 2010). There are several techniques to

improve numerical stability, but we followed Rasmussen and Williams (2006) and added jitter to the diagonal components of the covariance matrix. However, as far as we have experimented, this technique alone can prevent errors associated with singular matrices, but cannot prevent the redundant exploration. Therefore, we adopted the penalized expected improvement (EI) described below.

The hyper-parameters $\boldsymbol{\theta}$ are optimized by maximizing the negative log marginal likelihood, defined as following equation:

$$\log p(\boldsymbol{g} \mid \boldsymbol{z}, \boldsymbol{\theta}) = -\frac{S}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{K}_{\boldsymbol{\theta}}| - \frac{1}{2}\boldsymbol{g}^{\top}\boldsymbol{K}_{\boldsymbol{\theta}}^{-1}\boldsymbol{g}, \tag{15}$$

where $\boldsymbol{K}_{\boldsymbol{\theta}}$ denotes the covariance matrix that depends on $\boldsymbol{\theta}$, with elements determined by the kernel function $k(z, z' \mid \boldsymbol{\theta})$, and $|\boldsymbol{K}_{\boldsymbol{\theta}}|$ represents the determinant. The gradient of the negative log marginal likelihood [Eq. (15)] is expressed as follows:

$$\frac{\partial \log p(\boldsymbol{g} \mid \boldsymbol{z}, \boldsymbol{\theta})}{\partial \theta} = -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{K}_{\boldsymbol{\theta}}^{-\mathbf{1}}\frac{\partial \boldsymbol{K}_{\boldsymbol{\theta}}}{\partial \theta}\right) + \left(\boldsymbol{K}_{\boldsymbol{\theta}}^{-\mathbf{1}}\boldsymbol{g}\right)^{\top}\frac{\partial \boldsymbol{K}_{\boldsymbol{\theta}}}{\partial \theta}\left(\boldsymbol{K}_{\boldsymbol{\theta}}^{-\mathbf{1}}\boldsymbol{g}\right), \tag{16}$$

where $\frac{\partial \boldsymbol{K}_{\boldsymbol{\theta}}}{\partial \theta}$ denotes the matrix of the same shape as the covariance matrix $\boldsymbol{K}_{\boldsymbol{\theta}}$, and the elements of the matrix are $\frac{\partial}{\partial \theta}k(z, z' \mid \theta)$, which is each element of the covariance matrix $\boldsymbol{K}_{\boldsymbol{\theta}}$ differentiated by the hyper-parameter θ. More accurate modeling and evaluation can be expected by optimizing the hyper-parameters in each training cycle where new input data z^{*} is given.

To improve the numerical stability of optimization calculations, our system employs multi-start optimization, which starts optimization calculations from multiple initial values by adding values generated by Latin hyper-cube sampling (LHS; McKay et al., 2000) to the hyper-

parameters from the previous training cycle. In addition, we adopted the L-BFGS-B algorithm (Byrd et al., 1995) as the optimization method.

The modeling of response surfaces using the GPR has been described above. Next, we discuss evaluation using an acquisition function. The acquisition function is a combination of the mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{K}$ obtained by the GPR. First, following Lunderman et al. (2021), we adopted the EI, defined by the following equation, as the acquisition function:

$$EI(\mu, \sigma) = (\hat{g} - \mu)\Phi(d) + \sigma\phi(d). \quad (17)$$

Here, $\hat{g}$ denotes the provisional optimum solution, i.e., the minimum value of the objective function in the previous training cycle. In addition, σ represents the standard deviation, which is the square root of K. Furthermore, d denotes the difference between the mean and tentative optimal value normalized by the standard deviation and can be written as $d = (\hat{g} - \mu)/\sigma$. Here, Φ and ϕ are the normal cumulative distribution function and the normal probability density function, respectively.

However, using the EI, the inverse matrix in Eq. (12), (15), and (16) could not be calculated stably due to the redundant exploration of the same input data. Therefore, we then adopted penalized EI. The local penalization method proposed by González et al. (2015) is an approach that smoothly decreases the acquisition function value near the input data. This approach assumes that the objective function is a Lipschitz continuous function and prevents the redundant exploration by setting a spherical region centered on the input data and adding a penalty to the acquisition function within that region.

In addition, since the algorithm falls into a local solution of the acquisition function, the next input data cannot be obtained appropriately, so we optimized the acquisition function (see also Shahriari et al., 2016). The use of multi-start optimization and the L-BFGS-B algorithm are the same as for the hyper-parameter optimization. To optimize the penalized EI, it is necessary to calculate the penalized EI and its derivative at the input data. The derivative of penalized EI can be described as follows:

$$\nabla \ln \widetilde{EI} = EI^{-1}\nabla EI + \sum_{s=1}^{S} \varphi(z^*, z_s)^{-1}\nabla\varphi(z^*, z_s). \tag{18}$$

Here, $\widetilde{EI}$ indicates the penalized EI. The next input data is explored after calculating the total penalty at all input data. The penalty function takes the following form:

$$\varphi(z^*, z_s) = \frac{1}{2}\mathrm{erfc}(-u), \tag{19}$$

with

$$u = \frac{1}{\sqrt{2\sigma^2}}(L\|z^* - z_s\| - \hat{g} + \mu).$$

Here, erfc is the complementary error function, and L is the Lipschitz constant. In the BO using the penalized EI, changing the ratio of "exploration and exploitation" is possible by adjusting the Lipschitz constant. As a rule of thumb, if L is 0.1 or more and less than 0.5, the setting is exploitation-oriented; if L is 0.5 or more and less than 2.0, the setting is general; and if L is 2.0 or more and less than 10.0, the setting is exploration-oriented. The derivative of the penalty function takes the following form:

$$\nabla\varphi(z^*, z_s) = \frac{e^{-u^2}}{\sqrt{2\pi\sigma^2}}\frac{2L}{\|z^* - z_s\|}(z^* - z_s). \tag{20}$$

313 The derivative of the EI can be described as follows:

314 $$\nabla EI = \frac{d\sigma}{dz}\phi(d) - \Phi(d)\frac{d\mu}{dz}. \quad (21)$$

315 The derivative of the penalized EI is described above. The penalized EI at an input data

316 is written as follows:

317 $$\ln \widetilde{EI} = \ln EI + \sum_{s=1}^{S} \ln \varphi(z^*, z_s). \quad (22)$$

318 The local penalization method calculates the total product of the acquisition function and the

319 penalty at each input data and maximizes it. In Eq. (22), the total sum is calculated by

320 applying a logarithmic characteristic.

321

322

3. Experimental Setup

a. Lorenz-96 40-variable model

We conducted an observational system simulation experiment (OSSE) using the L96 to investigate whether the BO improves the practicality of the LPF. The L96 is a toy model that simulates atmospheric variables along certain latitudes. The time evolution of the atmospheric variable is expressed as follows:

$$\frac{dx_n}{dt} = (x_{n+1} - x_{n-2})x_{n-1} - x_n + F, \tag{23}$$

where x and t denote the state variables and time step, respectively, as described in Section 2a. The subscripts n $(n = 1, \dots, N)$ represent the indices of the grid point. Since the L96 has periodic boundary conditions, the following relationship with respect to state variable at each grid point: $x_{-1} = x_{39}, x_0 = x_{40}$ and $x_{41} = x_1$ are satisfied. Each term on the right side represents the following: the first is advection, the second is diffusion, and the third is forcing F. The shift of the grid point in the advection term expresses the nonlinearity of the atmosphere. Here, one variable is simulated at each grid point in 40 grid points. The fourth-order Runge–Kutta scheme is used for time integration, where forecast time step $\Delta t = 0.01$.

Observations are generated by adding Gaussian random noise $\mathcal{N}(0, 1)$ to truth, which is a long-term integration of the L96. The observations are collected at all grid points and every 0.05 time units. We assume that the observed variables match the simulated variables and that the observation errors are uncorrelated. In addition, as a gross error check, observations are rejected if the difference between forecasts and observations exceeds 10

times the observation error.

All observations are assimilated using the LPF with 64 particles over 2 years, where 0.2 time units correspond to one Earth day, which is the error-doubling time for synoptic weather. The initial particles are generated by the long-time integration of the L96 initialized with random states.

b. Data assimilation method

First, we investigated under what conditions the LPF can estimate the more accurate analyses than the LETKF. Following Poterjoy (2016), we changed the observation operator: the linear observation operator [Eq. (24.1)] that returns the state variables as the observation variables, the weak nonlinear observation operator [Eq. (24.2)] that returns the absolute value, the strong nonlinear observation operator [Eq. (24.3)] that returns the logarithm of absolute value.

$$h(\boldsymbol{x}) = \boldsymbol{x} \tag{24.1}$$

$$h(\boldsymbol{x}) = |\boldsymbol{x}| \tag{24.2}$$

$$h(\boldsymbol{x}) = \ln(|\boldsymbol{x}|) \tag{24.3}$$

Next, we investigated the effects of changes in the inflation factors α, τ, and the localization scale r on the root mean square error between the truth and the analysis (RMSE(t vs. a)). In the LETKF, α was varied in increments of 0.001 in the range of 1.01-1.10; In the LPF, τ was varied in increments of 0.01 in the range of 0.1-1.0. In addition, r

was varied in increments of 0.1 in the range of 1-10 in both the LETKF and the LPF.

c. Parameter estimation

Furthermore, To efficiently estimate the optimum of the inflation factor τ and the localization scale r, we defined the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in the LPF as the objective function, and estimated τ and r that minimize this function using the BO.

$$g(z) = -\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(y_{n,t} - h\left(\bar{x}_{n,t}^{f}(z)\right)\right)^{2}} \tag{25}$$

where g and z denote the RMSE(o vs. f) and the input data, respectively, as described in Section 2b; y and h represent the observation and observation operator, respectively, as outlined in Section 2a; $\bar{x}_{n,t}^{f}$ is the mean of the forecast particle at nth grid point and tth time step.

As the truth cannot be obtained in a real atmosphere, we used the RMSE(o vs. f). In addition, depending on the weights of observations and forecasts, the analysis may not necessarily be close to the truth. On the other hand, the observations are perturbed around the truth, and the forecast error is expected to be smaller than the observation error in the first guess but to grow larger than the observation error over time (Otsuka and Miyoshi, 2015). Therefore, we evaluated forecast accuracy by comparing future observations and extended forecasts. This approach is equivalent to indirectly evaluating the analysis

accuracy. Extended forecasts are conducted for all particles. This assumption holds if the optimum analyses are estimated and outliers in the observations are rejected. Although this assumption is valid in the experimental settings of this study, it may not always hold in general.

In addition, since the BO in this study uses the extended forecasts as arguments for the objective function, parameter estimation that takes into account model errors that develop over time is expected to also be possible. When the RMSE(o vs. f) is used as the objective function, the BO estimates the most fitting parameter for all observations within the experimental period. In this study, we estimated parameters that minimize the period average RMSE(o vs. f) by executing the OSSE multiple times during the same period. Therefore, extending the experiment period will enable us to estimate parameters that lead to long-term stable operation of the LPF.

Unlike an online system, an offline system performs analysis-forecast cycles and training cycles separately. Therefore, we could use the future observations. Our system is reasonable, considering that the NWP is performed using the optimum parameters for the past period. In addition, the length of the extended forecast was set to 0.4 time units based on the error-doubling time.

The offline system was executed according to the following procedure:

1) Execute the OSSE using τ and r generated by the LHS.

2) Calculate the RMSE(o vs. f)s and provide them as the initial input data to the BO.

3) Estimate τ and r that minimize the RMSE(o vs. f) using the BO.

Here, we show the flowchart of the offline system in Fig. 1. The numbers of each process correspond to the numbers in Fig. 1.

