Bayesian optimization for parameter estimation of 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

September 20th, 2024

---

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 linearity or Gaussianity. However, its application to numerical weather prediction is limited by the exponentially increasing number of particles required as the dimensionality of the dynamical system rises. Although a local particle filter (LPF) achieves the PF in high-dimensional systems through localization, the LPF remains unstable owing to its high parameter sensitivity. In the PF, maintaining particle diversity is essential to prevent “weight collapse,” and an inflation factor that smooths weights among particles is a crucial parameter that should be optimized in the LPF.

Bayesian optimization (BO) is a method for parameter estimation that minimizes (or maximizes) an objective function and is used for parameter optimization of neural networks. This study discussed the benefits of using BO within the LPF framework. As a proof of concept, we used BO to estimate the inflation factor that minimizes the root mean square error between observations and forecasts in the Lorenz-96 40-variable model. The BO quickly estimated the optimal inflation factor equivalent to the brute-force method, allowing the LPF to work stably for a decade scale. Furthermore, our method demonstrated robustness to changes in initial conditions and observations. In conclusion, using BO could greatly reduce the burden and computational cost associated with parameter optimization. The development of BO is expected to lead to further practical application of the LPF and ultimately improve the accuracy of forecasts for torrential rainfall. The benefits of BO will

eventually be demonstrated in experiments with atmospheric models 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), a slight error in the initial state can evolve into a major error. Data assimilation is a technique for estimating analysis that is closer to the truth from forecast and observation and can improve forecast accuracy using highly accurate analysis as the initial state. The ensemble Kalman filter (EnKF; Evensen, 1994) and 4-dimensional variational method (e.g., Liu and Zou, 2001) are currently the mainstream data assimilation methods, but they assume linearity of the dynamical system and Gaussianity of the probability distribution. Therefore, when these assumptions are unsatisfied—around cumulus convection and storm tracks—the optimal analysis cannot be estimated (Kondo and Miyoshi, 2019).

Conversely, the particle filter (PF; Gordon et al., 1993) is an appropriate data assimilation method for dynamical systems with strong nonlinearity because it does not assume linearity and Gaussianity. However, as the PF estimates the analysis by resampling particles (ensemble members) based on weights obtained from the likelihood of the observations, “weight collapse” occurs in high-dimensional dynamical systems. Hence, PF requires a large particle size that exponentially increases with the dynamical system dimensions (Snyder et al., 2008). This requirement has been a bottleneck for its application to NWP.

Local particle filter (LPF; Penny and Miyoshi, 2016) achieves 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

88 nearby correlations. If well applied, the LPF can estimate a more accurate analysis than the
89 EnKF with non-Gaussian observation errors, nonlinear observation operators, and sparse
90 observation networks (Poterjoy and Anderson, 2016; Poterjoy, 2016; Penny and Miyoshi,
91 2016).

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

100 The parameters to be optimized and the computational cost of data assimilation
101 experiments are expected to increase with the improvement of LPF methods and the
102 advancement of systems for applying the LPF to operational NWP. The simplest way to
103 optimize the parameters is using the brute-force method (also known as manual tuning or
104 grid search). However, this method requires data assimilation experiments that increase
105 exponentially with the number of parameters. Therefore, an efficient optimization method is
106 needed.

107 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 BO uses Gaussian process regression (GPR) to emulate the objective function, it can efficiently explore a globally optimal parameter even when the relationship between input (parameter) and output data (system response) is a black box or a nondifferentiable multimodal function. In addition, it is easy to implement because BO works independently of other systems.

The effectiveness of BO in the EnKF framework has already been demonstrated (Lunderman et al., 2021). Therefore, this study aims to continue this line of work and contribute to the practical application of LPF by solving the parameter optimization problem. We validated the effectiveness with numerical experiments using 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. Section 4 compares the estimation results using the brute-force method and BO. Furthermore, the robustness of BO is investigated. Section 5 discusses the results and provides the prospects of BO, and Section 6 presents the conclusion.

2. Methods

a. Local particle filter (LPF)

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 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 $\boldsymbol{y}$ given $\boldsymbol{x}$; $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ is the prior distribution given all $\boldsymbol{y}$ up to one time step before analysis time; 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 m $(m = 1, \dots, M)$ denotes the indices of the particle, δ is the Dirac delta function, and $\mathcal{F}$ is the numerical model. The likelihood function is arbitrary; for example, if a Gaussian distribution we used, it can be written as:

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

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 m' denotes the indices of 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 column and row 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 instead of probabilistic resampling to reduce sampling error.

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 (Hunt et al., 2007). Specifically, it is implemented by computing the Schur product of the observation error covariance matrix $\boldsymbol{R}$ in Eq. (3) and the 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):

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

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 should be optimized.

In addition, maintaining particle diversity is crucial to avoid filter divergence. 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:

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

where τ represents the inflation factor. If τ is not 1, the weights w_t^m are smoothed, and all particles have equal weights as τ close to 0. However, 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 theoretical accuracy and stability is a trade-off on the inflation factor τ. The LPF is markedly sensitive to this parameter, as will be described in Section 4a.

b. Bayesian optimization (BO)

The BO estimates a parameter that minimizes an objective function through regression with the GPR and evaluation using the acquisition function. The GPR assumes that a joint distribution $p(\boldsymbol{g})$ of S input data $\boldsymbol{\tau} = \{\tau_1, \tau_2, \dots, \tau_S\}$ and corresponding output data $\boldsymbol{g} = \{g(\tau_1), g(\tau_2), \dots, g(\tau_S)\}$ follow the multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{K})$. This assumption is written as follows:

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

where $\mathcal{GP}$ 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 $\boldsymbol{K}$ are the kernel

function $k(\tau,\tau')$. As the kernel function was arbitrary, we used the Gaussian kernel with added white noise as a general choice:

$$k(\tau,\tau'|\boldsymbol{\theta}) = \theta_1 \exp\left(-\frac{|\tau-\tau'|^2}{\theta_2}\right) + \theta_3\delta(\tau,\tau'). \quad (11)$$

Here, the kernel function $k(\tau,\tau')$ defines the correlation between any two data τ and τ' in the input data $\boldsymbol{\tau}$. In addition, $\boldsymbol{\theta} = (\theta_1,\theta_2,\theta_3)$ denotes the positive hyper-parameters that define the kernel function, while δ represents the Dirac delta function. The GPR is updated through regression and evaluation of the objective function. The new joint distribution of output data $\boldsymbol{g}^*$ given new input data $\boldsymbol{\tau}^*$ is expressed as:

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

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

$$\boldsymbol{k}_* = \left(k(\tau^*,\tau_1), k(\tau^*,\tau_2), \ldots, k(\tau^*,\tau_S)\right)^{\top}, \quad (13)$$

$$k_{**} = k(\tau^*,\tau^*). \quad (14)$$

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

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

where $\boldsymbol{K}_{\boldsymbol{\theta}}$ denotes the covariance matrix that depends on $\boldsymbol{\theta}$, with elements determined by the kernel function $k(\tau,\tau'|\theta)$, and $|\boldsymbol{K}_{\boldsymbol{\theta}}|$ represents the determinant. The optimization method can be either a numerical or gradient method. We used the Broyden–Fletcher–Goldfarb–

Shanno algorithm (BFGS; Fletcher, 2000), following Mochihashi and Oba (2019). The gradient of the negative log marginal likelihood defined by Eq. (15) is expressed as follows:

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

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 $\frac{\partial}{\partial \theta}k(\tau,\tau'|\theta)$, which is each element of the covariance matrix $\boldsymbol{K}_{\boldsymbol{\theta}}$ differentiated by the hyper-parameter θ. Accurate regression and evaluation can be expected by optimizing the hyper-parameters in each training cycle that new input data τ^* is given.

