# Algorithmic tuning of spread–skill relationship in ensemble forecasting systems

Ekblom Madeleine, Tuppi Lauri, Shemyakin Vladimir, Laine Marko, Ollinaho Pirkka, Haario Heikki, Järvinen Heikki



**DOI:** 10.1002/qj.3695

ORIGINAL ARTICLE

# Algorithmic tuning of spread-skill relationship in ensemble forecasting systems

**Madeleine Ekblom[1]* | Lauri Tuppi[1] | Vladimir Shemyakin[2] | Marko Laine[3] | Pirkka Ollinaho[3] | Heikki Haario[2] | Heikki Järvinen[1]**

[1]Institute for Atmospheric and Earth System Research/Physics, University of Helsinki, Helsinki, Finland

[2]School of Engineering Science, Lappeenranta University of Technology, Lappeenranta, Finland

[3]Finnish Meteorological Institute, Helsinki, Finland

**Correspondence**
Madeleine Ekblom, Institute for Atmospheric and Earth System Research/Physics, P.O. Box 64, 00014 University of Helsinki, Finland
Email: madeleine.ekblom@helsinki.fi

**Funding information**
Academy of Finland (project n:o 313828), the Vilho, Yrjö and Kalle Väisälä Foundation of the Finnish Academy of Science and Letters, the Academy of Finland Center of Excellence programme (grant no. 307331), Doctoral Programme in Atmospheric Sciences of University of Helsinki, and the Academy of Finland Centre of Excellence of Inverse Modelling and Imaging (decision number 312122)

In ensemble weather prediction systems, ensemble spread is generated using uncertainty representations for initial and boundary values as well as for model formulation. The ensuing ensemble spread is thus regulated through, what we call, ensemble spread parameters. The task is to specify the parameter values such that the ensemble spread corresponds to the prediction skill of the ensemble mean - a prerequisite for a reliable prediction system. In this paper, we present an algorithmic approach suitable for this task consisting of a differential evolution algorithm with filter likelihood providing evidence. The approach is demonstrated using an idealized ensemble prediction system based on the Lorenz–Wilks system. Our results suggest that it might be possible to optimize the spread parameters without manual intervention.

**KEYWORDS**
Stochastic optimization, filter likelihood, differential evolution algorithm

## 1 | INTRODUCTION

Ensemble prediction is a computationally affordable way to approximate flow-dependent uncertainties of weather forecasts. These uncertainties are due to incomplete knowledge of model initial and boundary conditions as well as inadequacies in model formulation. These imperfections lead to forecast errors that grow with increasing forecast lead time, depending on the state of the atmospheric flow field (e.g. Lewis, 2005; Leutbecher and Palmer, 2008).

Uncertainties in model initial conditions (the analysis error) are in principle provided by data assimilation systems. In Kalman filter, for instance, an estimate of the analysis error is available as a Gaussian error covariance matrix. For multiple reasons (such as approximations in data assimilation, low-resolution analysis vs. high-resolution forecast, selective sampling of the fastest growing initial perturbations, etc.) the analysis uncertainty is not precisely known, hence the amplitudes of initial state perturbations are uncertain.

Sources of uncertainties in model formulation are poorly known. Typically, some stochastic representations of model uncertainties are applied in ensemble prediction models in order to perturb the forecast model dynamics and generate sufficient ensemble spread on top of the spread due to initial state perturbations. To some degree, trial-and-error is used to specify correct intensity for the stochastic model noise.

Ensemble forecasting introduces thus an additional class of unknown parameters which needs to be estimated. Firstly, there are closure parameters in physical parametrization schemes for which the forecast skill is sensitive. Atmospheric observations may directly help to specify a plausible range of the closure parameter values. Secondly, there are the parameters guiding amplitudes of stochastic perturbations of both initial state and model uncertainties (hereafter called "spread parameters", denoted with a vector $\boldsymbol{\Theta}$). The spread parameters are not directly related to any atmospheric observables. Their values must be indirectly inferred by trial-and-error based on the spread properties of the system, or by trying to pose and solve a new parameter estimation problem. It is important to note that both types of parameters must be correctly tuned to have a good ensemble prediction system. Here, we will only focus on the spread parameters and how they affect the forecast skill of an ensemble prediction system.

Solonen and Järvinen (2013) presented a tuning approach for ensemble prediction systems. In this work, we continue their work and present an algorithmic way of solving the parameter estimation problem. The aim is to objectively search for the most likely spread parameter values and find a reliable ensemble prediction system without manual intervention. We do not intend to change the stochastic physics scheme but tune the parameters of an already existing scheme algorithmically. Idealized experiments with the Lorenz – Wilks system indicate that this aim may be attainable. To our knowledge, this is the first time the spread parameters are being estimated using an algorithmic approach. We acknowledge that, on the one hand, it is a formidable computational task. On the other hand, manual tuning of ensemble prediction systems in operational weather prediction centers is a massive effort, too.

The structure of the paper is as follows: Section 2 poses the estimation problem, Section 3 develops the methods, Section 4 describes the implementation using Lorenz–Wilks system, Section 5 presents the results from the experiments, while Sections 6 and 7 discuss and conclude the paper.

## 2 | THE PROBLEM

As a background, we recall that ensemble methods are useful in tuning deterministic model closure parameters, even without embedding the parameter estimation problem into the state estimation problem (or, data assimilation). Such an estimation of model parameters can be based on a notion that deterministic forecast skill of weather prediction models is sensitive to the specified parameter values such that some parameter value combinations score better than

others. It is therefore possible to employ an ensemble forecasting system for on-line estimation of the posterior probability densities of closure parameters (Järvinen et al., 2012; Laine et al., 2012). In such an approach, parameter proposal densities are updated with new evidence emanating from skill scores of individual ensemble members. It is important to note that the skill scores for each ensemble member can be computed immediately after the forecast is completed and verifying observations become available. In the estimation process, score evaluation and density update thus alternate until some convergence is reached.