Fig. 1

We provided the initial input data generated by the LHS to the BO, performed the OSSE with the estimated τ and r, and repeated the training cycle that estimates τ and r, which minimize the RMSE(o vs. f) using the BO. In this experiment, we stopped the system after 20 training cycles and considered τ and r, which minimized the RMSE(o vs. f) as the estimations by the BO. In our system, we set the number of training cycles to 20 because the GPR prediction distribution hardly changed even when more input data was added. In general, the stopping criterion of the BO is often set based on the amount of computational resources to be invested in advance and the variation of the estimation.

To evaluate the estimation accuracy and convergence rate of the BO, we compared the estimation by the BO with the estimation by the RS, following Snoek et al. (2012).In addition, as the RMSE(o vs. f) is defined as the objective function, the estimation by the BO is influenced by the Gaussian noise used to generate the observations. Therefore, we conducted 35 numerical experiments with different Gaussian noises to investigate the robustness of the BO to observations. Some numerical experiments satisfy a statistically significant number of samples with a 95% confidence coefficient.

In addition, we investigated how the estimation accuracy changes when the dimension of the parameters estimated by the BO is increased from one dimension (τ) to two dimensions

422 (τ, r). In particular, since the Lipschitz constant [Eq. (19)], which determines the ratio of
423 "exploration and exploitation", and the number of initial input data are expected to have a
424 marked influence on the estimation accuracy of the BO. Therefore, we also conducted
425 sensitivity experiments for these parameters.
426
427

4. Result and Discussion

a. Data assimilation method

First, we investigated under what conditions the LPF can estimate the more accurate analysis than the LETKF. Figure. 2 shows the time series of the RMSE(t vs. a) and ensemble spread for LETKF and LPF. The RMSE(t vs. a) of the LETKF fluctuated in the range of 0.5-5.0, showing large fluctuations, especially in the first half of the experiment period, which corresponds to the spin-up period. On the other hand, the RMSE(t vs. a) of the LPF fluctuated in the range of 0.5-2.5. In addition, the ensemble spread of the LETKF fluctuated in the range of 0.5-1.0, while that of the LPF fluctuated in the range of 0.75-1.5. Fig. 2

These results indicate that under condition where the nonlinearity of the observation operator is strongest [Eq.(24.3)], the RMSE(t vs. a) of the LPF is smaller than that of the LETKF (the results of Eq.(24.1) and Eq.(24.2) are omitted). In addition, the experiments were conducted with different observation densities, but the RMSE(t vs. a) of the LPF was not smaller than that of the LETKF (not shown). These results are because the LETKF assumes a linear observation operator, while the LPF does not require such an assumption.

Next, we investigated how the RMSE(t vs. a) changes when the inflation factor α, τ, and localization scale r are varied. Figure. 3a shows the response surface of the RMSE(t vs. a) in the LETKF. In the LETKF, the minimum error of 1.024 was obtained when r = 6.5 and α = 1.100. The overall trend of the response surface shows that r = 2 and α = 1.05-1.10 were the appropriate localization scale and inflation factors. In addition, the RMSE(t vs. a) tends Fig. 3

448 to increase as alpha decreases. This result is because the inflation cannot easily
449 compensate for the uncertainty caused by insufficient ensemble size and the assumption of
450 linearity. On the other hand, the RMSE(t vs. a) tends to increase as r increases. This result
451 is because the sampling error of ensembles increases when remote observations are
452 assimilated. The minimum error was not included in the region of the response surface
453 mentioned above, and the boundaries of the contours were unclear. This feature was not
454 seen in the response surface when α was varied in increments of 0.01 and r in increments
455 of 1 (not shown). The OSSE has uncertainty due to sampling error, and this feature was
456 thought to appear because the local optimum can be found by reducing the parameter
457 increment size.

458 Figure. 3b shows the response surface of the RMSE(t vs. a) in the LPF. In the LPF, the
459 minimum error of 0.586 was obtained when r = 1.9 and τ = 0.53. The overall trend of the
460 response surface shows that r = 1-3 and τ = 0.4-0.6 are the appropriate localization scales
461 inflation factors. Since r was similar in the LETKF, setting this value in the L96 is considered
462 appropriate. In addition, the RMSE(t vs. a) increased when τ was too large or too small. This
463 feature is because when tau is too large, the observations are not assimilated, and when τ
464 is too small, the filter becomes unstable. r suggested the simple response in which the
465 RMSE(t vs. a) decreases as r decreases. On the other hand, when τ = 0.5 and r = 4-10,
466 there was a region where the RMSE(t vs. a) remained constant regardless of changes in r,
467 showing the complex response. This result suggests that τ is a more important parameter

468 for stabilizing the LPF.

469 Combining these results with the result in Fig. 2, it can be seen that when using the strong
470 nonlinear observation operator, the LPF can estimate the more accurate analysis than the
471 LETKF; however, in doing so, τ and r must be set to the optimum.

472

473 *b. Parameter estimation*

474 We optimized only τ using the BO. Figure. 4 shows the time series for the estimation of τ,
475 the minimum RMSE(o vs. f) by the BO, and the minimum RMSE(o vs. f) by the RS. Since
476 the minimum RMSE (o vs. f) was lower than that by the RS, it can be seen that the BO can
477 estimate the more accurate τ. In addition, the training cycle in which the RMSE(o vs. f)
478 converged was the 2nd cycle in both methods.

479 In Fig. 4, the parameter that minimizes the RMSE(o vs. f) was τ = 0.17 at the 2nd training Fig. 4
480 cycle, but in Fig. 6, the parameter that minimizes the RMSE(t vs. a) was τ = 0.46. Although
481 the estimation by the BO has not converged to the optimum, this result was because there
482 was marked noise due to the extended forecast; the response surfaces of the RMSE(o vs.
483 f) and the RMSE(t vs. a) were markedly different (not shown). The BO is a method for
484 efficiently exploring the optimum parameter. Therefore, the input data that minimizes the
485 RMSE(t vs. a) among the explored input data was adopted in practice. In this case, τ = 0.47
486 was adopted. At this point, the RMSE(t vs. a) = 0.777, indicating that using the BO enables
487 the LPF to operate stably.

To investigate the estimation results of the 1-dimensional BO in detail, the prediction distributions of the GPR were plotted. Figure. 5 (a)-(d) show the variation of mean, standard deviation (95% confidence interval), EI, penalty, penalized EI, and input data in the GPR corresponding to Fig. 4. Fig. 5

In the 0th training cycle (Fig. 5a), only the RMSE(o vs. f) when τ = 0.65 and 0.47 were given as the initial input data. The GPR standard deviation around the input data was small and showed a narrow distribution around τ = 0.4-0.7. In addition, the GPR mean was the convex function with the maximum around τ = 0.5-0.6, and at this point, the GPR could not predict whether the RMSE(o vs. f) would be smaller when τ was closer to 0.1 or 1.0. Since the EI is large when the GPR mean is small and the GPR standard deviation is large, the EI was the concave function, and the EI at τ = 0.1 is slightly larger than at τ = 1.0, reaching the maximum of the EI at this point. The penalty showed the distribution with three peaks connected in a row, with values decreasing around the input data. Since the penalized EI is calculated as the product of the EI and the penalty, the penalized EI became the concave function with sharp corners around the input data, unlike the EI.

In the 1st training cycle (Fig. 5b), the GPR standard deviation decreased around where τ = 1.0 was explored. In addition, the GPR mean predicted that the RMSE(o vs. f) would be smaller when τ was closer to 0.1 than in the 0th training cycle. The addition of input data markedly changed the distribution of the EI, which became the function that increased almost monotonically as τ decreased. In the penalty, the rightmost of the three peaks

508 became larger, resulting in the distribution with a downward slope on the left side. This result
509 is because the smaller the GPR mean, the smaller the penalty (see [Eq. (19)]). As a result,
510 the penalized EI showed the maximum at τ = 0.1. Still, the distribution was considerably
511 flatter than that of the EI.

512 In the 2nd training cycle (Fig. 5c), the GPR standard deviation decreased around τ = 0.1
513 because that point was explored. There was almost no change in the distribution of the GPR
514 mean. Although the gradient decreased, the EI continued to show the maximum at τ = 0.1.
515 If the penalty had not been implemented, it is expected that it would be impossible to
516 calculate the inverse matrix stably due to the redundant exploration. In the penalty, the
517 downward trend on the left shoulder was maintained, but the overall value increased. This
518 result is because the smaller GPR standard deviation, the larger penalty (see [Eq. (19)]). As
519 a result, the penalized EI showed the maximum at τ = 0.17, and the redundant exploration
520 was avoided.

521 In the 3rd-19th training cycles, the penalty decreased as input data were added, but the
522 GPR standard deviation, GPR mean, EI, and penalized EI did not change markedly (not
523 shown). In the 20th training cycle (Fig. 5d), the GPR standard deviation and GPR mean
524 remained almost unchanged compared to the 2nd training cycle, indicating that the GPR
525 had converged. Therefore, the EI also remained virtually unchanged. Since the input data
526 were explored relatively evenly, the penalty decreased overall, and the only feature of
527 distribution was a tendency for the penalty to increase as the GPR mean increased.

Ultimately, the penalized EI showed fairly small values overall, indicating that the input data were sufficiently explored.

Next, we verified whether the GPR prediction of the 1-dimensional BO was reasonable by comparing it with the response surface of the RMSE(t vs. a). Comparing Fig. 5d and Fig. 6, the sigmoid curve-like distribution in the GPR mean and GPR standard deviation was consistent. In addition, the input data were slightly dense around τ = 0.4 and 0.7, which match the regions with large curvatures in the true response surface (Fig. 6). This result is because these regions were explored intensively to capture the complex changes in the response surface. The range of optimum tau were explored intensively, and there were relatively large amount of the input data in this range; τ = 0.47 (at this time, the RMSE(t vs. a) = 0.777) was explored, indicating that the 1-dimensional BO can estimate τ close to the true optimum while modeling the true response surface with high accuracy.

Fig. 6

To confirm the practicality of the BO, we investigated the robustness of the BO to changes in the observations. Figure 7a shows the box-and-whisker of τ. In all of the 5th, 10th, 15th, and 20th training cycles, even when the observations were changed, the upper and lower limits of the box for τ fluctuated by only about 0.2 at most. This variation corresponds to 20% of the parameter exploration range, indicating that the estimation by the BO was reasonably robust against changes in the observations. In the 5th training cycle, the length of the whisker was about 0.3, but in the subsequent training cycles, the length of the whisker increased to about 0.4. This change means that the BO was shifting from "exploitation" to

Fig. 7

548 "exploration", and it is thought that the length of the boxes and whiskers was increasing
549 because the input data were being explored evenly. In addition, the absence of outliers
550 indicates that the BO was not exploring the extreme input data. When the response surface
551 is simple (see Fig. 5), the box-and-whisker has few outliers because the estimation by the
552 BO is unlikely to fall into a local solution.

553 Figure 7b shows the box-and-whisker of the RMSE(o vs. f). The upper and lower limits of
554 the boxes and whiskers were within the range of the RMSE(o vs. f) = 3.0-4.0 in all training
555 cycles, and the variation was smaller than that of τ. This result is because there is the certain
556 range of optimum τ, as shown by the light blue shade. The estimation of τ may appear to
557 scatter as the training cycle progresses. Still, this is not a problem in practice because the
558 input data that minimizes the RMSE(t vs. a) among the explored input data is adopted.