The regression with GPR is described above. The evaluation of the objective function using the acquisition function is then described. The acquisition function is a combination of mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{K}$ obtained through the GPR. Expected improvement (EI) was used as a general choice:

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

where $\hat{g}$ denotes the tentative optimal parameter that minimizes the objective function and $\boldsymbol{\sigma}$ represents the standard deviation, which is the square root of $\boldsymbol{K}$. In addition, $\boldsymbol{d}$ denotes the difference between the mean and tentative optimal parameter normalized by the standard deviation ($\boldsymbol{d} = (\hat{g} - \boldsymbol{\mu})/\boldsymbol{\sigma}$). Here, Φ and ϕ are the normal cumulative distribution function and the probability density function, respectively. The acquisition function is also arbitrary. This study selected the Gaussian kernel defined by Eq. (11) and EI defined by Eq. (17) for simplification. However, BO can be effective when appropriate functions are chosen

or designed for specific problems. For example, when hyper-parameters have high dimensionality, the Markov chain Monte Carlo method (Metropolis et al., 1953) rather than the BFGS can obtain globally optimal parameters.

3. Experimental Setup

a. Lorenz-96 40-variable model

We conducted an observing system simulation experiment (OSSE) using the L96 to validate the effectiveness of BO in the LPF framework. The L96 is a toy model that simulates atmospheric variables along certain latitudes. The time evolution of the atmospheric variable is expressed as follows:

$$\frac{dx_n}{dt} = (x_{n+1} - x_{n-2})x_{n-1} - x_n + F, \tag{18}$$

where x and t denote the state variables and time, respectively, as described in Section 2a. The subscript $n\,(n = 1, \ldots, N)$ represents the grid point. 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. In the L96, a steady solution can be obtained, where 40 variables and the forcing term $F = 8$ (Lorenz and Emanuel, 1998). 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 256 particles over 11 years, where 0.2 time units correspond to one Earth day in terms of 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. Parameter optimization

We explored an optimal combination of the localization scale r and the inflation factor τ that yields the highest analysis accuracy in the LPF using the brute-force method. The optimal combination was explored by computing the root mean square error (RMSE) between the truth and analysis for varying r $(1 \leq r \leq 10)$ in increments of 1 and τ $(0.1 \leq \tau \leq 1.0)$ in increments of 0.1. This study evaluated the benefit of BO by comparing the parameters obtained through the brute-force method with those obtained using BO.

As BO estimates the parameter that minimizes the objective function, designing the objective function appropriately is crucial. This study defined the objective function as the forecast error of the LPF.

$$g(\tau) = -\frac{1}{T}\sum_{t=1}^{T}\sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(y_t - h\left(\bar{x}_{n,t}^{f}(\tau)\right)\right)^2} \tag{19}$$

where g and τ denote the RMSE and inflation factor, respectively, as described in Section 2b; y and h represent the observation and observation operator, respectively, as outlined in

289 Section 2a; $\bar{x}_{n,t}^{f}$ is the mean of the forecast particle at nth grid point and tth time step.

290 As the truth cannot be obtained in a real atmosphere, we used the RMSE between
291 observations and forecasts. Furthermore, the observations are perturbed around the truth,
292 and the forecast error is expected to be smaller than the observation error in the first guess
293 but to grow larger than the observation error over time (Otsuka and Miyoshi, 2015).
294 Therefore, we evaluated forecast accuracy by comparing future observations and extended
295 forecasts. Extended forecasts are conducted for all particles.

296 In addition, although this study employed a perfect model, BO can estimate the parameter
297 considering model errors using the objective function, including the extended forecasts.
298 When the RMSE between observations and forecasts is used as the objective function, the
299 BO estimates the most fitting parameter for all observations within the experimental period.
300 Therefore, by extending the experimental period, the parameter for the long-term stable
301 work of the LPF can be estimated.

302 As future observations are not obtained in an online system because the analysis-forecast
303 cycle and training cycle are performed concurrently, we employed an offline system. Our
304 method is reasonable considering that the optimal parameters in the past are used for future
305 NWP. In addition, the length of the extended forecast was set to 0.4 time units based on the
306 error-doubling time.

307 As will be described in Section 4, we estimated only the inflation factor τ with the fixed
308 localization scale r in BO. The offline system was implemented as follows:

1) The OSSE was performed with the upper and lower limits of τ to obtain initial training data.

2) The RMSE between observations and forecasts was computed.

3) The τ that minimizes the RMSE was estimated using BO.

The overall flowchart of the offline system is shown in Fig. 1. The numbers associated with each process correspond to those in Fig. 1.

Fig. 1

We provided BO with RMSE for $\tau = 0.1$ and 1.0 as the initial training data and repeated the training cycle, which involved performing the OSSE with the estimated τ and finding the τ that minimizes the RMSE using BO. In this experiment, we stopped the training cycle after 20 iterations were completed, and the τ that minimized the RMSE within these training cycles was employed as the optimal inflation factor τ by BO. The number of iterations is determined based on the computational budget or fluctuation of the estimation. In the brute-force method, r and τ were individually divided into 10 parts. However, in the BO, r was fixed, while τ was divided into 100 parts to equalize the computational budget for parameter optimization.

In the training cycle, the training data (inflation factor τ) with the highest EI was explored preferentially. We set the standard deviation of the GPR to zero for the training data that had already been explored to prevent revisiting the same data. If the standard deviation of the GPR at the explored training data was not artificially set to zero, the same training data was explored again, leading to BO stopping owing to rank deficiency in the inverse matrix computation of the mean and covariance in Eq. (12).

In this study, as the RMSE between observations and forecasts is defined as the objective function, the BO estimation 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. We also investigated the training cycles of the BO required to estimate the optimal τ.

We set the initial values of the hyper-parameters in Eq. (11) to $\theta_1, \theta_2, \text{and } \theta_3 = e^{1.0}, e^{0.4}, \text{and } e^{0.1}$, following Mochihashi and Oba (2019). However, changes in $\boldsymbol{\theta}$ affect the mean and covariance in Eq. (12) and EI defined by Eq. (17) through the Gaussian kernel, thus changing the BO estimation. Therefore, to investigate the robustness of the BO to $\boldsymbol{\theta}$, we experimented with the changed initial values of hyper-parameters $\theta_1, \theta_2, \text{and } \theta_3 = 1.0, 1.0, \text{and } 1.0$.

Note the following difference. The BO estimates τ that minimizes the RMSE between observations and forecasts; however, we evaluated the estimation accuracy using the RMSE between the truth and analysis. Statistics are computed for 10 years after a one-year spin-up.

4. Results

a. Optimization using the brute-force method

Initially, we optimized the parameters using the brute-force method. Figure 2 shows the Fig. 2 contours of analysis accuracy in the LPF with each localization scale r and inflation factor τ. The optimal combination was obtained when $r = 3$ and $\tau = 0.4$; the RMSE was 0.276. Furthermore, the combinations that yield equal analysis accuracy were obtained within r $(2 \leq r \leq 7)$ and τ $(0.3 \leq \tau \leq 0.5)$. Notably, if τ was smaller than 0.2, the RMSE was almost constant even when r was varied. This result means that the LPF hardly assimilates the observations in cases where τ was smaller than 0.2. In addition, if τ was larger than 0.4, the LPF diverged rapidly, and the allowable range of r became smaller as τ increased. This result suggests that "weight collapse" is likely to occur when τ is greater than 0.4, and it is necessary to reduce the dimensionality of the observation through localization.

Overall, the LPF is markedly sensitive to τ but not so much to r, making it crucial to set τ appropriately to improve analysis accuracy. Therefore, we estimated only τ and fixed r in subsequent experiments.

b. Optimization by BO

Next, we optimized the parameters using BO. Figure 3 shows the variation of estimated Fig. 3 τ, the RMSE between observations and forecasts, and the EI across each training cycle. This figure illustrates a case with the highest analysis accuracy among 35 numerical

experiments with different observations. In the 4th training cycle, the estimation matched the optimal inflation factor found using the brute-force method and fluctuated around this optimal value in the subsequent training cycles. In addition, the RMSE was almost constant, and the EI decreased in subsequent training cycles. Thus, BO can estimate the optimal inflation factor equivalent to the brute-force method in several training cycles.

Among the 20 training cycles, the optimal inflation factor determined by BO was obtained in the 12th cycle, with $\tau = 0.44$ and an RMSE of 0.274 between the truth and analysis (the RMSE between the truth and analysis is not shown in Fig. 4). The slight difference from the optimal inflation factor $\tau = 0.4$ determined by the brute-force method is because τ was divided into 10 parts in the brute-force method but 100 parts in the BO.