We now try to extend this line of thinking to the estimation of the spread parameters of ensemble systems. We will employ a cost function based on filter likelihood, as it can be computed from the ensemble simulations. Note that the filter likelihood collects data from several assimilation intervals for each cost function evaluation, and should thus be able to converge with relatively small ensemble sizes.

Here, the optimization target is the spread-skill relationship of the ensemble prediction system with ensemble size $N$. This relationship can be assessed using a sufficiently long sequence of consecutive ensembles. Let such a sequence be $S_M$, where $M$ indicates the number of consecutive ensembles. Let us further assume that there is a likelihood function $\mathcal{L}(\boldsymbol{\Theta})$ that can be computed from $S_M$ and that it is sensitive to the spread-skill relationship such that reliable sequences correspond to high likelihood. From the sequence $S_M$, one can compute the likelihood for one spread parameter vector $\boldsymbol{\Theta}$. To explore the spread parameter space spanned by the elements of $\boldsymbol{\Theta}$, one needs to test the sequence $S_M$ with $K$ different parameter combinations. There is thus a need to generate $K$ sequences $(S_{M,1}, S_{M,2}, \ldots, S_{M,K})$, each testing corresponding parameter vectors $(\boldsymbol{\Theta}_1, \boldsymbol{\Theta}_2, \ldots, \boldsymbol{\Theta}_K)$, resulting in corresponding likelihood values $(\mathcal{L}(\boldsymbol{\Theta}_1), \mathcal{L}(\boldsymbol{\Theta}_2), \ldots, \mathcal{L}(\boldsymbol{\Theta}_K))$. The likelihood values can be used as path-finders in the parameter space in moving towards better parameters and higher likelihoods.

The set-up above would constitute one assessment step aiming to distinguish well-performing parameters from poor parameter values. One such assessment step would thus imply running $K \times M \times N$ individual model forecasts. Figure 1 illustrates this arrangement.

An outline of a sequential and iterative procedure to solve this type of parameter estimation problem is as follows. Assume there is an ensemble prediction system available where the parameters regulating the stochastic perturbation amplitudes can be externally controlled (using a "namelist", for instance). The steps are:

1. Initialize $K$ ensemble systems by uniquely perturbing the default parameter values $\boldsymbol{\Theta}$;
2. For each ensemble system, run an ensemble of size $N$ over the sequence $S_M$;
3. Compute the likelihood $\mathcal{L}(\boldsymbol{\Theta})$ for each $S_M$ using verifying observations;
4. Re-initialize the $K$ ensemble systems using the updated parameter vectors;
5. Return to 2. Continue until the likelihood is insensitive to changes in $\boldsymbol{\Theta}$.

This outline is generic in the sense that it leaves the definition of the likelihood function and the update procedure open. It nevertheless follows ideas of classical insensitivity analysis of finding the region in the parameter space where small inaccuracies in the specification of parameter values do not compromise the usefulness of forecasts.

## 3 | METHODS

This section presents a solution to the problem posed above. In particular, an efficient formulation of the cost function and the update procedure are provided with justification.

## 3.1 | Cost function from filter likelihood

For a proper cost function formulation in the spread parameter tuning problem, we need to be able to calculate the likelihood function. The likelihood function is defined as the probability distribution of the observations given the unknown parameters. Now we are working with ensembles of dynamical models, for which an assimilation system defines the analysis and model predictions. A prototypical assimilation system is based on Kalman filter (Kalman, 1960), and advanced systems can mostly be seen as generalizations or approximations of the basic Kalman filter. For a system of linear equations and Gaussian error models, the classical Kalman formulas provide an efficient way to evaluate the likelihood of an assimilation system. We will show how this idea of the filter likelihood can be used in ensemble system tuning.

A basic assimilation system provides an analysis of the current state by a mean vector and its uncertainty covariance matrix. From the analysis, predictions for the next assimilation time window are calculated using the dynamical model. Let $\mathbf{y}_k^p$ be the model predictions projected on the observations with $\mathbf{y}_k$ being the actual observations, in a new assimilation window. Let $\mathbf{C}_k^y$ be the uncertainty covariance of these one-step-ahead predictions. The Kalman filter likelihood (see, e.g. Hakkarainen et al., 2013) written in a cost function format as twice the negative log likelihood is

$$J(\boldsymbol{\Theta}) = -2\log(\text{likelihood}) = \sum_{k=1}^{n}\left[(\mathbf{y}_k - \mathbf{y}_k^p)^T(\mathbf{C}_k^y)^{-1}(\mathbf{y}_k - \mathbf{y}_k^p) + \log|\mathbf{C}_k^y|\right], \tag{1}$$

where the sum is over the assimilation windows $k$ and $n$ is the total number of assimilation windows. This formula follows directly from the linear Kalman filter theory. Since the unknown parameters are part of the forecast model, this means that both the mean prediction $\mathbf{y}_k^p$ and its covariance $\mathbf{C}_k^y$ are functions of the parameters. Hence, the formula must also include a term with the determinant of the covariance $|\mathbf{C}_k^y|$.