559 We optimized τ and r using the BO. Figure 8 shows the time series of the estimation for Fig. 8
560 τ and r, the minimum RMSE(o vs. f) by the BO, and the minimum RMSE(o vs. f) by the RS.
561 Since the minimum RMSE(o vs. f) was lower than the minimum RMSE(o vs. f) by the RS,
562 the BO can estimate τ and r with higher accuracy than the RS. In the RS, the estimation
563 converged at the 12th training cycle; while in the BO, the estimation converged at the 3rd
564 training cycle, indicating that the BO can optimize τ and r with fewer computational
565 resources than the RS.

566 In Fig. 8, the parameters that minimize the RMSE(o vs. f) were τ = 0.28 and r = 1.0 in the
567 3rd training cycle; however, the parameters that minimize the RMSE(t vs. a) in Fig. 2 were

568 τ = 0.53 and r = 1.9. Although the estimation by the BO has not converged to the optimum,
569 as in Fig. 4, this result is because the noise from the extended forecast was large and the
570 response surfaces of the RMSE(o vs. f) and the RMSE(t vs. a) were markedly different (not
571 shown). The BO is a method for efficiently exploring the optimum parameters. The input data
572 that minimizes the RMSE(t vs. a) among the explored input data is adopted in practical
573 applications. In this case, τ = 0.41 and r = 1.0 were adopted. At this point, the RMSE(t vs.
574 a) = 0.969, demonstrating that the BO can stabilize the LPF.

575 In addition, focusing on the fluctuations in the estimation of τ and r, the two showed
576 inverse correlation. This result can be explained as follows: When r is large, more
577 observations are assimilated, reducing the differences among particle weights in the LPF.
578 This mechanism has a similar effect to lowering tau in Eq. (9), causing the BO to explore the
579 input data while balancing τ and r. Therefore, it is considered that the fluctuations exhibit an
580 inverse correlation.

581 To investigate the estimation results of the 2-dimensional BO in detail, the prediction Fig. 9
582 distribution of the GPR were plotted. Fig. 9a-e show the GPR mean, GPR standard deviation
583 (95% confidence interval), EI, penalty, penalized EI, and input data variation corresponding
584 to Fig. 8.

585 In the 0th training cycle, only the RMSE(o vs. f) with τ = 0.12, 0.43, 0.5, 0.67, and 0.86,
586 and r = 7.0, 4.3, 5.6, 2.7, and 9.0 were given as the initial input data. The GPR mean (Fig.
587 9a) had the maximum at τ = 0.9 and r = 6, showing the prediction distribution similar to the

588 contour lines of a 2-dimensional normal distribution. In addition, the GPR standard deviation
589 (Fig. 9b) had the minimum at τ = 0.5 and r = 6, showing the contour lines along the
590 distribution of input data.

591 At this time, the amplitude parameter θ_1 in Eq.(11) was 1.884 (the minimum: 0.1, the
592 maximum: 10.0), the length scale parameter θ_2 of τ was 0.1 (the maximum), the length scale
593 parameter θ_3 of r was 1.0 (the maximum), and the noise parameter θ_4 was 1.0^(-10) (the
594 minimum). In this case, the variability of the GPR mean is small, there is a strong correlation
595 over a wide range of the GPR mean, and the GPR standard deviation decreases near the
596 input data.

597 The EI increases when the GPR mean is small and the GPR standard deviation is large.
598 Therefore, the EI (Fig. 9c) showed the maximum around τ = 0.1, r = 10 and τ = 0.1, r = 1.
599 The penalty (Fig. 9d) decreased around the input data, and the penalty was small in regions
600 where τ was small. This result is because the smaller GPR mean, the smaller penalty (see
601 Eq. (19)). Since the penalized EI (Fig. 9e) is calculated as the product of the EI and the
602 penalty, the two cancel each other out, resulting in the prediction distribution similar to the
603 GPR standard deviation (Fig. 9b). However, since the penalized EI reaches the maximum
604 at τ = 0.1 and r = 10, the input data explored in the 0th training cycle was located at this
605 point.

606 Following Fig. 9, we investigated how the GPR prediction distribution in the 2-dimensional
607 BO changes as the input data increases. Fig. 10a-e shows the GPR mean, GPR standard

608 deviation (95% confidence interval), EI, penalty, penalized EI, and input data variation
609 corresponding to Fig. 8.
610 The GPR mean (Fig. 10a) had the maximum around τ = 0.5 and r = 8. In addition, as the Fig. 10
611 input data increased sufficiently, the GPR standard deviation (Fig. 10b) showed the almost
612 uniform prediction distribution.
613 At this point, the hyper-parameters of the Gaussian kernel were all the same as those in
614 the 0th training cycle, except that the amplitude parameter θ_1 in Eq.(11) decreased to 1.304.
615 For this reason, the variation of the GPR mean became smaller, and it is considered that
616 the prediction distribution similar to the contour lines of a 2-dimensional normal distribution
617 was obtained.
618 In addition, in the 20th training cycle, the position of the minimum in the GPR mean
619 changed from the region with small τ to the region with small r, which was consistent with
620 the trend in response surface of the RMSE(t vs. a) (Fig. 2). This is because the increase in
621 input data enabled the overall trend of response surface to be captured, allowing the position
622 of the true minimum to be estimated more accurately.
623 When comparing the 2-dimensional response surface (Fig. 2) and the 1-dimensional
624 response surface (Fig. 6), the former exhibited the more complex distribution. Furthermore,
625 when comparing the GPR prediction distribution in the 2-dimensional BO (Fig. 10) with the
626 GPR prediction distribution in the 1-dimensional BO (Fig. 5), the former shows less
627 agreement with the true response surface. This result suggests that as the dimension of

estimated parameters increases, the estimation using the BO becomes more difficult. To model complex response surfaces, it is important to adopt the kernel functions and the acquisition functions tailored to the characteristics of problems. While libraries such as GPyOpt are robust systems that combine numerous functions. However, our system is simpler, which may explain why we obtained these results.

Since the GPR standard deviation was the almost uniform prediction distribution, the EI (Fig. 10c) showed large values in regions where the GPR mean was small (regions where τ was small). Excluding the initial input data, the input data was biased toward regions where τ was small and r was small. Therefore, the penalty (Fig. 10d) also showed small values in regions where r was small. Despite the dense input data in regions where τ was small, the penalty was large because the GPR mean was large. As a result of the EI and the penalty canceling each other out, the penalized EI (Fig. 10e) gradually increased as r increased.

The input data explored in the 20th training cycle was τ = 0.1 and r = 5.8. In addition, the exploration was conducted in regions where τ was small, rather than in regions where r was large. If the emphasis is on finding values close to the true minimum, the regions where the EI is large (where r is small) should be explored. However, it was not explored because the penalty had a marked impact. The penalty is determined by the balance among the Lipschitz constant L, the provisional optimum solution $\hat{g}$, and the mean μ in Eq. (19). Therefore, adjusting the Lipschitz constant is likely to bring about marked changes in the behavior of exploration.

Furthermore, we investigated the effects of changes in the dimension of response surface, Lipschitz constant, and number of initial input data on the estimation by the BO. Table. 1 summarizes the results of sensitivity experiment. In both 2-dimensional BO and 1-dimensional BO, as the Lipschitz constant increased, the minimum RMSE(o vs. f) decreased, and the estimation tended to be the same regardless of changes in the number of initial input data. In addition, focusing on cases with each Lipschitz constant, an increase in the number of initial input data did not necessarily improve the estimation accuracy of the BO. The system stopped in the case of 1-dimensional BO with L = 0.1 and 2 initial input data because the same input data was redundantly explored due to excessive emphasis on exploitation, and the inverse matrix of Eq. (12), (15), and (16) could not be calculated stably.

Next, focusing on the best cases for each dimension of response surface, the difference in the minimum RMSE(o vs. f) is less than 0.1, and it appears that the estimation accuracy of the BO did not decrease even if the number of estimated parameters increased. However, since the GPR did not model the true response surface very well (Fig. 2, 10) and the minimum RMSE(o vs. f) decreased as the Lipschitz constant increased (Table 1), the following conclusion can be drawn. That is, simple GPR prediction distribution is the limit for the kernel functions and the acquisition functions in our system, and the Lipschitz constant compensates for this shortcoming.

In other words, as the dimension of response surfaces increases, modeling using the GPR becomes more difficult. On the other hand, increasing the Lipschitz constant increases the

668 amount of input data to be explored. Therefore, even if the modeling using the GPR is
669 inaccurate, the BO can obtain a reasonable estimation.
670 On the other hand, when the Lipschitz constant is large, the influence of GPR mean in the
671 penalty [Eq. (19)] became relatively small, and the penalty became small only around the
672 input data (not shown). In this case, since the exploration did not consider the GPR mean,
673 it became difficult to capture sudden changes in the GPR prediction distribution. In the case
674 of simple GPR prediction distributions such as those obtained in this study, this problem can
675 be ignored. However, it is desirable to use the GPR prediction distributions that model
676 complex response surfaces and to adopt an appropriate Lipschitz constant.
677

5. Conclusion

The PF is a powerful data assimilation method that does not assume the linearity in the time evolution of errors or the Gaussian error distributions. However, the number of particles required increases exponentially with the dimensions of the dynamical system, which is a bottleneck when applying the PF to the NWP. The LPF is a method that realizes the PF in high-dimensional systems by the localization. In addition, when using the strong nonlinear observation operator, the LPF can provide a more accurate analysis than the LETKF. However, this accuracy is limited to cases where the inflation factor τ and localization scale r are set to the optima. Furthermore, as the resolution of the response surface increases and the number of estimated parameters increases (e.g., the resampling frequency and the amplitude of the Gaussian kernel), the effort and computational resources required for optimization calculations increase, and efficient parameter estimation methods are needed.

Therefore, we developed a system that uses the BO to estimate τ and r, minimizing the RMSE(o vs. f). As a result, in the case of a one-dimensional problem, the BO could model the true response surface with high accuracy and estimate τ with higher accuracy than the RS. In addition, this result was robust to changes in the observations to a certain extent. Furthermore, we found that it is important to avoid the redundant exploration using the local penalization method to stabilize the BO.

In the case of a two-dimensional problem, the BO could estimate τ and r with higher accuracy than the RS. In addition, this result was robust to changes in the observations to

698 a certain extent. However, the BO could not model the true response surface very well,
699 suggesting that it is important to adopt the kernel functions (e.g., a combination of Gaussian
700 kernel and linear kernel) and acquisition functions (e.g., using upper confidence bound and
701 improvement probability in the early training cycle and the EI in the latter training cycle)
702 tailored to the characteristics of the problem to model complex response surfaces. In
703 addition, it was found that when a simple kernel function is used, setting the Lipschitz
704 constant to a large value allows the system to operate stably.

705 Furthermore, we would like to discuss considerations for the practical application of the
706 LPF. First, as the number of particles decreases, the response surface that stabilizes the
707 LPF operation becomes narrower, so estimation by the BO is expected to become difficult.
708 In addition, although the L96 was used in this study as a proof of concept, when using more
709 advanced models, it is considered appropriate to divide the region and perform estimation
710 by the BO because the optima of τ and r are not uniform across the globe.

711 Unlike gradient methods, the BO is superior in that it can efficiently explore for globally
712 optimal parameter even when the shape of the response surface for the input and output
713 data is unknown or when the function is a multi-peaked function that cannot be differentiated.
714 This method is a vital technology for enhancing the practicality of the LPF. On the other hand,
715 to promote the use of the BO in the data assimilation framework, it is important to accumulate
716 knowledge that contributes to the fundamental understanding of the BO, as described in
717 Section 4, rather than simply using the BO as a tool. We hope that this study will contribute

718 to the promotion of the BO. In addition, further development of this technology (e.g., enabling
719 online optimization) will enhance the practicality of the LPF and ultimately improve the
720 accuracy of heavy rainfall prediction. The usefulness of the BO will eventually be
721 demonstrated in atmospheric model experiments aimed at the practical application of the
722 LPF.
723
724

Data Availability Statement

The source code used in this study is available upon request to the corresponding author.