We further investigated the BO estimation results in terms of the GPR. Figures 4a-d show the mean and standard deviation (95% confidence interval) of the GPR and the EI in each training cycle corresponding to Fig. 3. In the 0th training cycle (Fig. 4a), only the RMSE between observations and forecasts with $\tau = 0.1$ and 1.0 were given as the initial training data. At $\tau = 0.1$ and 1.0, the standard deviation was zero, attributed to a treatment to activate BO. The standard deviation implies the uncertainty of the regression. However, the relationship between τ and RMSE is unique, and no uncertainty exists if the settings of the data assimilation experiment are identical. Therefore, this treatment is appropriate. In addition, the LPF diverged with both $\tau = 0.1$ and 1.0. However, the RMSE with $\tau = 1.0$ was larger, causing the GPR to become a rightward monotonically increasing function. In this

Fig. 4

387 training cycle, the EI was the maximum at around $\tau = 0.1$, with the GPR mean being the
388 minimum.

389 In the 4th training cycle (Fig. 4b), the GPR became a downward convex function and the
390 standard deviation in $\tau \geq 0.5$ was slightly larger than elsewhere. The major change in the
391 GPR was caused by the training data at $\tau = 0.33$, which was explored in the 3rd training
392 cycle. As the entropy of the training data up to the 2nd training cycle was small, the 3rd
393 training data was explored. The EI shape was similar to a Gaussian function, with a
394 maximum around $\tau = 0.4$. In addition, the EI peak moved from 0.11 to 0.12, 0.33, 0.41, and
395 0.4 in the 0th–4th training cycle. This result reflects an intention to explore $\tau \geq 0.5$, which
396 contains some unknown data, and to exploit around $\tau = 0.1$, which had the minimum GPR
397 in the previous training cycles. Furthermore, the standard deviation of the overall GPR and
398 EI decreased as the training cycle progressed. This indicates that the regression accuracy
399 of the GPR improved with the accumulation of training data, and the newly obtained entropy
400 decreased.

401 In the 12th training cycle (Fig. 4c), the overall shape of the GPR did not change noticeably,
402 but the standard deviation was smaller than in the previous training cycles. Although the
403 training data were intensively explored around $\tau = 0.4$, the entropy from these training data
404 in the 4–12th training cycles was small, and the BO estimation fluctuated around $\tau = 0.4$.
405 Exploitation was emphasized over exploration because the EI was further reduced, resulting
406 in two comparable peaks around $\tau = 0.4$.

The 20th training cycle (Fig. 4d) is also shown for reference, but no notable change occurred between the 12th and 20th training cycles. Although the two comparable peaks merged into one, this occurred because the focus gradually shifted to exploring $\tau \geq 0.5$. As we avoided reusing the same training data, an excessive number of training cycles could lead to exploring the training data far from the optimal inflation factor. Based on these results, the next experiment investigated the necessary and sufficient number of training cycles.

We investigated the robustness of the BO for the different observations. Figures 5a and b show the box-and-whiskers of inflation factor τ and the RMSE between truth and analysis for the various observations. The quantiles of τ fluctuated only about 0.1 with changing observations, indicating that BO is a robust method. As the training cycle progressed, the τ estimation fluctuated around the optimal inflation factor determined by the brute-force method, while the RMSE remained nearly constant. This result reflects that some range in τ yields optimal analysis accuracy (see Fig. 2). In addition, the median of τ tends to shift from the optimal inflation factor. This result reflects a feature of BO that involves “exploration and exploitation” (acquiring unknown data and estimating based on accumulated training data, unlike gradient methods such as Newton’s method) and thus explored distant τ as the training cycle increased (see Figs. 4a-d). However, this feature is not a disadvantage for practical use because operational NWP is conducted with the parameters that yield the highest analysis accuracy within the training cycles. The outliers of τ are training data obtained through exploration, and the result of having only one outlier in each training cycle

Fig. 5

through 35 numerical experiments indicates BO stability. Furthermore, the large whiskers of τ and RMSE in the 5th training cycle became smaller in the 10th and 15th training cycles but became larger again in the 20th training cycle, indicating that approximately 10 or 15 training cycles are sufficient to estimate the optimal τ.

We also investigated the robustness of the BO for the initial value of hyper-parameter $\boldsymbol{\theta}$. Fig. 6 Figure 6 is similar to Fig. 3 but with different initial values. Notably, the same optimal inflation factor as in Fig. 3 was obtained in the 13th training cycle. As changes in the initial values of hyper-parameters and the optimization method affect "exploration and exploitation," the path to the optimal inflation factor has changed. However, the final estimation result remains the same, indicating that BO is a robust method.

Figures 7a-d is similar to Figs. 4a-d but with different initial values. In the 0th training cycle Fig. 7 (Fig. 7a), no notable difference from Fig. 4a is observed. However, in the 4th training cycle (Fig. 7b), the mean and standard deviation of the GPR considerably differed from Fig. 4b and the training data around $\tau = 0.1$ was still being explored. Conversely, in the 12th training cycle (Fig. 7c), the training data away from the optimal inflation factor were explored. Notably, the transition of the standard deviation between the training data with a standard deviation of zero and the unknown data was smooth, and this is noticeable around $\tau = 0.6$ and 1.0 in Fig. 7(c). Although the training cycle is different, this difference in smoothness was clear when compared to around $\tau = 0.4$ and 1.0 in Fig. 4(b). This difference suggests that changing the initial value of hyper-parameters influences GPR regression accuracy.

Although difficult to interpret, further investigation of the GPR changes caused by the hyper-parameter changes could provide new insights. In addition, the smaller standard deviation and EI than in Fig. 4c indicate that entropy was efficiently obtained from the unknown data. In the 20th training cycle (Fig. 7d), the standard deviation and EI were further reduced because the training data at $\tau = 0.86$ was explored.

Finally, we evaluated BO estimation accuracy. Figure 8 shows the time series of the RMSE between the truth and analysis and the ensemble spread with the optimal inflation factor $\tau = 0.44$ determined by the BO. With the RMSE around 0.3 and the ensemble spread around 0.5, the BO contributed to stabilizing the LPF over a decade scale. However, when τ was not optimized, such as $\tau = 1.0$, the RMSE was greater than 1.0, and the ensemble spread was around 0.1 (not shown), indicating that the LPF diverged owing to “weight collapse.” As the LPF did not work with $\tau = 1.0$, even with 32768 particles, the ability of BO to optimize τ is extremely important regarding the computational cost.

Fig. 8

5. Discussion

This study estimated only the inflation factor τ using the L96 as a proof of concept. However, it is possible to simultaneously estimate the optimal localization scale, resampling frequency, and amplitude of the Gaussian kernel. In addition, if online optimization can be realized, the practicality of the LPF will be dramatically improved.

A simple kernel function is sufficient for single-parameter optimization. However, multiparameter optimization increases the dimension of input parameters in the objective function defined by Eq. (19), making regression using the GPR challenging. In this case, combining multiple kernel functions may be required to increase the expressiveness of the GPR. However, as the number of hyper-parameters increases, $\boldsymbol{\theta}$ tends to fall into a local solution of the likelihood defined by Eq. (15). Therefore, the estimation accuracy of the BO is expected to decrease. Hence, choosing on the initial value of hyper-parameters and the optimization method would be important in multiparameter optimization.

In addition, changing the initial value of hyper-parameters led to the exploration of training data away from the optimal inflation factor, and the entropy was efficiently obtained from the unknown data. Based on this result, employing approaches where initial training data are given at equal intervals—such as Latin hypergeometric sampling (McKay et al., 2000) and Sobol sequence (Sobol, 1967)—could efficiently estimate the optimal inflation factor. The importance of this approach increases as the number of parameters to be optimized increases. Furthermore, in this study, the BO could estimate the optimal inflation factor even

482 if τ, which diverges the LPF as the initial training data, were given. This BO feature is useful
483 in enhancing LPF utility.

484 Furthermore, with a decreasing number of particles, the conditions under which the LPF
485 works become increasingly severe, making estimation by BO challenging, similar to
486 multiparameter optimization. Additionally, as the optimal parameters are not uniform for all
487 domains in the advanced model, estimating the parameters by dividing the domains into
488 patches is appropriate.