As noted by Solonen and Järvinen (2013), the likelihood, Equation 1, has an intuitive interpretation of balancing the accuracy and precision of the assimilation system predictions. In terms of model ensembles we see that if the spread is too small, then the covariance $\mathbf{C}_y^k$ is small compared to the prediction error $\mathbf{y}_k - \mathbf{y}_k^p$, and the first term $(\mathbf{y}_k - \mathbf{y}_k^p)^T(\mathbf{C}_k^y)^{-1}(\mathbf{y}_k - \mathbf{y}_k^p)$ in the likelihood will be large, penalizing the overall likelihood. On the other hand, if the spread is too large, the first term becomes small, but the second term $\log|\mathbf{C}_k^y|$ becomes large. The maximum likelihood solution is found as a compromise between these two: the prediction error $\mathbf{y}_k - \mathbf{y}_k^p$ needs to be in balance with the ensemble spread $\mathbf{C}_y^k$. This is the key observation that guides us to use the filter likelihood based cost function in ensemble system parameter tuning.

In ensemble Kalman filter assimilation (e.g., Evensen, 2003), it might be possible to evaluate the likelihood in Equation 1 by replacing the analytical covariances by those calculated from the ensemble, especially if the size of the ensemble is larger than the number of observations in one assimilation window. In practice, many assimilation systems approximate the prediction covariance $\mathbf{C}_k^y$ by a diagonal matrix of the corresponding prediction variances. This leads us to the approximate formulation

$$J = \sum_{k=1}^{n}\left[\sum_{l=1}^{L}\left(\frac{(y_l - y_l^p)^2}{\sigma_{y_l}^2 + \sigma_{p_l}^2} + \log(\sigma_{y_l}^2 + \sigma_{p_l}^2)\right)\right], \tag{2}$$

where $y_l^p$ and $\sigma_{p_l}^2$ are prediction mean and variance for the observation number $l$ calculated from the model propagated ensemble and $y_l$ and $\sigma_{y_l}^2$ are the corresponding observation and its error variance. Note that the cost function is a sum over both assimilation windows $k = 1, \ldots, n$ and over observations $l = 1, \ldots, L$.

In ensemble prediction systems, the initial state perturbations are derived from the related data assimilation sys-

tem. In addition, some model noise is added. Thus, although an ensemble prediction system is not directly driven by Kalman filter type data assimilation, it aims to have the same balance between predictions and their uncertainties as is coded in the filter likelihood formula. In the following, we will adopt the ensemble based filter likelihood cost function using Equation 2 to optimize the ensemble spread parameters. The approximate maximum likelihood solution is searched and verified using traditional ensemble verification metrics to check that high likelihood indeed corresponds to good verification metrics.

## 3.2 | Differential evolution

An approximate maximum likelihood solution for the spread parameters is computationally intensive. As seen above, one evaluation step would imply a need for running $K \times M \times N$ forecasts. Therefore, as few steps as possible should be taken in the likelihood optimization. Additionally, the target (the filter likelihood) is stochastic; the same $\boldsymbol{\Theta}$ does not result in the same likelihood $\mathcal{L}(\boldsymbol{\Theta})$ in consecutive sequences $S_M$ because of the stochasticity of the prediction system, and especially when each $S_M$ covers a new epoch (i.e., the time and data covered by one sequence $S_M$; when the sequence covers a new epoch, the state of the atmosphere is also new). The latter is of course useful in realistic multi-scale multi-phase optimization cases when the system has inherently rich dynamics, seasonal variations, varying boundary conditions, and so forth, in order to cover the climatology as thoroughly as possible.

Differential Evolution optimizer (Storn and Price, 1997, DE) is applied here. DE utilizes ideas of population-based algorithms and it is easy to use in ensemble-based approaches with a stochastic target. DE is based on a concept of population and heuristically follows the process of natural selection. The DE algorithm proceeds by having a population of candidate solutions $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K$ for which the likelihood values $\mathcal{L}(\boldsymbol{\Theta}_1), \ldots, \mathcal{L}(\boldsymbol{\Theta}_K)$ are computed. A new trial population is generated with some simple and heuristic formula which aim to slightly shift the positions of the elements of each $\boldsymbol{\Theta}_i$ in the parameter space. If a trial position is an improvement over the candidate position, the trial position supersedes the candidate position. If a trial position is less likely than the candidate position, it is discarded. After some steps, the fittest dominate the population and the likelihoods become insensitive to small position shifts.

It should be noted that in DE there is no guarantee for finding the optimal solution. In our tests the DE performance was, however, satisfactory. Besides the standard steps of the DE algorithm, Shemyakin and Haario (2018) use two more methods to ensure sufficient diversity of parameter population: generation jumping and recalculation step, where the former one was previously available in literature and the latter one a technique introduced in the paper. These two additional steps help to prevent the algorithm from getting stuck in a local minimum. Details about the different steps and their schemes are found in the Appendix.

## 3.3 | Spread parameter estimation method

We now present our proposal to solve the spread parameter estimation problem. The aim is to optimize the parameter vector $\boldsymbol{\Theta}$ consisting of parameters regulating the perturbation amplitudes of both the initial state and the prediction model. The solution method uses the DE algorithm as an optimizer and the filter likelihood based cost function to provide evidence of the spread parameters. Here, the population consists of the parameter vectors $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K$. The current generation is evaluated over the present sequence $S_M$, and the optimization continuously evolve towards new generations (and possibly new epochs) where the most promising vectors $\boldsymbol{\Theta}_k$ survive and replace less fit vectors. The method works as follows:

**1.** Population initialization: Based on user-specified parameter search boundaries for each individual parameter $\theta_i$

of vector $\boldsymbol{\Theta}$, sample a candidate population of size $K$: $\boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_K$

2. Calculate the cost function for the candidate population:
   a. Forecast step: For each parameter vector $\boldsymbol{\Theta}_k$, run $M$ consecutive ensembles of size $N$ (over one epoch to create $S_M$)
   b. Evaluation step: For each $\boldsymbol{\Theta}_k$, calculate $\mathcal{L}(\boldsymbol{\Theta}_k)$ using Equation 2 with all observations collected in the sequence $S_M$
3. Population update: Using the DE formula, create a trial population based on the candidate population
4. Calculate the cost function for the trial population (as in steps 2a and 2b)
5. Population selection:
   a. For all population members, compare the cost function values of the candidate and the trial population, and select the one with the lower cost function value
   b. Supersede the candidate population with the surviving members thus forming the new candidate population
   c. Return to step 3. Continue until a stopping criteria is fulfilled (a given number of steps is reached or the cost function becomes insensitive to population updates).