Acknowledgments

We thank Dr. Takuya Kawabata (Meteorological Research Institute) and Associate Professor Takeshi Shibuya (University of Tsukuba) for providing valuable comments. In addition, We would like to express my sincere gratitude to the two reviewers and editor, Professor Shunji Kotsuki (Chiba University), for their honest review. Part of this research was supported by JST SPRING, Grant Number JPMJSP2124. This research was also supported by the Fundamental Technology Research of MRI (M5 and P5), a Grant-in-Aid for Scientific Research (KAKENHI) (Grant Numbers JP23H05494, JP23K17465, and JP21K13995) from the Japan Society for the Promotion of Science, and the Environmental Research and Technology Development Fund (Grant Numbers JPMEERF20245004) of the Environmental Restoration and Conservation Agency of Japan (ERCA).

References

Byrd, R. H., and P. Lu, J. Nocedal, C. Zhu, 1995: A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM J. Sci. Comput.*, **16(5)**, 1190-1208, https://doi.org/10.1137/0916069.

Evensen, G., 1994: Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *J. Geophys. Res*. **99**, 10143–10162, https://doi.org/10.1029/94JC00572.

Farchi, A., and M. Bocquet, 2018: Review article: Comparison of local particle filters and new implementations. *Nonlinear Processes Geophys.*, **25**, 765-807, https://doi.org/10.5194/npg-25-765-2018.

Gaspari, G., and S. E. Cohn, 1999: Construction of correlation functions in two and three dimensions. *Quart. J. Roy. Meteor. Soc.*, **125**, 723-757, https://doi.org/10.1002/qj.49712555417.

González, J., and Z. Dai, P. Hennig, N. D. Lawrence, 2015: Batch Bayesian Optimization via Local Penalization. *arXiv*, https://arxiv.org/abs/1505.08052.

Gordon, N. J., D. J. Salmond, and A. F. M. Smith, 1993: Novel approach to nonlinear/non-Gaussian Bayesian state estimation. IEE Proceedings F (Radar and Signal Processing), 140:2, 107–113, https://doi.org/10.1049/ip-f-2.1993.0015.

Hunt, B. R., E. J. Kostelich, and I. Szunyogh, 2007: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Phys. D*, **230**, 112–126,

https://doi.org/10.1016/j.physd.2006.11.008.

Kondo, K., and T. Miyoshi, 2019: Non-Gaussian statistics in global atmospheric dynamics: a study with a 10 240-member ensemble Kalman filter using an intermediate atmospheric general circulation model. *Nonlinear Processes Geophys.*, **26**, 211–225, https://doi.org/10.5194/npg-26-211-2019.

Kotsuki, S., T. Miyoshi, K. Kondo, and R. Potthast, 2022: A local particle filter and its Gaussian mixture extension implemented with minor modifications to the LETKF, *Geosci. Model Dev.*, **15**, 8325–8348, https://doi.org/10.5194/gmd-15-8325-2022.

Le Dimet, F. X., and O. Talagrand, 1986: Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus A*, **38(2)**, 97–110, https://doi.org/10.3402/tellusa.v38i2.11706.

Lorenz, E. N., and K. A. Emanuel, 1998: Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model. *J. Atmos. Sci.*, **55**, 399–414, https://doi.org/10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2.

Lunderman, S., M. Morzfeld, and D. J. Posselt, 2021: Using global Bayesian optimization in ensemble data assimilation: parameter estimation, tuning localization and inflation, or all of the above. *Tellus A*, **73(1)**, p. 1924952, https://doi.org/10.1080/16000870.2021.1924952.

Mckay, M. D., R. J. Beckman, and W. J. Conover, 2000: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output From a Computer Code.

Technometrics, **42(1)**, 55–61, https://doi.org/10.1080/00401706.2000.10485979.

Mockus, J., 1989: Mathematics and its Applications: Bayesian Approach to Global Optimization: Theory and Applications. Kluwer Academic Publishers, 270pp. https://doi.org/10.1007/978-94-009-0909-0.

Otsuka, S., and T. Miyoshi, 2015: A Bayesian Optimization Approach to Multimodel Ensemble Kalman Filter with a Low-Order Model. *Mon. Wea. Rev.*, **143**, 2001–2012, https://doi.org/10.1175/MWR-D-14-00148.1.

Penny, S. G., and T. Miyoshi, 2016: A local particle filter for high-dimensional geophysical systems. *Nonlinear Processes Geophys.*, **23**, 391–405, https://doi.org/10.5194/npg-23-391-2016.

Poterjoy, J., 2016: A Localized Particle Filter for High-Dimensional Nonlinear Systems. *Mon. Wea. Rev.*, **144**, 59–76, https://doi.org/10.1175/MWR-D-15-0163.1.

Poterjoy, J., and J. L. Anderson, 2016: Efficient Assimilation of Simulated Observations in a High-Dimensional Geophysical System Using a Localized Particle Filter. *Mon. Wea. Rev.*, **144**, 2007–2020, https://doi.org/10.1175/MWR-D-15-0322.1.

Potthast, R., A. Walter, and A. Rhodin, 2019: A Localized Adaptive Particle Filter within an Operational NWP Framework. *Mon. Wea. Rev.*, **147**, 345–362, https://doi.org/10.1175/MWR-D-18-0028.1.

Rasmussen, C. E., and H. Nickisch, 2010: Gaussian Processes for Machine Learning (GPML) Toolbox. *J. Mach. Learn. Res*, **11**, 3011–3015,

https://dl.acm.org/doi/abs/10.5555/1756006.1953029.

Rasmussen, C. E., and C. K. I. Williams, 2006: Gaussian Processes for Machine Learning. the MIT Press, 266pp.

Sawada, Y., 2020: Machine learning accelerates parameter optimization and uncertainty assessment of a land surface model. *J. Geophys. Res.: Atmos.*, **125**, e2020JD032688, https://doi.org/10.1029/2020JD032688.

Shahriari, B., and K. Swersky, Z. Wang, R. P. Adams, N. de Freitas, 2016: Taking the Human Out of the Loop: A Review of Bayesian Optimization. *Proc. IEEE*, **104(1)**, 148-175, https://doi.org/10.1109/JPROC.2015.2494218.

Snoek, J., H. Larochelle, and R. P. Adams, 2012: Practical Bayesian Optimization of Machine Learning Algorithms, https://doi.org/10.48550/arXiv.1206.2944.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson, 2008: Obstacles to High-Dimensional Particle Filtering. *Mon. Wea. Rev.*, **136**, 4629–4640, https://doi.org/10.1175/2008MWR2529.1.

Stordal, A. S., H. A. Karlsen, G. Nævdal, H. J. Skaug, and B. Vallès, 2011: Bridging the ensemble Kalman filter and particle filters: the adaptive Gaussian mixture filter. *Comput. Geosci.*, **15**, 293–305. https://doi.org/10.1007/s10596-010-9207-1.

List of Figures

Fig. 1. Flowchart of Bayesian optimization (BO) within the local particle filter (LPF) framework. Since the data assimilation system and the BO are implemented independently, the LPF can also be replaced with a local ensemble transform Kalman filter. Here, t $(t = 1, \dots, T)$ represents the time step, g denotes the objective function, and the subscripts s $(s = 1, \dots, S)$ denotes the indices of the input data (inflation factors α, τ, and localization scale r) and the corresponding output data (root mean square error between the observations and the forecasts; RMSE(o vs. f)). In the observing system simulation experiment (OSSE), the observations are assimilated every 6 Earth hours (0.05 time units) using the LPF, and the RMSE(o vs. f)s through the two Earth days (0.4 time units) extended ensemble forecasts at the same time step are calculated. This process in the objective function converts the input data to the output data. In the BO, input data that minimizes the objective function is estimated by modeling response surface using Gaussian process regression and evaluating using an acquisition function (penalized expected improvement). Then, the training cycles, which involved performing the OSSE with the estimated input data, are repeated. Note that the BO offline optimizes τ and r.

Fig. 2. Time series of root mean square error between the truth and the analysis (RMSE(t vs. a)) and ensemble spread for local ensemble transform Kalman filtering (LETKF) and

local particle filter (LPF) using 64 ensemble members (particles) and a strong nonlinear observation operator. The vertical axis shows the RMSE(t vs. a) and the ensemble spread, and the horizontal axis shows the assimilation cycle. The localization scale of the LETKF was set to r = 6.5 and the inflation factor was set to α = 1.100 (the optimum in Fig. 3a). In addition, the localization scale of the LPF was set to r = 1.9 and the inflation factor was set to τ = 0.53 (the optimum in Fig. 3b).

Fig. 3. Response surface of root mean square error between the truth and the analysis in local ensemble transform Kalman filter (LETKF) and local particle filter (LPF) using 64 ensemble members (particles) and strong nonlinear observation operator. The vertical axis shows the localization scale r, and the horizontal axis shows the inflation factor α and τ. The minimum error of 1.024 in the LETKF was obtained when r = 6.5 and α = 1.100 (cross mark). In addition, the minimum error of 0.586 in the LPF was obtained when r = 1.9 and τ = 0.53 (cross mark).

Fig. 4. Time series of estimation by 1-dimensional Bayesian optimization (BO). The blue line shows the inflation factor τ, the green line shows the minimum root mean square error between the observations and the forecasts (RMSE(o vs. f)) in the previous training cycle estimated by the BO, and the purple line shows the minimum RMSE(o vs. f) in the previous training cycle estimated by random sampling. In addition, the light blue

shaded region indicates the range of optimum inflation factor (τ = 0.33-0.49) for which the root mean square error between the truth and the analysis in local particle filter is less than 1.0 (the filter operates stably). The horizontal axis represents the training cycle, the first vertical axis represents τ, and the second vertical axis represents the minimum RMSE(o vs. f). The Lipschitz constant was set to L = 2.0, and the number of initial input data was set to 2 (optimum experimental settings in Table 1).

Fig. 5. Prediction distribution of Gaussian process regression (GPR) using the inflation factor τ and the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter as input and output data. The training cycle in this figure corresponds to Fig. 4. The expected value and uncertainty of the RMSE(o vs. f) are obtained as the mean (blue line) and standard deviation (blue shade) of the GPR. The green line is the penalized expected improvement (EI), the purple line is the penalty, the yellow line is the EI. Red dots indicate input data already explored, and yellow dots indicate input data explored during that training cycle. The horizontal axis represents τ, the first vertical axis represents the RMSE(o vs. f), the second vertical axis represents the penalized EI, the third vertical axis represents the penalty, and the fourth vertical axis represents the EI. (a)-(d) are the prediction distributions at the 0th (i.e., when only the initial input data were given), 1st, 2nd, and 20th training cycles, respectively.

881 Fig. 6. Response surface of root mean square error between the truth and the analysis
882 (RMSE(t vs. a)) in local particle filter (LPF) using 64 ensemble members (particles) and
883 the strong nonlinear observation operator. The vertical axis represents the RMSE(t vs.
884 a), and the horizontal axis represents the inflation factor τ. The localization scale was
885 fixed at r = 3, and the minimum error of 0.719 was obtained when τ = 0.46 (cross mark).
886 In addition, the light blue shade indicates the range of optimum inflation factor (τ = 0.33-
887 0.49) where the RMSE(t vs. a) of the LPF is 1.0 or less (the filter operates stably).
888
889 Fig. 7. Variation of the estimation by 1-dimensional Bayesian optimization when using
890 different observations. (a) Box-and-whisker of the inflation factor τ. The blue line
891 indicates the median, the lower limit of the box indicates the first quartile, the upper limit
892 of the box indicates the third quartile, the lower limit of the whisker indicates the
893 minimum, and the upper limit of the whisker indicates the maximum. In addition, the light
894 blue shaded region indicates the optimum range of the inflation factors (τ = 0.33-0.49)
895 for which the root mean square error between the observations and the forecasts
896 (RMSE(o vs. f)) in local particle filter is less than 1.0 (the filter operates stably). (b) Box-
897 and-whisker of the RMSE(o vs. f). The red line indicates the median, and the other plots
898 are the same as in (a). The limits of the vertical axis in (a) and (b) are set to reflect the
899 boundaries of τ and the RMSE(o vs. f), respectively. The horizontal axis represents the
900 number of training cycles.