489 The BO is effective because it efficiently explores a globally optimal parameter, even when
490 the input and output data relationship is a black box function or a nondifferentiable
491 multimodal function. However, as the relationship must be unique, the BO may only work
492 well if the parameters to be optimized are sensitive to the objective function (well-posed
493 problems). This feature, described in Section 3b, should be considered during BO
494 implementation.

495 In addition, when evaluating the BO estimation accuracy, the ensemble spread was
496 excessive in the LPF. This result was in contrast to the fact that the RMSE and the ensemble
497 spread almost match in the EnKF with a perfect model. The result indicates that the particles
498 become a wide-hemmed probability distribution owing to inflation. A departure between the
499 prior distribution and observations causes a “weight collapse” in the PF. Thus, the inflation
500 maintains the particle diversity by smoothing the weights among particles—widening the
501 hem of the original probability distribution. However, this treatment causes discrepancies

between the RMSE and ensemble spread. Covariance inflation in the EnKF is used to compensate for the underestimation of uncertainty due to the limitations of ensemble size (Miyoshi, 2011). Similarly, inflation in the PF is equivalent and is considered a treatment for a limited particle size.

6. Conclusions

The PF is a powerful data assimilation method that does not assume linearity and Gaussianity. However, the exponentially increasing number of required particles as the dimensions of the dynamical system increase presents a bottleneck in its application to NWP. Although the LPF achieves PF in high-dimensional systems through localization, the brute-force method in this study reveals that the inflation factor τ is more important than the localization scale r for stabilizing the LPF. The high parameter sensitivity of LPF is not negligible for its application to the NWP, and exploring the optimal parameters becomes increasingly challenging as the number of parameters to be optimized rises.

To address this challenge, we employed BO to estimate the optimal inflation factor τ that minimizes the RMSE between observations and forecasts. As a result, the BO rapidly estimated the optimal inflation factor equivalent to the brute-force method, allowing the LPF to work stably for a decade scale. Furthermore, BO was robust to changes in the initial values of the hyper-parameters and observations. Overall, using BO could greatly reduce the burden and computational cost of parameter optimization and is beneficial as it allows the LPF to work with a limited number of required particles using the optimal parameters.

Investigating the benefit of BO in data assimilation is crucial, and we hope that this study will motivate the use of BO. The development of BO is expected to enhance the practical application of LPF and ultimately improve the accuracy of forecasts for torrential rainfall. The benefit of BO will eventually be demonstrated in experiments with atmospheric models,

which aim to advance the practical application of the LPF.

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, Degree Programs in Systems and Information Engineering) for providing valuable comments. 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

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.

Fletcher, R., 2000: Practical Methods of Optimization. John Wiley & Sons, Ltd, 436pp, https://doi.org/10.1002/9781118723203.fmatter.

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.

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.

Liu, H., and X. Zou, 2001: The Impact of NORPEX Targeted Dropsondes on the Analysis and 2–3-Day Forecasts of a Landfalling Pacific Winter Storm Using NCEP 3DVAR and 4DVAR Systems, *Mon. Wea. Rev.*, **129**, 1987–2004, https://doi.org/10.1175/1520-0493(2001)129%3C1987:TIONTD%3E2.0.CO;2.

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.

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, 1953:

587 Equation of State Calculations by Fast Computing Machines. *J. Chem. Phys*, **21(6)**,
588 1087–1092, https://doi.org/10.1063/1.1699114.
589 Miyoshi, T., 2011: The Gaussian Approach to Adaptive Covariance Inflation and Its
590 Implementation with the Local Ensemble Transform Kalman Filter. *Mon. Wea. Rev.*, **139**,
591 1519–1535, https://doi.org/10.1175/2010MWR3570.1.
592 Mochihashi, D., and S. Oba, 2019: Machine learning professional series: Gaussian
593 process and machine learning. Horikoshi S. (eds.), Kodansha scientific, 233pp.
594 Mockus, J., 1989: Mathematics and its Applications: Bayesian Approach to Global
595 Optimization: Theory and Applications. Kluwer Academic Publishers, 270pp.
596 https://doi.org/10.1007/978-94-009-0909-0.
597 Otsuka, S., and T. Miyoshi, 2015: A Bayesian Optimization Approach to Multimodel
598 Ensemble Kalman Filter with a Low-Order Model. *Mon. Wea. Rev.*, **143**, 2001–2012,
599 https://doi.org/10.1175/MWR-D-14-00148.1.
600 Penny, S. G., and T. Miyoshi, 2016: A local particle filter for high-dimensional geophysical
601 systems. *Nonlinear Processes Geophys.*, **23**, 391–405, https://doi.org/10.5194/npg-23-
602 391-2016.
603 Poterjoy, J., 2016: A Localized Particle Filter for High-Dimensional Nonlinear Systems.
604 *Mon. Wea. Rev.*, **144**, 59–76, https://doi.org/10.1175/MWR-D-15-0163.1.
605 Poterjoy, J., and J. L. Anderson, 2016: Efficient Assimilation of Simulated Observations in a
606 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.

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.

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.

Sobol', I. M., 1967: On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, **7(4)**, 86-112, https://doi.org/10.1016/0041-5553(67)90144-9.

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.

627 List of Figures

628 Fig. 1. Flowchart of Bayesian optimization (BO) within the local particle filter (LPF)
629 framework, where t ($t = 1, \ldots, T$) represents the time, g denotes the objective function,
630 and S is the number of input data (inflation factor) and corresponding output data (root
631 mean square error; RMSE). In the observing system simulation experiment (OSSE), the
632 observations are assimilated every 6 Earth hours (0.05 time units) using the LPF, and
633 the RMSE between observations and two Earth day (0.4 time units) extended ensemble
634 forecasts at the same time is calculated. This process in the objective function converts
635 the input data to the output data. In the BO, the input and output data relationship is
636 approximated using the Gaussian process regression, and the input data that minimizes
637 the objective function is estimated using the expected improvement. Then, the training
638 cycles, which involved performing the OSSE with the estimated input data, are repeated.
639 Note that BO optimizes the inflation factor in the LPF offline.

640

641 Fig. 2. Parameter dependence of root mean square error (RMSE) between the truth and
642 analysis of the local particle filter (LPF) with 256 ensemble members. The horizontal and
643 vertical axes show the localization scale r and inflation factor τ, respectively. The
644 minimum error of 0.276 was obtained when $r = 3$ and $\tau = 0.4$ (the cross mark). “FILTER
645 DIVERGENCE” denotes the region with an RMSE of more than 1.0.

646

Fig. 3. Summary of estimation results obtained using Bayesian optimization. The red line is the root mean square error (RMSE) between observations and forecasts, the blue line is the inflation factor τ, and the green line is the expected improvement (EI). The dashed black line represents the optimal inflation factor using the brute-force method. The blue dot shows the optimal inflation factor among the 20 training cycles.

Fig. 4. Estimation result of the relationship between the inflation factor τ and root mean square error (RMSE) (observations versus forecasts) using Gaussian process regression (GPR). This figure corresponds to Fig. 3. The regression and its uncertainty for the RMSE are obtained as the mean and standard deviation (95% confidence interval) of the GPR. The blue line is the mean, the blue shades are the standard deviations, the green line is the expected improvement (EI), and the red dots are the training data. (a)–(d) show the estimation results in the 0th (when only the initial training data are given), 4th, 12th, and 20th training cycles, respectively.