## 4 | IMPLEMENTATION USING THE LORENZ – WILKS SYSTEM

An idealized ensemble prediction system is used here to demonstrate the spread parameter estimation. In the experiment set-up, synthetic observations are used for evaluation of the forecast data, both in the likelihood evaluation and in the verification metrics.

### 4.1 | Lorenz95 system

Lorenz95 model (Lorenz, 1996; Palmer and Hagedorn, 2006, pp. 40-58) is an idealized non-linear model useful in numerical experimentation. The full Lorenz95 model is described as

$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - x_k + F + (hc/b)\sum_{j=1}^{J} U_{j,k}; \tag{3a}$$

$$\frac{dU_{j,k}}{dt} = -cbU_{j+1,k}(U_{j+2,k} - U_{j-1,k}) - cU_{j,k} + F_u + (hc/b)X_k. \tag{3b}$$

It contains both slow state variables $X_k$, which represent the large scale, and fast variables $U_{j,k}$ representing sub-grid scale processes.

Wilks (2005) developed a version of the Lorenz95 model, where the net effect of the fast variables is parametrized

$$\frac{dX_k}{dt} = -X_{k-2}X_{k-1} + X_{k-1}X_{k+1} - x_k + F - g(X_k) + \eta_k(t); \tag{4a}$$

$$g(X_k) = \sum_{i=0}^{B-1} b_i X_k^i; \tag{4b}$$

$$\eta_k(t + \Delta t) = \phi\eta_k(t) + \sigma_e(1 - \phi^2)^{1/2} z_k(t). \tag{4c}$$

Here $g(X_k)$ represents the deterministic net effect of the unresolved fast variables and $\eta_k(t)$ the stochastic forcing accounting for model errors (in this case: deficiencies in the deterministic parametrization). The stochastic term uses $z_k(t) \sim \mathcal{N}(0,1)$, $\sigma_e$ determines the magnitude of the stochastic term, and $\phi$ is the degree of the lag-1 autocorrelation of the stochastic forcing. The stochastic physics scheme used in this case has two special cases: when $\phi = 0.0$ and $\phi = 1.0$. The case when $\phi = 1.0$ leads to the stochastic physics being constant $\eta_k(t + \Delta t) = \eta_k(t)$. This scheme is completely dependent on the initialization of $z_k(t)$. The second case is when $\phi = 0.0$, which leads to the stochastic physics scheme $\eta_k(t + \Delta t) = \sigma_e z_k(t)$.

In our ensemble prediction system, $\sigma_e$ and $\phi$ are the spread parameters related to the model uncertainty. We will denote the system by "Lorenz–Wilks system" to clarify that both the original and the parametrized version of the Lorenz95 system are used.

## 4.2 | Experiment set-up

The full model, Equation 3, is used to generate the truth and synthetic observations, which directly represent the slow state variables. Three of every five true state is observed (i.e., 24 out of 40 chosen as three variables of five for eight blocks) and Gaussian noise with zero mean and standard deviation 0.35 is added to the true states to get the synthetic observations. The parametrized model, Equation 4, serves as the forecast model in the ensemble predictions. Ensemble Kalman filter (Evensen, 2003; Houtekamer and Mitchell, 2005, EnKF) is used in data assimilation to generate model initial states and their error variance estimates. Data assimilation is performed off-line using the deterministic part of Equation 4 to avoid stochastic effects on the analysis. An ensemble size of 200 was used to get a good approximation of the covariance of the analysis. The observation error matrix was set to $0.35^2\mathbf{I}$ and the model error matrix to $0.4^2\mathbf{I}$.

The initial states for the ensemble predictions are generated using the following scheme

$$X_{k,0} = X_{k,a} + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \lambda \cdot \mathrm{var}(X_{k,a})), \tag{5}$$

where $X_{k,a}$ is the analysis for state $k$ and $\mathrm{var}(X_{k,a})$ its corresponding variance, both obtained from the data assimilation. For each state $k$ in Equation 4, the initial values $X_{k,0}$ are centered around the analysis $X_{k,a}$ and the initial value perturbation $\epsilon$ is drawn from a Gaussian distribution with zero mean and variance defined by the product of the the diagonal of the error covariance matrix and the initial spread parameter $\lambda$. The parameter $\lambda$ regulates the amplitude of the error variance (and hence the initial value perturbations) and is our third spread parameter. In our system, $\lambda = 1$ would imply that the initial spread obtained from the data assimilation system is correct, and this is what we expect.

We wish to know the optimal values of the three ensemble spread parameters: initial spread parameter $\lambda$, stochastic physics amplitude parameter $\sigma_e$, and stochastic auto-correlation parameter $\phi$. The experiment is divided into two parts: (1) mapping of the cost function in the spread parameter space, and (2) optimizing the three spread parameters using the proposed method. The former is a brute force approach where the parameter vectors are pre-selected allowing one to hand-pick the most likely ones, whereas the latter follows the previously described solution method. The ensemble forecasts of one likelihood evaluation are computed the same way in both cases:

1. Each ensemble has a unique parameter vector $\boldsymbol{\Theta} = (\lambda, \sigma_e, \phi)$;
2. The ensemble size is $N$ and each member is initialized using Equation 5; the forecast length 2.0 time units with output every $\mathrm{d}t$.
3. Ensembles are run using the stochastic model, Equation 4, over $S_M$; the ensembles within an epoch are consec-

utive, whereas the epochs $S_M$ cover the same data in the mappings and are consecutive in the optimization part leading to completely new data introduced every algorithm step.