Fig. 8. Time series of estimation by 2-dimensional Bayesian optimization (BO). The blue line shows the inflation factor τ, the orange line shows the localization scale r, the green line shows the minimum root mean square error between the observations and the forecasts (RMSE(o vs. f)) in the previous training cycle by the BO, and the purple line shows the minimum RMSE(o vs. f) in the previous training cycle by the random sampling. The light blue shade indicates the range of optimum inflation factor (τ = 0.32-0.67) where the root mean square error between the truth and the analysis in local particle filter is less than or equal to 1.0 (the filter operates stably). In addition, the light orange shade indicates the range of optimum localization scale (r = 1.0-4.2). The horizontal axis represents the training cycle, the first vertical axis represents τ, the second vertical axis represents r, and the third vertical axis represents the minimum RMSE(o vs. f). The Lipschitz constant was set to L = 2.0, and the number of initial input data was set to 5 (optimum experimental settings in Table 1).

Fig. 9. Prediction distribution of Gaussian process regression (GPR) using the inflation factor τ and the localization scale r at the 0th training cycle (i.e., when only initial input data were given) and the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter as the input and output data. (a) is the GPR mean, (b) is the GPR standard deviation, (c) is the expected improvement (EI), (d)

is the penalty, and (e) is the penalized EI prediction distribution. The training cycle in this figure corresponds to Fig. 8. The expected value and uncertainty of the RMSE(o vs. f) are obtained as the mean and standard deviation of the GPR. The color bar is set so that the larger value of the GPR mean and standard deviation, the greener color, and the smaller value, the bluer color. In addition, the color bar is set so that the larger value of the EI, penalty, and penalized EI, the greener color, and the smaller value, the yellower color. The red dots indicate the input data that has been explored, and the yellow dots indicate the input data explored in that training cycle. The horizontal axis represents τ, and the vertical axis represents r.

Fig. 10. Prediction distribution of Gaussian process regression (GPR) using the inflation factor τ and the localization scale r at the 20th training cycle and the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter as the input and output data. (a) is the GPR mean, (b) is the GPR standard deviation, (c) is the expected improvement (EI), (d) is the penalty, and (e) is the penalized EI prediction distribution. The training cycle in this figure corresponds to Fig. 8. The expected value and uncertainty of the RMSE(o vs. f) are obtained as the mean and standard deviation of the GPR. The color bar is set so that the larger value of the GPR mean and standard deviation, the greener color, and the smaller value, the bluer color. In addition, the color bar is set so that the larger value of the EI, penalty, and penalized

EI, the greener color, and the smaller value, the yellower color. The red dots indicate the input data that has been explored, and the yellow dots indicate the input data explored in that training cycle. The horizontal axis represents τ, and the vertical axis represents r.

961 List of Tables

962

963 Table 1 Variation of the estimation by Bayesian optimization (BO) with respect to changes
964 in the dimension of response surface, Lipschitz constant, and initial input data. The
965 rightmost column shows the minimum root mean square error between the observations
966 and the forecasts (RMSE(o vs. f)) in 20 training cycles, with the best cases for each
967 dimension of response surface highlighted in yellow. Although several cases have the
968 same minimum RMSE(o vs. f), the Lipschitz constant is generally set to L = 0.5-2.0. In
969 addition, the smaller number of initial input data, the fewer computing resources are
970 required. Therefore, the case with L = 2.0 and 5 initial input data for 2-dimensional BO,
971 and L = 2.0 and 2 initial input data for 1-dimensional BO are highlighted. In addition,
972 since the ideal number of initial input data is about 10 times the dimension of response
973 surface, the number of initial input data was changed in increments of 5 for the 2-
974 dimensional BO and 2 for the 1-dimensional BO so that the number of cases would be
975 the same. “(diverged)” indicates that the system stopped due to numerical instability in
976 the middle of 20 training cycles.

977

978

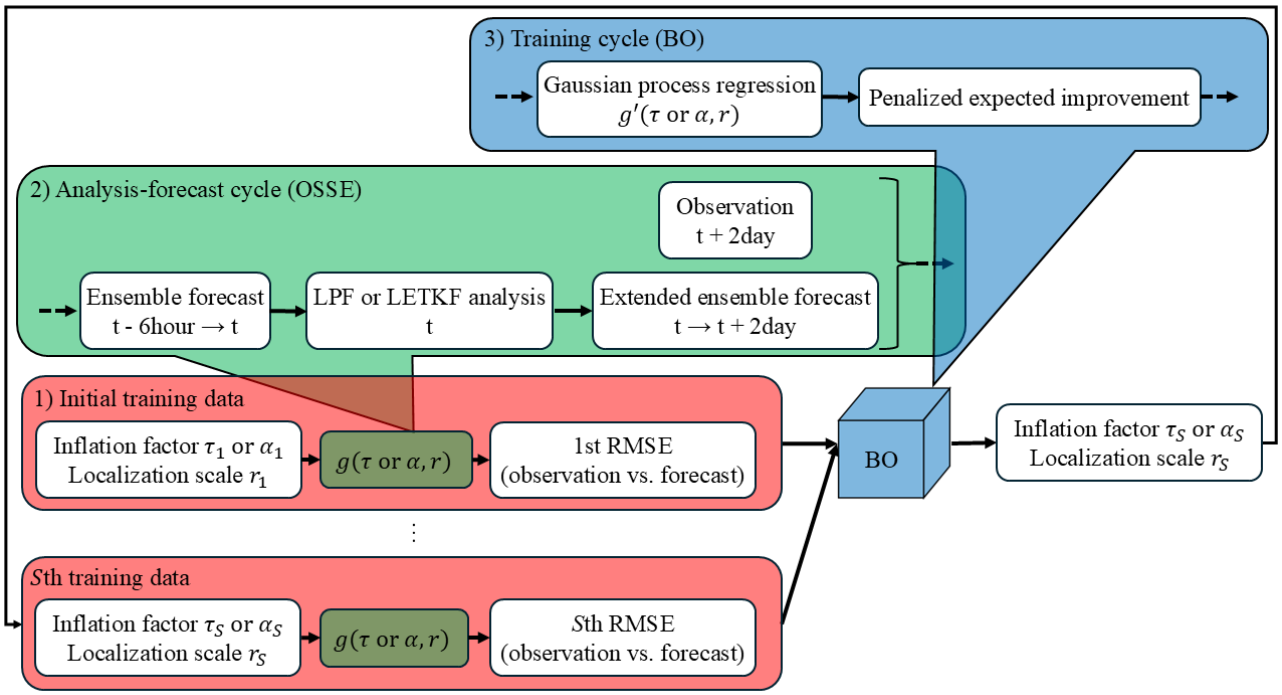

Fig. 1. Flowchart of Bayesian optimization (BO) within the local particle filter (LPF) framework. Since the data assimilation system and the BO are implemented independently, the LPF can also be replaced with a local ensemble transform Kalman filter. Here, t $(t = 1, \ldots, T)$ represents the time step, g denotes the objective function, and the subscripts s $(s = 1, \ldots, S)$ denotes the indices of the input data (inflation factors α, τ, and localization scale r) and the corresponding output data (root mean square error between the observations and the forecasts; RMSE(o vs. f)). In the observing system simulation experiment (OSSE), the observations are assimilated every 6 Earth hours (0.05 time units) using the LPF, and the RMSE(o vs. f)s through the two Earth days (0.4 time units) extended ensemble forecasts at the same time step are calculated. This process in the objective function converts the input data to the output data. In the BO, input data that minimizes the objective function is estimated by modeling response

surface using Gaussian process regression and evaluating using an acquisition function (penalized expected improvement). Then, the training cycles, which involved performing the OSSE with the estimated input data, are repeated. Note that the BO offline optimizes τ and r.

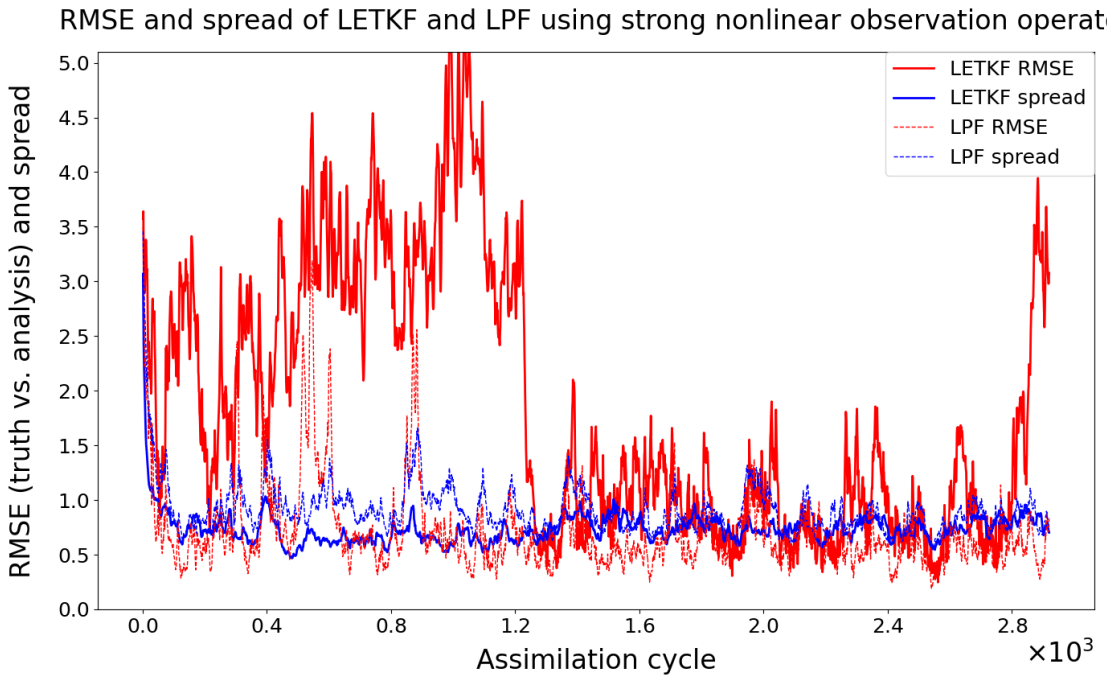

Fig. 2. Time series of root mean square error between the truth and the analysis (RMSE(t vs. a)) and ensemble spread for local ensemble transform Kalman filtering (LETKF) and local particle filter (LPF) using 64 ensemble members (particles) and a strong nonlinear observation operator. The vertical axis shows the RMSE(t vs. a) and the ensemble spread, and the horizontal axis shows the assimilation cycle. The localization scale of the LETKF was set to r = 6.5 and the inflation factor was set to α = 1.100 (the optimum in Fig. 3a). In addition, the localization scale of the LPF was set to r = 1.9 and the inflation factor was set to τ = 0.53 (the optimum in Fig. 3b).