Fig. 5. Summary of estimation results obtained by the Bayesian optimization (BO) for different observations. (a) Box-and-whisker plot of the inflation factor τ. The blue line indicates the median, the box spans the first and third quartiles, the whiskers represent the minimum and maximum values, and the dots are outliers. The dashed black line shows the optimal inflation factor using the brute-force method. (b) Box-and-whisker plot

667 of the root mean square error (RMSE) between the truth and analysis. The red line is
668 the median, although it is difficult to see owing to the slight statistical variation. The
669 dashed black line indicates the RMSE using the optimal inflation factor determined by
670 the brute-force method. Otherwise, it is as shown in (a). The outliers in (a) indicate
671 inflation factors with an RMSE greater than 1.0. As the limits on the vertical axes in (a)
672 and (b) are set to reflect each bound of the inflation factor and the RMSE, the outliers in
673 (b) are out of sight.
674
675 Fig. 6. Summary of estimation results obtained using Bayesian optimization. The red line is
676 the root mean square error (RMSE) between observations and forecasts, the blue line is
677 the inflation factor τ, and the green line is the expected improvement (EI). The dashed
678 black line represents the optimal inflation factor using the brute-force method. The blue
679 dot shows the optimal inflation factor among the 20 training cycles. In Fig. 3, the initial
680 values of the hyper-parameters in the Gaussian kernel were set to $\theta_1, \theta_2, \text{and } \theta_3 =$
681 $e^{1.0}, e^{0.4}, \text{and } e^{0.1}$, but in Fig. 6, they were set to $\theta_1, \theta_2, \text{and } \theta_3 = 1.0, 1.0, \text{and } 1.0$. Notably,
682 the vertical axis in the expected improvement (EI) has been changed.
683
684 Fig. 7. Estimation result of the relationship between the inflation factor τ and root mean
685 square error (RMSE) (observations versus forecasts) using Gaussian process
686 regression (GPR). This figure corresponds to Fig. 6. The regression and its uncertainty

687 for the RMSE are obtained as the mean and standard deviation (95% confidence
688 interval) of the GPR. The blue line is the mean, the blue shades are the standard
689 deviations, the green line is the expected improvement (EI), and the red dots are the
690 training data. (a)–(d) show the estimation results in the 0th (when only the initial training
691 data are given), 4th, 12th, and 20th training cycles, respectively. In Fig. 4, the initial
692 values of the hyper-parameters in the Gaussian kernel were set to $\theta_1, \theta_2, \text{and } \theta_3 =$
693 $e^{1.0}, e^{0.4}, \text{and } e^{0.1}$, but in Fig. 7, they were set to $\theta_1, \theta_2, \text{and } \theta_3 = 1.0, 1.0, \text{and } 1.0$.
694
695 Fig. 8. Time series of the root mean square error (RMSE) between the truth and analysis
696 and the ensemble spread with the optimal inflation factor $\tau = 0.44$ determined by the
697 Bayesian optimization (BO). The red line is the RMSE and the blue line is the ensemble
698 spread. Note that the horizontal axis is the assimilation cycle, not the training cycle.
699
700

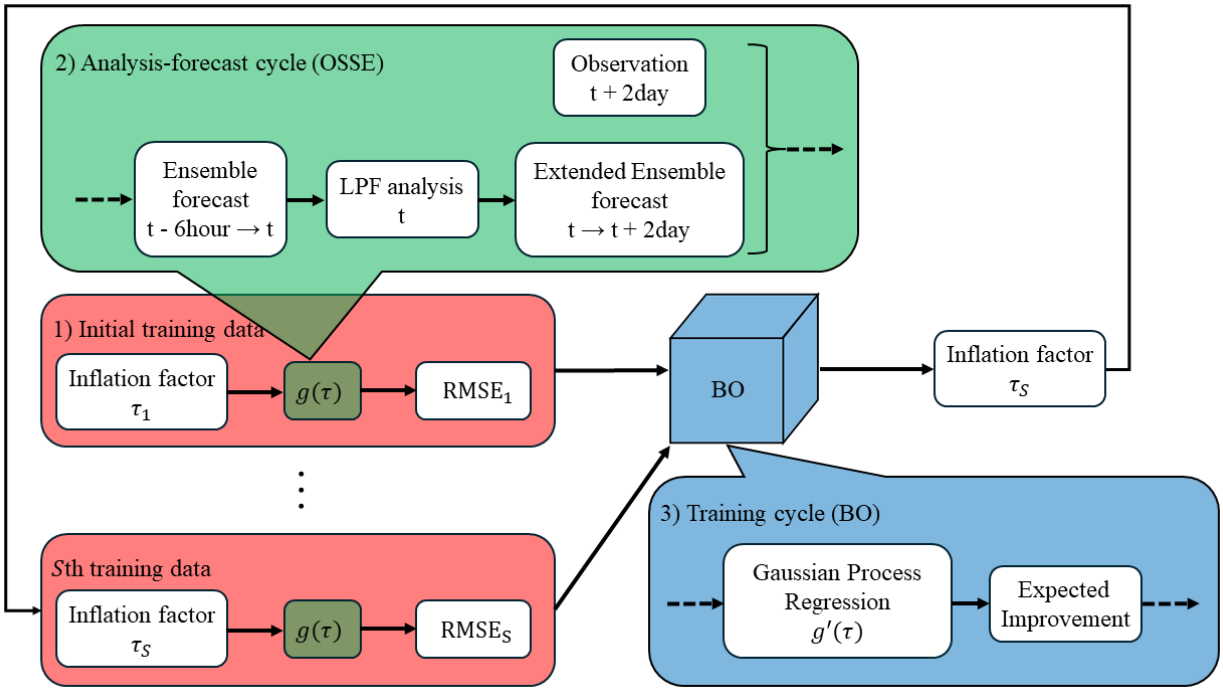
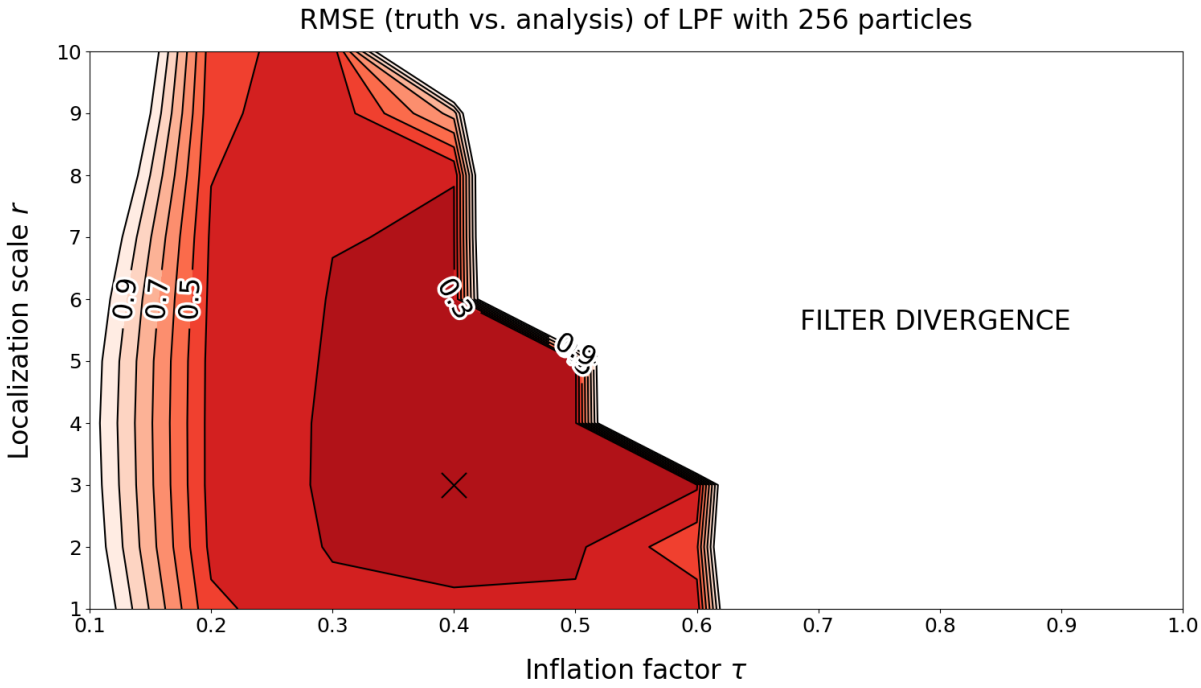


Fig. 1. Flowchart of Bayesian optimization (BO) within the local particle filter (LPF) framework, where t $(t = 1, \ldots, T)$ represents the time, g denotes the objective function, and S is the number of input data (inflation factor) and corresponding output data (root mean square error; RMSE). In the observing system simulation experiment (OSSE), the observations are assimilated every 6 Earth hours (0.05 time units) using the LPF, and the RMSE between observations and two Earth day (0.4 time units) extended ensemble forecasts at the same time is calculated. This process in the objective function converts the input data to the output data. In the BO, the input and output data relationship is approximated using the Gaussian process regression, and the input data that minimizes the objective function is estimated using the expected improvement. Then, the training cycles, which involved performing the OSSE with the estimated input data, are repeated. Note that BO optimizes the inflation factor in the LPF offline.