In the first part, where we look at the mapping of the parameter space, we will use an ensemble of size 20 and the length of the sequence $S_M$ is 10. We will look at three different output frequencies: d$t$=0.1 time units, d$t$=0.2 time units, and d$t$=0.4 time units. Since the forecast length is 2.0 time units, this means that the number of observations included in the calculation of the cost functions depend on the number of assimilation windows $2.0/dt$, see Equation 2 for details. Here, all sequences $S_M$ run over the same epoch.

In the optimization part, we will look at how changes in the ensemble size ($N$), the population size ($K$), and the length of the sequence ($M$) affect the convergence of the DE algorithm for three different output frequencies (d$t$=0.1, 0.2, or 0.4 time units). To get a more realistic set-up, the sequences are consecutive leading to new data being introduced at every algorithm step. We will run three reference runs (one for each output frequency) with an ensemble size of 20, a population size of 20, and a sequence of 10 consecutive ensembles. We will then run similar tests with reduced amount of resources (smaller ensemble, smaller DE population, and shorter sequence $S_M$) and then compare those results to the reference runs.

| $K = 10$ | $M = 1$ | $M = 5$ | $M = 10$ | | $K = 20$ | $M = 1$ | $M = 5$ | $M = 10$ |
|---|---|---|---|---|---|---|---|---|
| $N = 5$ | A: 50 | B: 250 | C: 500 | | $N = 5$ | D: 100 | E: 500 | F: 1000 |
| $N = 10$ | - | - | - | | $N = 10$ | G: 200 | H: 1000 | I: 2000 |
| $N = 20$ | - | - | - | | $N = 20$ | J: 400 | K: 2000 | L: 4000 |

**TABLE 1** The table shows the name of the tests (A-L) and the total number of forecast runs per algorithm step for different combinations of ensemble size ($N$), population size ($K$), and length of sequence ($M$). For example, the cell containing "A:50" refers to test A with 50 forecast runs per step. We will denote the most expensive test, test L, *reference test*.

Table 1 gives an overview of the different tests and how many forecast runs one algorithm step requires. During one algorithm step, a population of $K$ members each being associated with an ensemble of size $N$ for a sequence of length $M$. Hence, the product of $K$, $M$, and $N$ gives the total number of forecast runs. The higher the total number of forecast runs, the more expensive is the set-up described. As an example, the most expensive test, test L, needs 80 times more forecast runs than the cheapest test, test A.

The set-up details are found in the Appendices A.1 and A.2.

## 4.3 | Verification metrics

To verify the goodness of the ensemble prediction systems, we use traditional ensemble forecast verification methods: (1) spread-skill relationship, (2) rank histogram, and (3) continuous ranked probability score.

1. The error of the ensemble mean and the ensemble spread should be in balance in an optimal ensemble prediction system. The relation between spread and skill is described as (e.g. Leutbecher and Palmer, 2008)

$$\frac{1}{M}\sum_{j=1}^{M}\left(\epsilon_j^2 - \frac{N+1}{N-1}s_j^2\right) \to 0, \quad \text{for} \quad M \to \infty \tag{6}$$

where $M$ is the length of the ensemble sequence, $N$ is the size of the ensemble, $\epsilon_j^2$ is the mean squared error of ensemble mean, and $s_j^2$ is the ensemble variance. The term $(N+1)/(N-1)$ is a correction factor for using finite ensembles. We will use the term *skill-spread difference* to denote the difference between the error of the ensemble mean and the ensemble spread, where the spread is multiplied by the square root of the correction factor, $\sqrt{(N+1)/(N-1)}$.

2. A rank histogram shows the reliability of an ensemble prediction system (Hamill, 2001). The rank of a single ensemble forecast is the order of the observation compared to the sorted ensemble (a number between 1 and N+1) and the rank histogram the corresponding histogram of rank numbers from many ensemble forecasts. It is desirable to have a flat rank histogram while a U- or A-shape points to under- or over-dispersive systems. Using a rank histogram alone may be misleading and it should therefore be used together with other verification metrics.
3. Continuous ranked probability score (Hersbach, 2000, CRPS) tells how well the probability distribution of the ensemble forecast matches the probability distribution of the observation. CRPS is defined as

$$CRPS = \int_{-\infty}^{\infty} (P(x) - P_a(x))^2 dx, \tag{7}$$

where $P(x)$ and $P_a(x)$ are the cumulative probability distributions of the ensemble forecast and the observation, respectively. The smaller the CRPS value, the closer are the ensemble members located around the observation and the better is the ensemble forecast.

When calculating the verification metrics, the ensemble data is compared to the synthetic observations.

## 5 | RESULTS FROM LORENZ–WILKS DEMONSTRATIONS

The results consist of two parts: (1) mapping of the parameter space, and (2) optimizing the three spread parameters, both presented with the verification results.