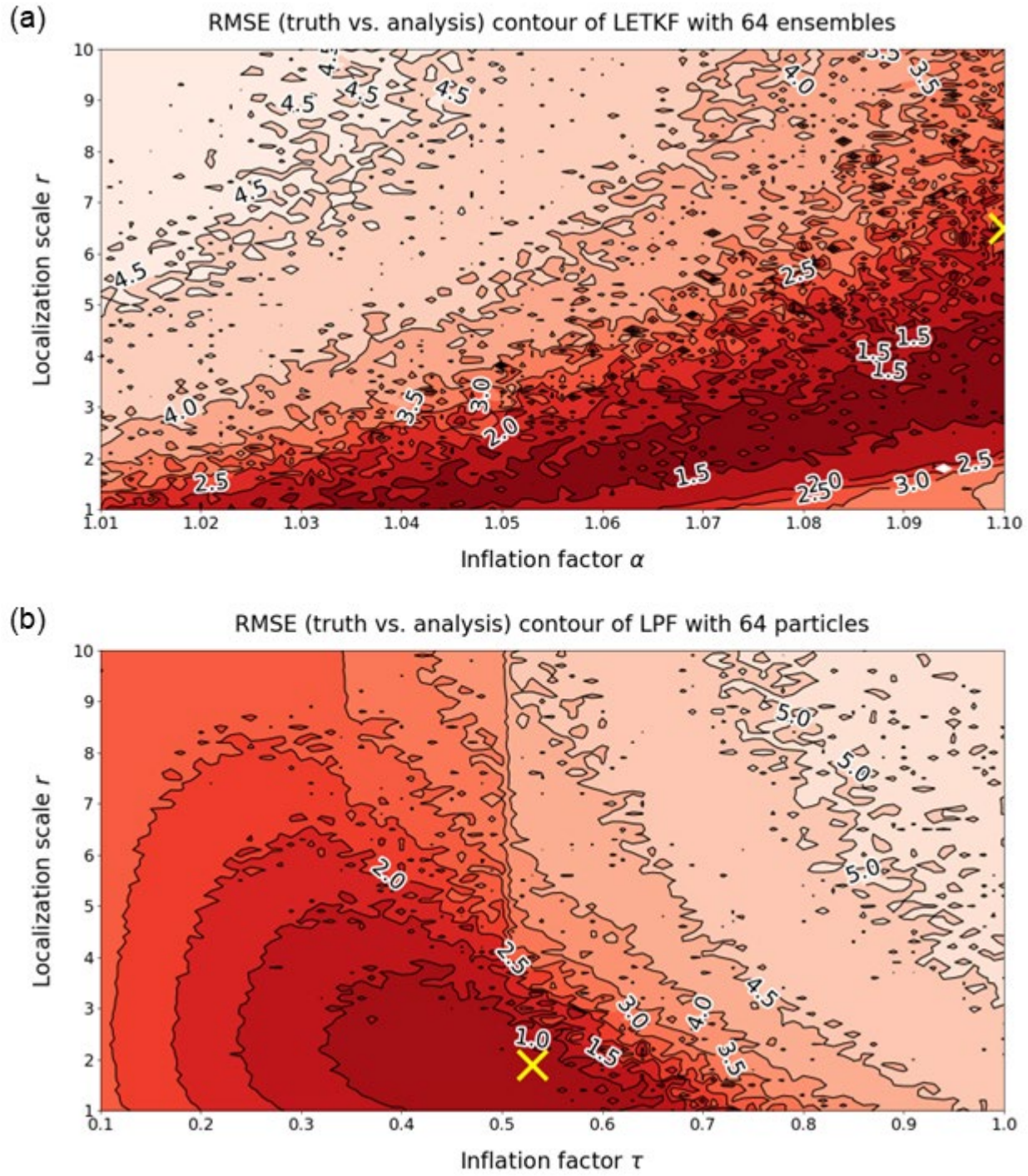

Fig. 3. Response surface of root mean square error between the truth and the analysis in local ensemble transform Kalman filter (LETKF) and local particle filter (LPF) using 64 ensemble members (particles) and strong nonlinear observation operator. The vertical axis shows the localization scale r, and the horizontal axis shows the inflation factor α

1012 and τ. The minimum error of 1.024 in the LETKF was obtained when r = 6.5 and α = 1.100

1013 (cross mark). In addition, the minimum error of 0.586 in the LPF was obtained when r =

1014 1.9 and τ = 0.53 (cross mark).

1015

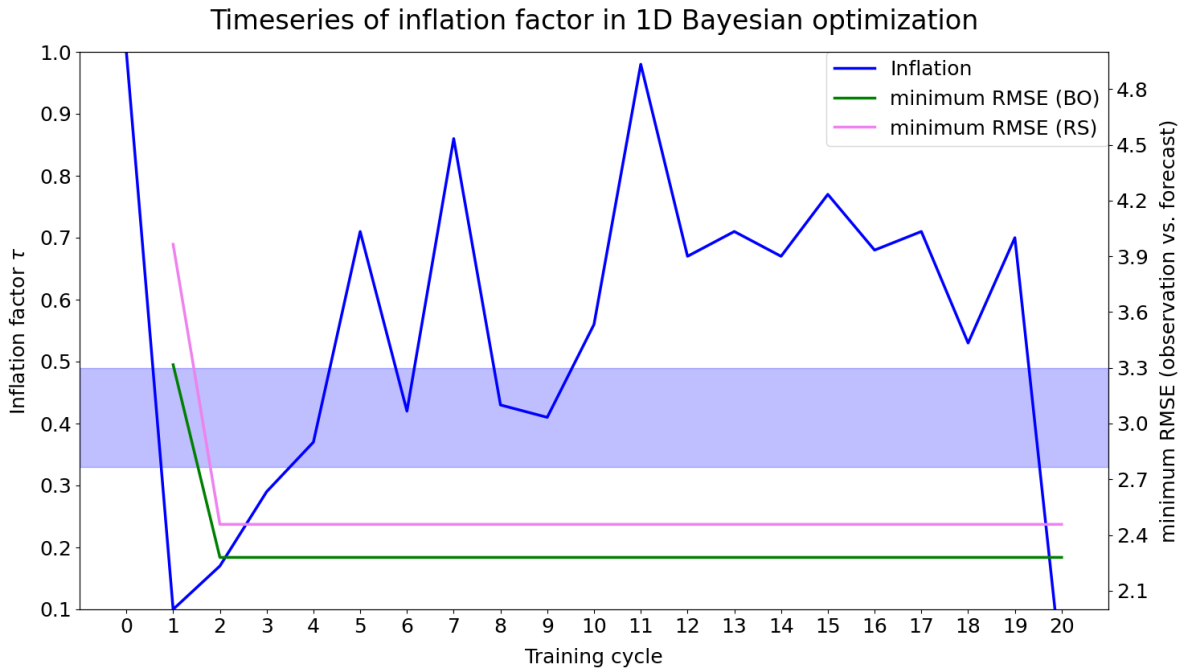

Fig. 4. Time series of estimation by 1-dimensional Bayesian optimization (BO). The blue line shows the inflation factor τ, the green line shows the minimum root mean square error between the observations and the forecasts (RMSE(o vs. f)) in the previous training cycle estimated by the BO, and the purple line shows the minimum RMSE(o vs. f) in the previous training cycle estimated by random sampling. In addition, the light blue shaded region indicates the range of optimum inflation factor (τ = 0.33-0.49) for which the root mean square error between the truth and the analysis in local particle filter is less than 1.0 (the filter operates stably). The horizontal axis represents the training cycle, the first vertical axis represents τ, and the second vertical axis represents the minimum RMSE(o vs. f). The Lipschitz constant was set to L = 2.0, and the number of initial input data was set to 2 (optimum experimental settings in Table 1).

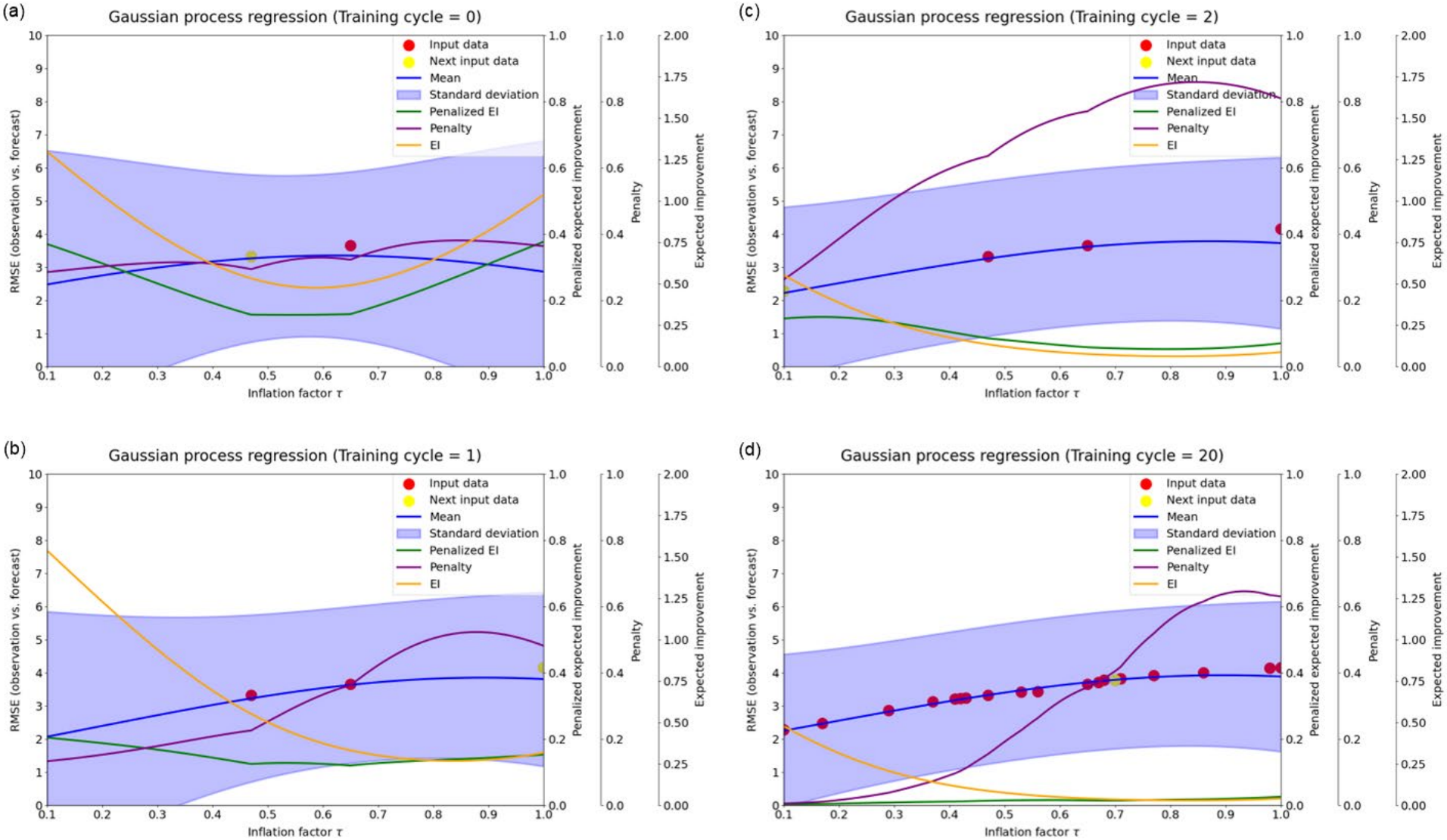
(a)
Gaussian process regression (Training cycle = 0)
(b)
Gaussian process regression (Training cycle = 1)
(c)
Gaussian process regression (Training cycle = 2)
(d)
Gaussian process regression (Training cycle = 20)
Input data
Next input data
Mean
Standard deviation
Penalized EI
Penalty
EI
RMSE (observation vs. forecast)
Penalized expected improvement
Penalty
Expected improvement
Inflation factor τ

Fig. 5. Prediction distribution of Gaussian process regression (GPR) using the inflation factor τ and the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter as input and output data. The training cycle in this figure corresponds to Fig. 4. The expected value and uncertainty of the RMSE(o vs. f) are obtained as the mean (blue line) and standard deviation (blue shade) of the GPR. The green line is the penalized expected improvement (EI), the purple line is the penalty, the yellow line is the EI. Red dots indicate input data already explored, and yellow dots indicate input data explored during that training cycle. The horizontal axis represents τ, the first vertical axis represents the RMSE(o vs. f), the second vertical axis represents the penalized EI, the third vertical axis represents the penalty, and the fourth vertical axis represents the EI. (a)-(d) are the prediction distributions at the 0th (i.e., when only the initial input data were given), 1st, 2nd, and 20th training cycles, respectively.