713

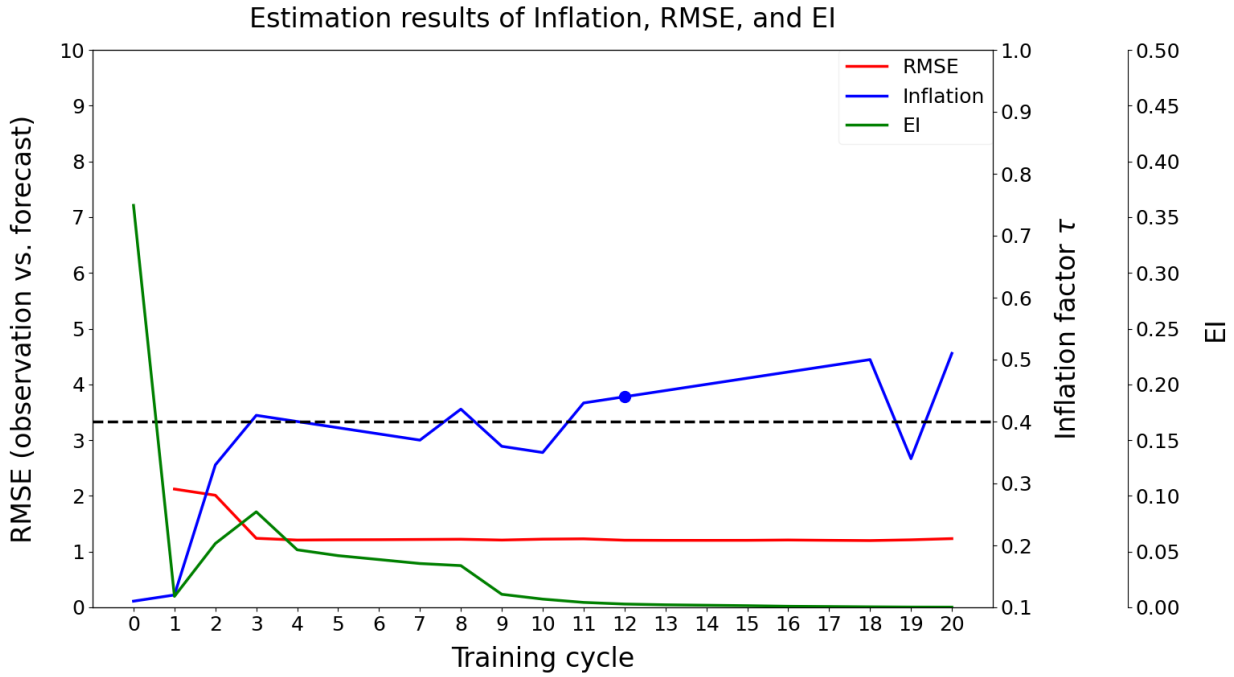
714



715 Fig. 2. Parameter dependence of root mean square error (RMSE) between the truth and
716 analysis of the local particle filter (LPF) with 256 ensemble members. The horizontal and
717 vertical axes show the localization scale r and inflation factor τ, respectively. The
718 minimum error of 0.276 was obtained when $r = 3$ and $\tau = 0.4$ (the cross mark). “FILTER
719 DIVERGENCE” denotes the region with an RMSE of more than 1.0.

720

721



722 Fig. 3. Summary of estimation results obtained using Bayesian optimization. The red line is
723 the root mean square error (RMSE) between observations and forecasts, the blue line is
724 the inflation factor τ, and the green line is the expected improvement (EI). The dashed
725 black line represents the optimal inflation factor using the brute-force method. The blue
726 dot shows the optimal inflation factor among the 20 training cycles.

727

728

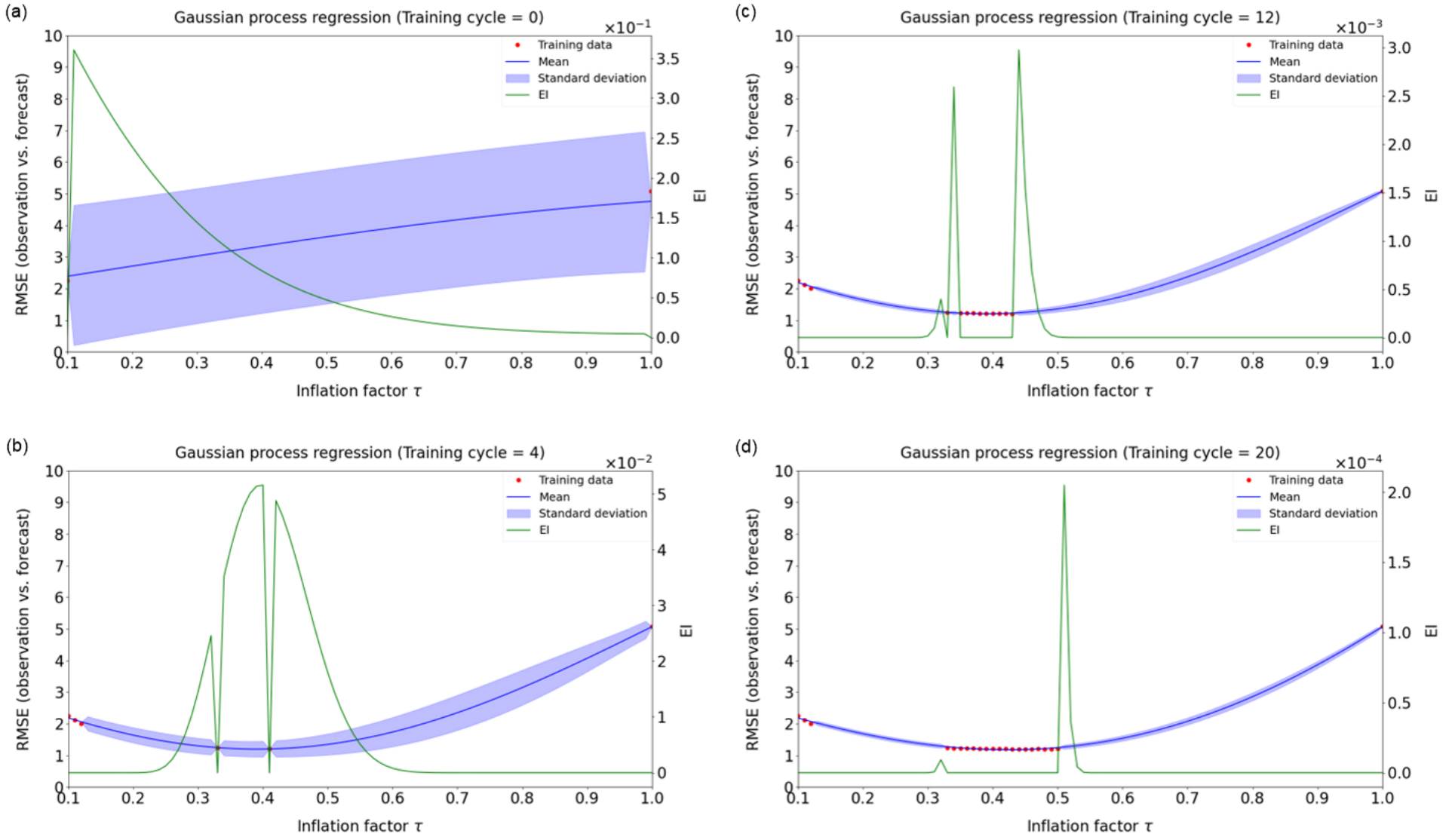

(a)
Gaussian process regression (Training cycle = 0)
×10⁻¹
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ
(c)
Gaussian process regression (Training cycle = 12)
×10⁻³
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ
(b)
Gaussian process regression (Training cycle = 4)
×10⁻²
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ
(d)
Gaussian process regression (Training cycle = 20)
×10⁻⁴
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ

Fig. 4. Estimation result of the relationship between the inflation factor τ and root mean square error (RMSE) (observations versus forecasts) using Gaussian process regression (GPR). This figure corresponds to Fig. 3. The regression and its uncertainty for the RMSE are obtained as the mean and standard deviation (95% confidence interval) of the GPR. The blue line is the mean, the blue shades are the standard deviations, the green line is the expected improvement (EI), and the red dots are the training data. (a)–(d) show the estimation results in the 0th (when only the initial training data are given), 4th, 12th, and 20th training cycles, respectively.