### 5.1 | Mapping of the parameter space

The mapping of the parameter space was computed by running ensembles using 1000 pre-generated parameter vectors $\boldsymbol{\Theta} = (\lambda, \phi, \sigma_e)$. The parameter vectors were drawn from uniform distributions, where the lower and upper limits were 0.1 and 10.1 for $\lambda$, 0.0 and 1.0 for $\phi$, and 0.0 and 2.5 for $\sigma_e$. Here, we will show two types of mappings: the mapping of the $(\lambda, \sigma_e)$-space with $\phi$ fixed at 0.5 and $(\sigma_e, \phi)$-space with $\lambda$ fixed at 1.0. For each parameter vector, a cost function value is calculated as the mean of the -2log likelihood over 10 consecutive ensembles. The verification metrics are calculated in a similar manner, i.e., as the mean over the 10 consecutive ensembles.

Figures 2a, 2b, and 2c map the filter likelihood for the initial spread parameter $\lambda$ and the stochastic physics amplitude parameter $\sigma_e$ for three different observational distributions. In Figure 2a the output frequency is every 0.2 time units, in Figure 2b 0.4 time units, and in Figure 2c with 0.1 time units. The smaller the output frequency, the more observations are included in the calculation of the cost function. As the cost function is a sum over the entire forecast range, the output frequency affects the value of the cost function. An area of minimum values is found at and around $\lambda$ being equal to unity in all three cases. The area of small cost function values is slightly tilted such that the higher the value of $\sigma_e$ is, the smaller is the value of $\lambda$. In other words, when the initial spread is small, a high amplitude of stochastic physics is needed to maintain high likelihood, and *vice versa*. The changes in the temporal distributions

of the observations do not show any distinct changes in the parameter space.

Figure 2d shows the $(\sigma_e, \phi)$-space with the initial spread parameter $\lambda$ fixed at unity. The topology of the $(\phi, \sigma_e)$-space is rather flat with a diagonal valley in the middle such that if one parameter is small, the other needs to be large. However, cost function differences among the parameter combinations are small. The mapping of $\phi$ and $\sigma_e$ shows that this ensemble prediction system is less sensitive to the choices made in the stochastic physics scheme than in the initial value perturbations.

The verification metrics of the mapping in Figure 2a were computed for all parameter vectors, shown in Figure 3. The vectors corresponding to the ten lowest cost function values are marked with black to highlight their performance in the context of verification metrics. Those systems have the smallest CRPS values (Figure 3a), spread-skill differences close to zero (Figure 3b), and quite flat rank histograms (Figure 3c). Similar results as those in Figure 3 are achieved for the other mappings of Figure 2, but those plots are not shown here. Hence, a small cost function value indicates a good ensemble performance. This is expected based on Solonen and Järvinen (2013).

It is worth noting that the verification metrics in Figure 3 are calculated only over ten consecutive ensembles and that longer sequences would give less noisy results. Based on the maps in Figure 2 one can anticipate better identifiability for the parameter $\lambda$ than for $\phi$ or $\sigma_e$.

## 5.2 | Parameter optimization

The mapping of the parameter space shows that there is a minimum in the cost function that corresponds to well performing systems. Since all three ensemble spread parameters, $\lambda, \phi,$ and $\sigma_e$, are important for the spread-skill relation, we will optimize all three parameters at once.

The DE algorithm was initialized using the same lower and upper limits for $\lambda$, $\phi$, and $\sigma_e$ as in the mappings. Figure 4 shows the parameter evolution as a function of algorithm steps for the most expensive test, test L. The tests differ from each other on the temporal distribution of the observations: Figure 4a has an output every 0.2 time units, Figure 4b every 0.4 time units, and Figure 4c every 0.1 time units. Comparing these figures shows that the changes in temporal distribution of observations have a small impact on the convergence of the DE. The initial spread parameter $\lambda$ converges fast, but the two stochastic physics parameters do not have such a clear minimum, as was previously demonstrated in Figure 2d.

Figure 4 also exhibits patterns resembling a row structure. This is caused by the way DE works: it does not often mutate all parameters at the same time but favors a mutation of only one parameter at a time. Therefore, parameters often get the same values on consecutive iterations. Moreover, DE tends to optimize the most sensitive parameter first, and then move on to less sensitive parameters one at a time. The reason for this is that the effect of variations of the most sensitive parameter dominates in the cost function. Despite these features, DE is able to converge also with the least sensitive parameters as can be best seen in Figure 4b.

Figure 5 shows six different plots where the resources are reduced. Figures 5a, 5b, and 5c show results with experiments having an ensemble of size 10, a population of size 20, and the length of the sequence is 10, 5, and 1, respectively. Figures 5d, 5e, and 5f have an ensemble of size 5, a population of size 10, and the length of the sequence is 10, 5, and 1, respectively. When reducing the length of the sequence, the cost function becomes more stochastic. Reducing the ensemble size does also affect the calculation of the cost function, whereas a smaller population size has a direct effect on the optimizer itself. In all these cases, the algorithm shows steady convergence, especially for the initial spread parameter $\lambda$. However, the algorithm seems more stable when the DE population and the ensemble are of larger sizes.

When comparing two tests with different set-ups, it is worth looking at the total amount of forecast runs, calcu-

lated as the product of ensemble size, population size, length of sequence, and the number of algorithm steps. Figure 6 compares the cheapest test, test A, and the most expensive test, test L, both with output frequency every 0.1 time units. The plots show the distribution of the parameter vectors suggested by the DE algorithm for the two tests in three two-dimensional spaces: $(\lambda, \sigma_e)$, $(\lambda, \phi)$, and $(\sigma_e, \phi)$. Based on the distribution of the parameter vectors, the lines are drawn at the minimum and maximum of the parameters in the given space. Figure 6a shows the results when both test A and test L have used resources corresponding to 4000 forecast runs. For test L this equals to 1 step, whereas test A is run for 80 steps. Figure 6b shows similar results when both tests are run for 80 steps: 320000 forecasts are run for test L, but only 4000 for test A. After 80 steps, the distributions of the parameters are very similar for test A and test L. Hence, when running the algorithm over a long period, the algorithm also finds the area of minima for a test using less resources. For a complete picture of the convergence of test A and test L, see Figures 5f and 4c.