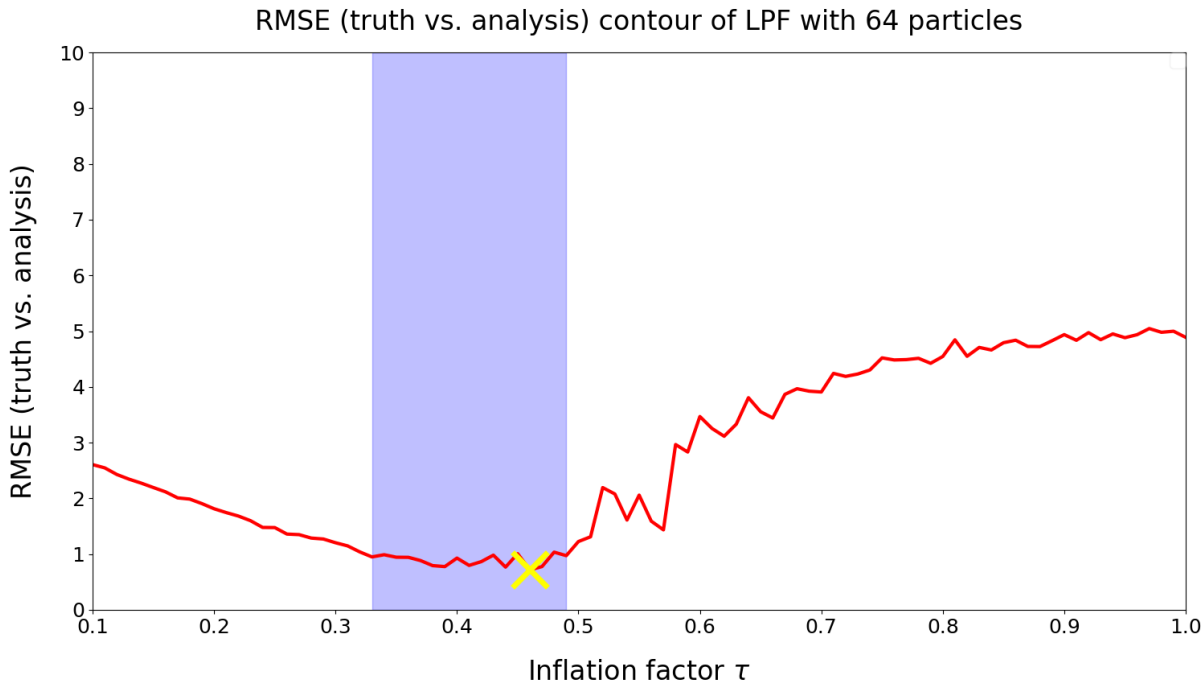

Fig. 6. Response surface of root mean square error between the truth and the analysis (RMSE(t vs. a)) in local particle filter (LPF) using 64 ensemble members (particles) and the strong nonlinear observation operator. The vertical axis represents the RMSE(t vs. a), and the horizontal axis represents the inflation factor τ. The localization scale was fixed at r = 3, and the minimum error of 0.719 was obtained when τ = 0.46 (cross mark). In addition, the light blue shade indicates the range of optimum inflation factor (τ = 0.33-0.49) where the RMSE(t vs. a) of the LPF is 1.0 or less (the filter operates stably).

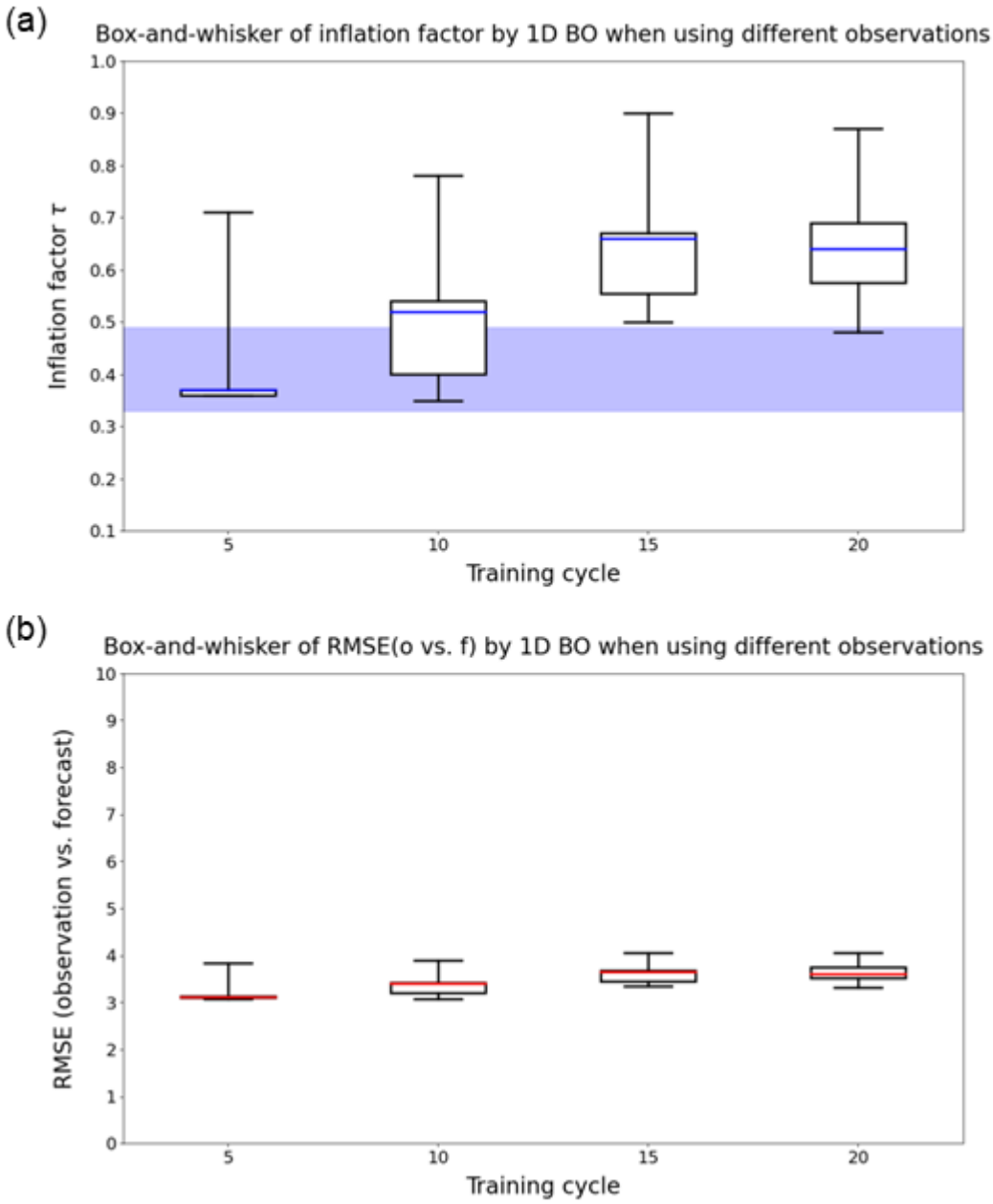

Fig. 7. Variation of the estimation by 1-dimensional Bayesian optimization when using different observations. (a) Box-and-whisker of the inflation factor τ. The blue line indicates the median, the lower limit of the box indicates the first quartile, the upper limit of the box indicates the third quartile, the lower limit of the whisker indicates the minimum, and the upper limit of the whisker indicates the maximum. In addition, the light blue shaded region indicates the optimum range of the inflation factors (τ = 0.33-0.49)

for which the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter is less than 1.0 (the filter operates stably). (b) Box-and-whisker of the RMSE(o vs. f). The red line indicates the median, and the other plots are the same as in (a). The limits of the vertical axis in (a) and (b) are set to reflect the boundaries of τ and the RMSE(o vs. f), respectively. The horizontal axis represents the number of training cycles.

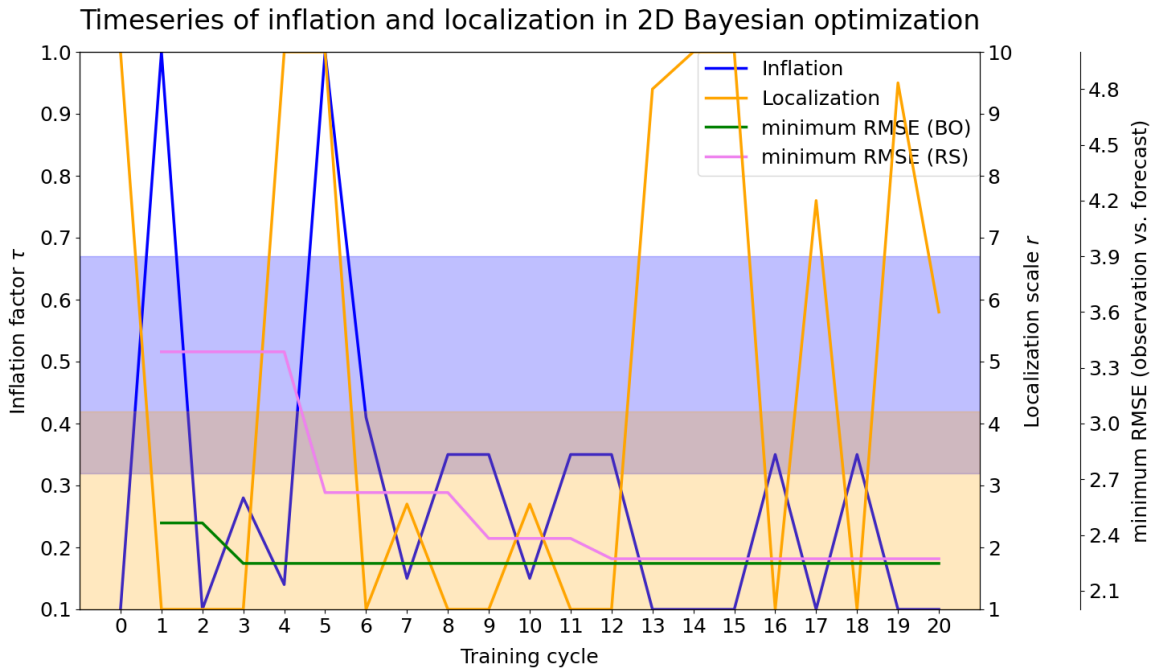

Fig. 8. Time series of estimation by 2-dimensional Bayesian optimization (BO). The blue line shows the inflation factor τ, the orange line shows the localization scale r, the green line shows the minimum root mean square error between the observations and the forecasts (RMSE(o vs. f)) in the previous training cycle by the BO, and the purple line shows the minimum RMSE(o vs. f) in the previous training cycle by the random sampling. The light blue shade indicates the range of optimum inflation factor (τ = 0.32-0.67) where the root mean square error between the truth and the analysis in local particle filter is less than or equal to 1.0 (the filter operates stably). In addition, the light orange shade indicates the range of optimum localization scale (r = 1.0-4.2). The horizontal axis represents the training cycle, the first vertical axis represents τ, the second vertical axis represents r, and the third vertical axis represents the minimum RMSE (o vs. f). The Lipschitz constant was set to L = 2.0, and the number of initial input data was set to 5 (optimum experimental

settings in Table 1).