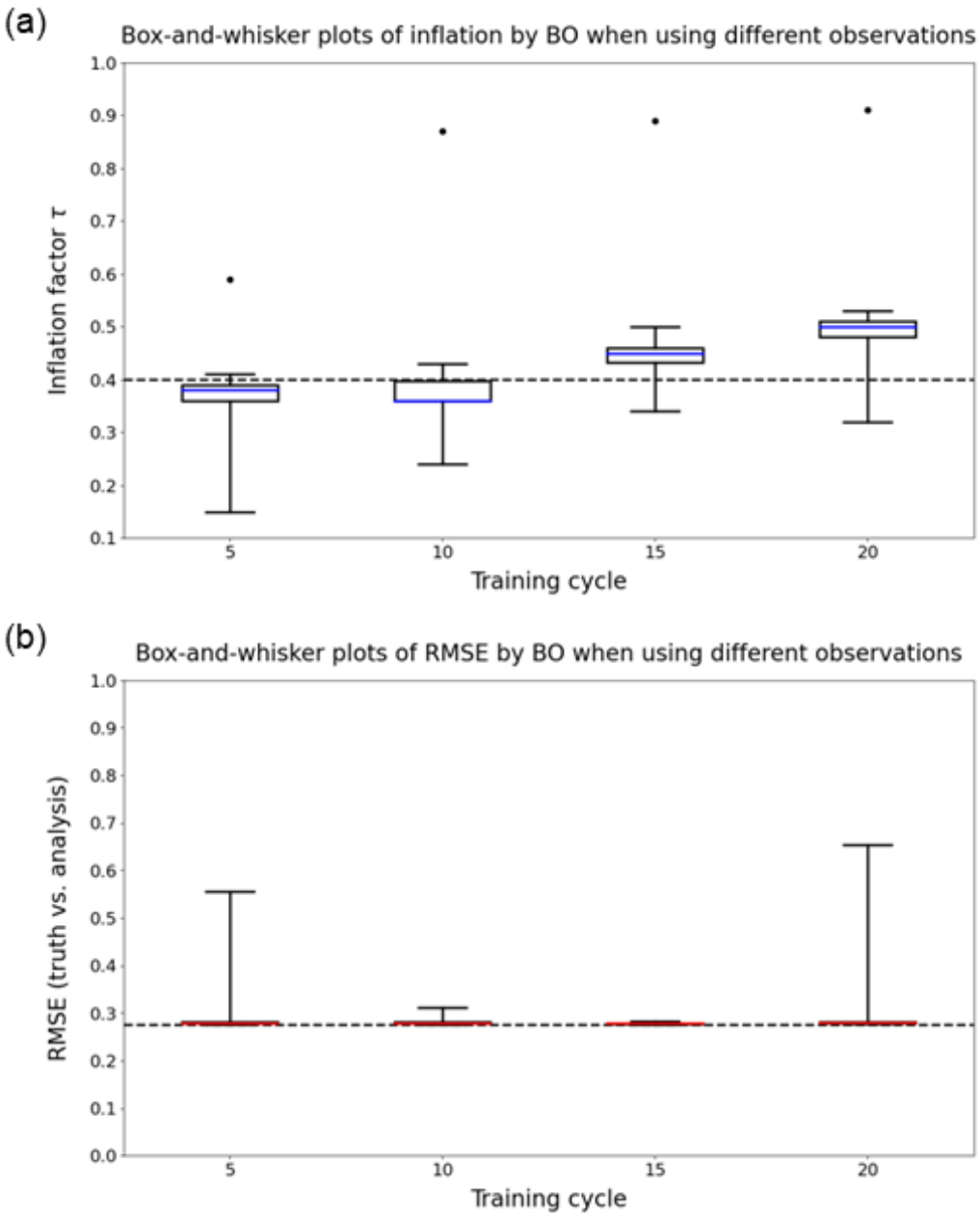


Fig. 5. Summary of estimation results obtained by the Bayesian optimization (BO) for different observations. (a) Box-and-whisker plot of the inflation factor τ. The blue line indicates the median, the box spans the first and third quartiles, the whiskers represent the minimum and maximum values, and the dots are outliers. The dashed black line shows the optimal inflation factor using the brute-force method. (b) Box-and-whisker plot of the root mean square error (RMSE) between the truth and analysis. The red line is

760 the median, although it is difficult to see owing to the slight statistical variation. The
761 dashed black line indicates the RMSE using the optimal inflation factor determined by
762 the brute-force method. Otherwise, it is as shown in (a). The outliers in (a) indicate
763 inflation factors with an RMSE greater than 1.0. As the limits on the vertical axes in (a)
764 and (b) are set to reflect each bound of the inflation factor and the RMSE, the outliers in
765 (b) are out of sight.
766
767