Figure 7 compares the verification metrics of ensemble systems with parameters suggested by the DE algorithm corresponding to test A. In this case, an ensemble of size 5, a population of size 10, and a sequence of length 1 were used in the algorithm part. The verification metrics are computed with an ensemble size of 20 and a sequence of length 10 run over the same epoch. The verification metrics show that the parameters suggested by the algorithm indeed correspond to good ensemble systems. The verification metrics also show that the ensembles tend to lead to an under-dispersive system at longer forecast ranges (which seems to be the case for nearly all tested parameter vectors, cf. Figure 3). It also shows that reducing the total amount of forecast runs still gives good results when looking at the verification metrics. The noisiness in the verification metrics is explained by the relatively short sequence.

## 6 | DISCUSSION

It is worth noting that the estimation of spread parameters with ensemble based methods demands $K \times M$ times the amount of computations compared with model physics parameters, assuming that both problems converge at the same rate. In any practical problem the parameters $K$ and $M$ should therefore be as small integers as possible. Also, for the generality of the result, the sequences $S_M$ (see Figure 1) should cover new epochs. The problem with the Lorenz95 model is that the dynamics is relatively simple and remains very similar from one epoch to the next, and there are no seasonal variation, for instance. The results presented here must therefore be confirmed with more realistic models. The key question for the future is thus: what are the convergence properties of this estimation problem, and how small can $K$, $M$ and $N$ be in realistic set-ups.

The choice of the cost function is essential. Here, a filter likelihood approach is applied because it takes into account the relationship between ensemble spread, skill of ensemble mean, and prediction errors. In filter likelihood, summation runs over all assimilation windows which has an advantage that it takes into account the entire forecast range, and a disadvantage as the ensemble spread-skill relationship may be dependent on the forecast range which will be reflected in the filter likelihood sum.

The length of the sequence $S_M$ is an important factor of the cost function and the estimation method. The length affects the stochasticity of the cost function; if the sequence is short, the cost function is more stochastic, especially if data is renewed. On the other hand, a long sequence increases the computational costs. It is therefore important to carefully choose the length $M$ such that the problem remains computationally feasible while not increasing the stochasticity too much. Another important factor of the method is the convergence of the DE algorithm. It seems, based on these experiments, that it is quite important to understand the parameter space and how the cost function behaves. If the cost function becomes too stochastic, the algorithm will have problems finding the optimal parameter vector. When comparing the plots in Figure 5, we see that the convergence is more steady when more resources are

used, which could be explained by the cost function and its stochasticity. Reducing the resources does however not rule out convergence, but could affect the stability of the method. This is important to understand when studying more complex models than the Lorenz–Wilks system.

Another approach is to construct the filter likelihood cost function for the parameters using several assimilation windows, and optimize it with respect to the parameters using iterative algorithms (see, e.g. Särkkä, 2013). The background motive of the presented work is, however, to develop methods for operational platforms where such approaches are not available: iterative cost function calls are ruled out due to computational issues and the assimilation is not performed by filtering but by an assimilation method unique for the prediction system (e.g., 4DVAR), combined with ensemble prediction system simulations. Hence, there are similarities to the idealized Lorenz–Wilks system, which make the method an option in optimizing parameters regulating spread-skill relationships in a realistic model set-up. However, the step from the idealized model to more realistic models is not taken here.

As manually tuning an ensemble prediction system is a resource consuming process, an algorithmic way of tuning the ensemble systems would be welcome. We hope that our described method could be a future tool for model developers for reducing the amount of human resources put on trial-and-error tuning of ensemble systems.

## 7 | CONCLUSIONS

The purpose of this study was to show the road to algorithmic optimization of ensemble spread parameters in an ensemble forecast system. We used an approach based on previous results, where the filter likelihood is used as a cost function in the evaluation of an ensemble prediction system (Solonen and Järvinen, 2013), and added an optimizer to find the optimal combination of ensemble spread parameters. We demonstrated the method with a Lorenz–Wilks ensemble prediction system with three ensemble spread parameters: initial spread parameter $\lambda$, stochastic amplitude parameter $\sigma_e$, and stochastic auto-correlation parameter $\phi$.

First, we demonstrated the choice of the filter likelihood-based cost function by mapping the parameter space. The cost function was plotted as a function of the ensemble spread parameters in two different projections of the three-dimensional parameter space. We demonstrated this for three different temporal distributions of observations: 0.1 time units, 0.2 time units, and 0.4 time units. For all parameter vectors corresponding verification metrics were calculated and the best parameter vectors with respect to the cost function showed good verification metrics (flat rank histogram, spread-skill difference close to 0, small CRPS values) resulting in good ensemble prediction systems. This was demonstrated for the case with an output frequency of 0.2 time units. This verifies that filter likelihood is a viable optimization target of ensemble systems. Based on this, we hypothesized that an algorithmic optimizer could find the region of minimum filter likelihood corresponding to parameter values with good verification scores.

The hypothesis was tested using a differential evolution algorithm (Shemyakin and Haario, 2018) as an optimizer and the filter likelihood method as a cost function. The cost function was calculated over a sequence of ensemble runs. We looked at how the ensemble size, the length of the sequence, and the population size affect the convergence of the algorithm. Reducing the amount of resources affect the convergence of the DE algorithm as it seems to become less stable, but does not rule out convergence. The results indicate that it is possible to optimize ensemble spread parameters in idealized systems without manual intervention. Traditional verification methods confirm that the parameter values suggested by the algorithm lead to good ensemble prediction systems.