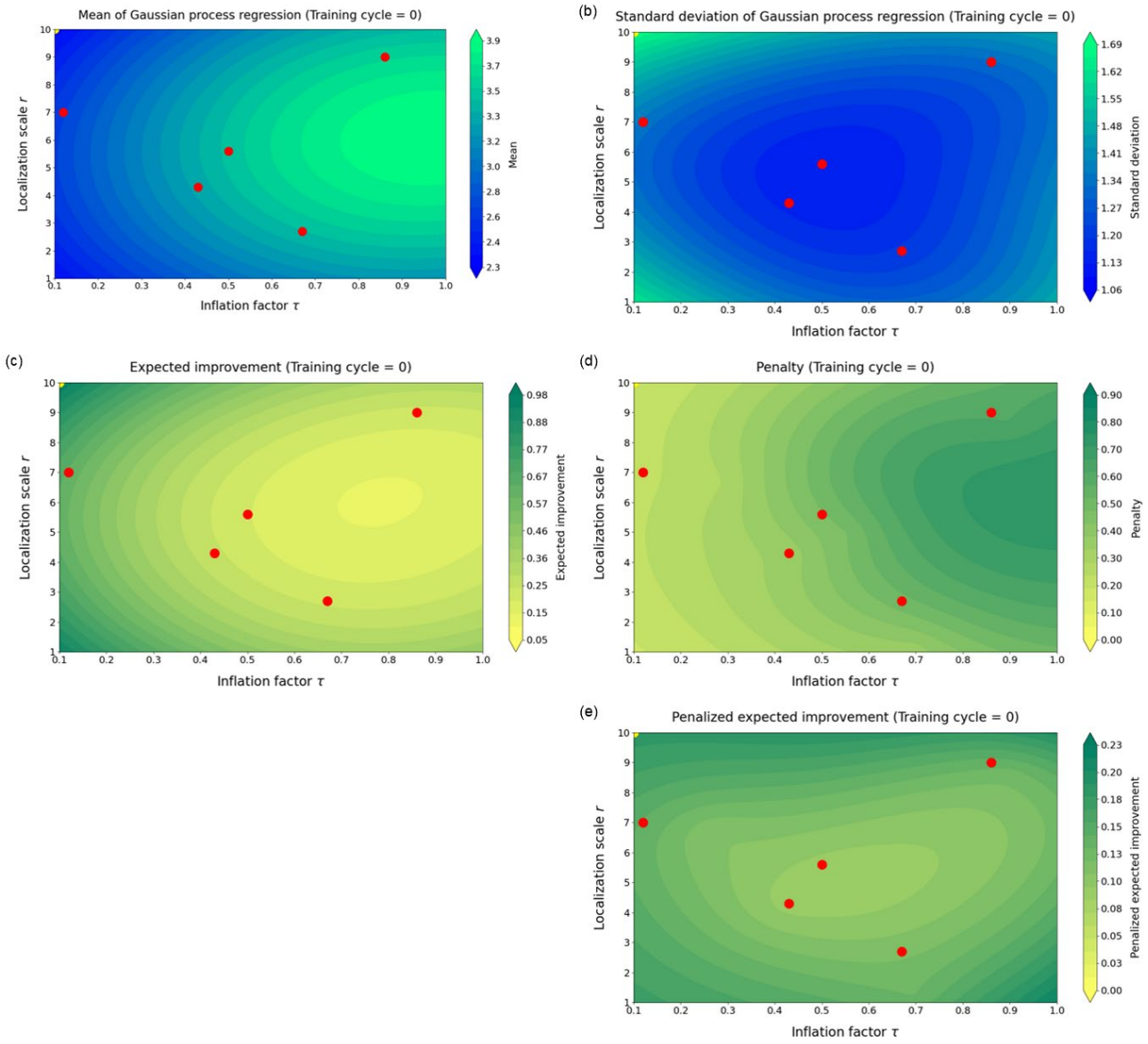

Fig. 9. Prediction distribution of Gaussian process regression (GPR) using the inflation factor τ and the localization scale r at the 0th training cycle (i.e., when only initial input data were given) and the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter as the input and output data. (a) is the GPR mean, (b) is the GPR standard deviation, (c) is the expected improvement (EI), (d) is the penalty, and (e) is the penalized EI prediction distribution. The training cycle in this figure corresponds to Fig. 8. The expected value and uncertainty of the RMSE(o vs. f)

are obtained as the mean and standard deviation of the GPR. The color bar is set so that the larger value of the GPR mean and standard deviation, the greener color, and the smaller value, the bluer color. In addition, the color bar is set so that the larger value of the EI, penalty, and penalized EI, the greener color, and the smaller value, the yellower color. The red dots indicate the input data that has been explored, and the yellow dots indicate the input data explored in that training cycle. The horizontal axis represents τ, and the vertical axis represents r.

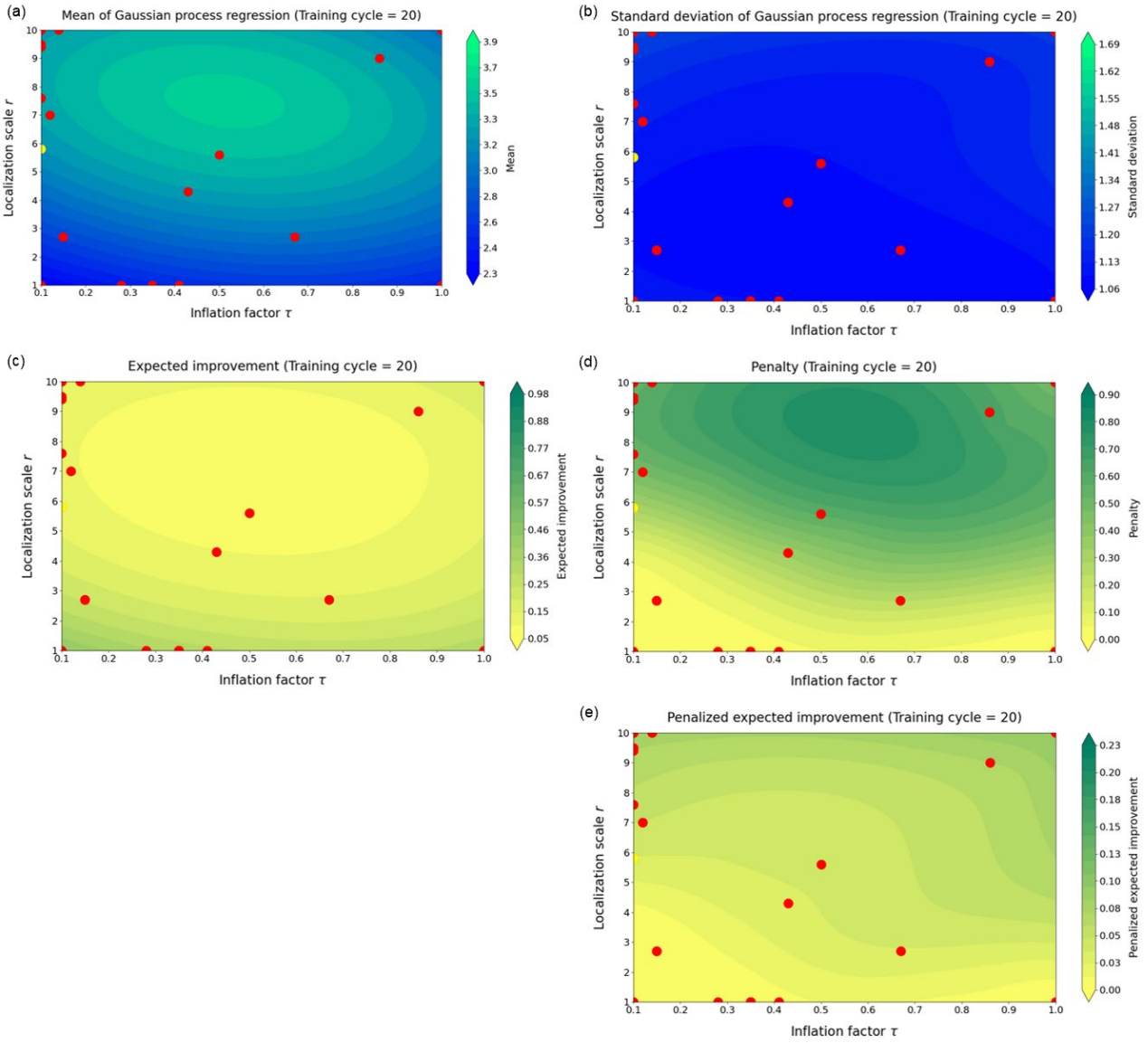

Fig. 10. Prediction distribution of Gaussian process regression (GPR) using the inflation factor τ and the localization scale r at the 20th training cycle and the root mean square error between the observations and the forecasts (RMSE(o vs. f)) in local particle filter as the input and output data. (a) is the GPR mean, (b) is the GPR standard deviation, (c) is the expected improvement (EI), (d) is the penalty, and (e) is the penalized EI prediction distribution. The training cycle in this figure corresponds to Fig. 8. The expected value and uncertainty of the RMSE(o vs. f) are obtained as the mean and

standard deviation of the GPR. The color bar is set so that the larger value of the GPR mean and standard deviation, the greener color, and the smaller value, the bluer color. In addition, the color bar is set so that the larger value of the EI, penalty, and penalized EI, the greener color, and the smaller value, the yellower color. The red dots indicate the input data that has been explored, and the yellow dots indicate the input data explored in that training cycle. The horizontal axis represents τ, and the vertical axis represents r.

Table 1 Variation of the estimation by Bayesian optimization (BO) with respect to changes in the dimension of response surface, Lipschitz constant, and initial input data. The rightmost column shows the minimum root mean square error between the observations and the forecasts (RMSE(o vs. f)) in 20 training cycles, with the best cases for each dimension of response surface highlighted in yellow. Although several cases have the same minimum RMSE(o vs. f), the Lipschitz constant is generally set to L = 0.5-2.0. In addition, the smaller number of initial input data, the fewer computing resources are required. Therefore, the case with L = 2.0 and 5 initial input data for 2-dimensional BO, and L = 2.0 and 2 initial input data for 1-dimensional BO are highlighted. In addition, since the ideal number of initial input data is about 10 times the dimension of response surface, the number of initial input data was changed in increments of 5 for the 2-dimensional BO and 2 for the 1-dimensional BO so that the number of cases would be the same. "(diverged)" indicates that the system stopped due to numerical instability in the middle of 20 training cycles.

2-dimension

Lipschitz constant	Number of initial training data	minimum RMSE
0.1	5	2.308
	10	2.515
	15	2.274
	20	2.352
0.5	5	2.453
	10	2.515
	15	2.274
	20	2.352
2.0	5	2.247

	10	2.333
	15	2.274
	20	2.308
10.0	5	2.247
	10	2.247
	15	2.247
	20	2.247

1-dimension

Lipschitz constant	**Number of initial training data**	**minimum RMSE**
0.1	2	2.280 (diverged)
	4	2.415
	6	2.315
	8	2.537
0.5	2	3.316
	4	2.415
	6	2.315
	8	2.537
2.0	2	2.280
	4	2.280
	6	2.315
	8	2.537
10.0	2	2.280
	4	2.280
	6	2.280
	8	2.280