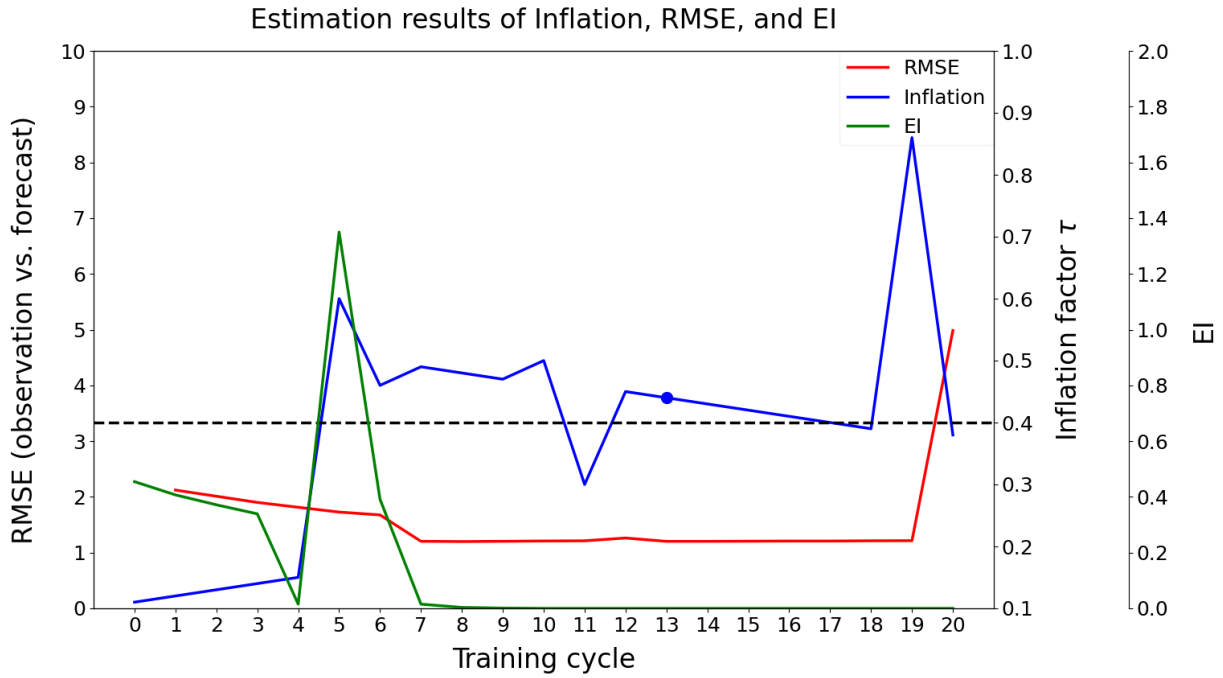
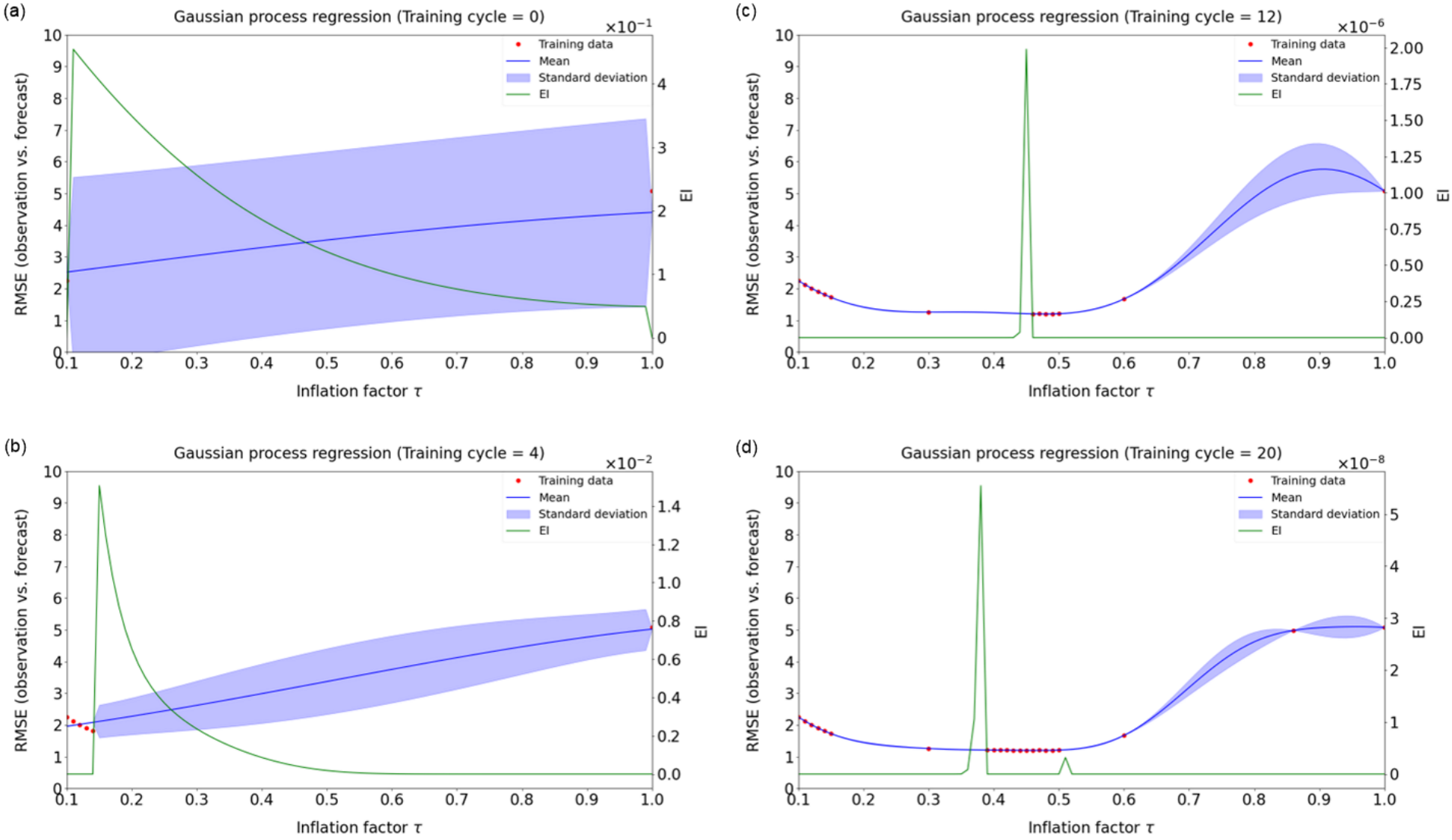
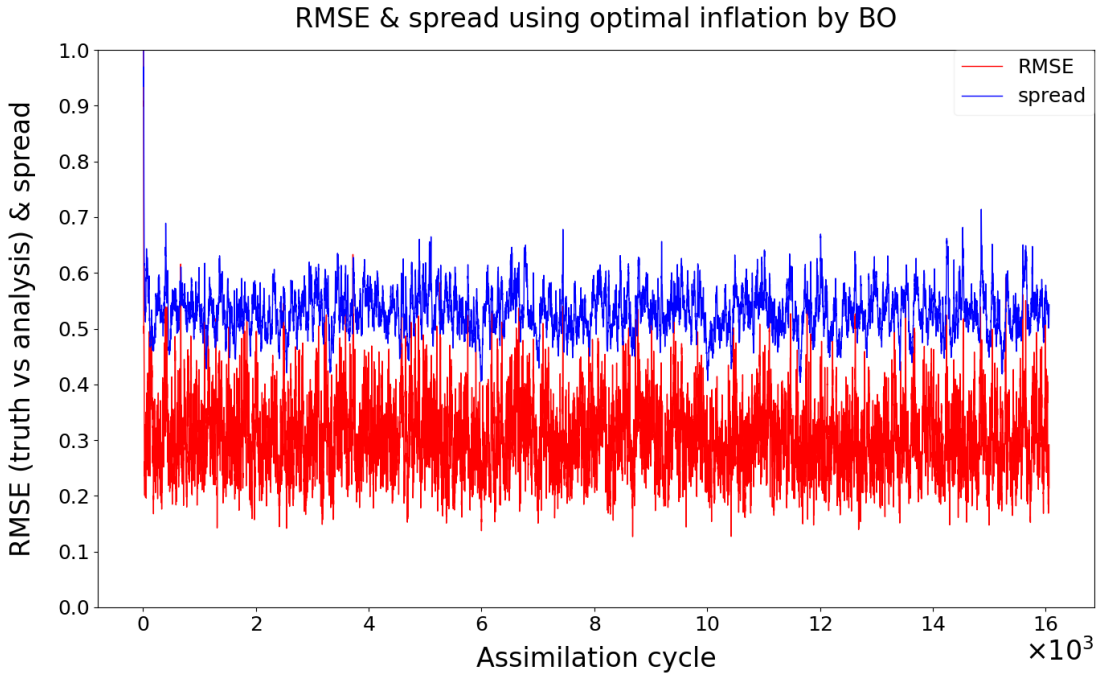


Fig. 6. Summary of estimation results obtained using Bayesian optimization. The red line is the root mean square error (RMSE) between observations and forecasts, the blue line is the inflation factor τ, and the green line is the expected improvement (EI). The dashed black line represents the optimal inflation factor using the brute-force method. The blue dot shows the optimal inflation factor among the 20 training cycles. In Fig. 3, the initial values of the hyper-parameters in the Gaussian kernel were set to $\theta_1, \theta_2, \text{and } \theta_3 = e^{1.0}, e^{0.4}, \text{and } e^{0.1}$, but in Fig. 6, they were set to $\theta_1, \theta_2, \text{and } \theta_3 = 1.0, 1.0, \text{and } 1.0$. Notably, the vertical axis in the expected improvement (EI) has been changed.


(a)
Gaussian process regression (Training cycle = 0)
×10⁻¹
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ
(c)
Gaussian process regression (Training cycle = 12)
×10⁻⁶
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ
(b)
Gaussian process regression (Training cycle = 4)
×10⁻²
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ
(d)
Gaussian process regression (Training cycle = 20)
×10⁻⁸
Training data
Mean
Standard deviation
EI
RMSE (observation vs. forecast)
EI
Inflation factor τ

779 Fig. 7. Estimation result of the relationship between the inflation factor τ and root mean
780 square error (RMSE) (observations versus forecasts) using Gaussian process
781 regression (GPR). This figure corresponds to Fig. 6. The regression and its uncertainty
782 for the RMSE are obtained as the mean and standard deviation (95% confidence
783 interval) of the GPR. The blue line is the mean, the blue shades are the standard
784 deviations, the green line is the expected improvement (EI), and the red dots are the
785 training data. (a)–(d) show the estimation results in the 0th (when only the initial training
786 data are given), 4th, 12th, and 20th training cycles, respectively. In Fig. 4, the initial
787 values of the hyper-parameters in the Gaussian kernel were set to $\theta_1, \theta_2, \text{and } \theta_3 =$
788 $e^{1.0}, e^{0.4}, \text{and } e^{0.1}$, but in Fig. 7, they were set to $\theta_1, \theta_2, \text{and } \theta_3 = 1.0, 1.0, \text{and } 1.0$.
789



790 Fig. 8. Time series of the root mean square error (RMSE) between the truth and analysis
791 and the ensemble spread with the optimal inflation factor $\tau = 0.44$ determined by the
792 Bayesian optimization (BO). The red line is the RMSE and the blue line is the ensemble
793 spread. Note that the horizontal axis is the assimilation cycle, not the training cycle.
794