We conclude that an algorithmic approach for tuning an ensemble prediction system is possible in an idealized case, such as the Lorenz–Wilks system. Scaling the results from the Lorenz–Wilks system to more realistic set-ups is left for future work.

## Acknowledgements

This work was supported by Academy of Finland (project n:o 313828), the Vilho, Yrjö and Kalle Väisälä Foundation of the Finnish Academy of Science and Letters, the Academy of Finland Center of Excellence programme (grant no. 307331), Doctoral Programme in Atmospheric Sciences of University of Helsinki, and the Academy of Finland Centre of Excellence of Inverse Modelling and Imaging (decision number 312122). Figures 2-7 are plotted with the help of Python Matplotlib library (Hunter, 2007).

# A | APPENDIX

## A.1 | List of symbols

| Symbol | Explanation | Value in tests |
|---|---|---|
| $S$ | sequence of ensembles | - |
| $\mathcal{L}(\cdot)$ | likelihood function | - |
| $\boldsymbol{\Theta}$ | Vector of ensemble spread parameters | $\dim(\boldsymbol{\Theta}) = 3$ |
| $K$ | Population size | 10 or 20 |
| $M$ | Length of an ensemble sequence ("epoch") | 1, 5, or 10 |
| $N$ | Ensemble size | 5, 10, or 20 (200 in EnKF) |
| $\mathbf{y}_k$ | Observation for $k$th assimilation window | |
| $\mathbf{y}_k^p$ | Model prediction for $k$th assimilation window | |
| $\mathbf{C}_k^y$ | Prediction error covariance matrix | |
| $n$ | Number of assimilation windows | 5, 10, or 20 |
| $L$ | Number of observations | 24 |
| $X_k$ | L95 state variable | - |
| $U_{j,k}$ | L95 fast variable | - |
| $J$ | Number of fast variables | 8 |
| $K$ | Number of slow state variables | 40 |
| $\eta$ | Stochastic process in Lorenz-95 | - |
| $\phi, \sigma_e$ | Stochastic physics parameters | - |
| $\lambda$ | Initial value spread parameter | - |
| $F, F_u$ | Forcing in L95 system | 10, 10 |
| $c, b, h$ | Parameters in L95 full system | 10,10, 1 |
| $\mathbf{E}, \mathbf{R}$ | Model error and observation error matrices in EnKF | $0.4^2\mathbf{I}, 0.35^2\mathbf{I}$ |
| $\Delta t$ | Integration time | 0.0025 in Eq. 3, 0.025 in Eq. 4 |
| $\mathrm{d}t$ | Output time | 0.1, 0.2, or 0.4 |
| $g(X_k)$ | Deterministic scheme in Lorenz95 model | $b_0 = 2.0, b_1 = 0.1$, the rest 0 |

## A.2 | Differential evolution

Differential evolution conducts optimization by sequentially employing four main steps initialization, mutation, crossover and selection, to a population:

1. Initialization: For all $x_{j,i,0} = r_j(b_{j,u} - b_{j,l}) + b_l$, where $b_{j,l}$, $b_{j,u}$ are lower and upper boundaries for the $j$th parameter, $r_j \sim \mathcal{U}(0,1)$. $j$ denotes parameter, $i$ denotes population member, 0 denotes generation.
2. Mutation: The population $\mathbf{x}$ is mutated

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F(\mathbf{x}_{r_1,g} - \mathbf{x}_{r_2,g}),$$

   where $F_{j,i,g} = (F_l + r_g(F_u - F_l))(1 + \delta(r_j - 0.5))$ determines the convergence rate. Here, $r_1, r_2, r_g, r_j$ are random numbers. Note that $F$ normally is constant, but here it is randomized within given limits.
3. Crossover: Only a given number of mutations survive. This happens with a pre-defined probability $C_r$

$$u_{j,i,g} = \begin{cases} v_{j,i,g} \text{ if } r_j \leq C_r \text{ or } j = j_r \\ x_{j,i,g} \text{ otherwise.} \end{cases}$$

4. Selection: This step determines whether the mutated or the current member of the population survives.

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} \text{ if } f(\mathbf{u}_{i,g}) \leq f(\mathbf{v}_{i,g}) \\ \mathbf{v}_{i,g} \text{ otherwise.} \end{cases}$$

   Note that the previous population is stored and only the cost function of the trial population is calculated. This saves CPU time.

In our study, we used the following set-up: the scale rate for the mutation was $F = 0.5$. The scale factor was randomized using the parameters $F_l = 0.5, F_u = 1.0$, and $\delta = 0.001$, crossover occurs with a probability $C_r = 0.1$, generation jumping occurred with a probability $J_p = 0.1$, and recalculation occurred at steps 5, 10, 25, 50, and 75. In the experiments, the mutation type was 2 and the scale factor type 5.

## references

Evensen, G. (2003) The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, **53**, 343–367.

Hakkarainen, J., Solonen, A., Ilin, A., Susiluoto, J., Laine, M., Haario, H. and Järvinen, H. (2013) A dilemma of the uniqueness of weather and climate model closure parameters. *Tellus A: Dynamic Meteorology and Oceanography*, **65**, 20147.

Hamill, T. M. (2001) Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, **129**, 550–560.

Hersbach, H. (2000) Decomposition of the continuous ranked probability score for ensemble prediction systems. *Weather and Forecasting*, **15**, 559–570.

Houtekamer, P. L. and Mitchell, H. L. (2005) Ensemble Kalman filtering. *Quarterly Journal of the Royal Meteorological Society*, **131**, 3269–3289.

Hunter, J. D. (2007) Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, **9**, 90–95.

Järvinen, H., Laine, M., Solonen, A. and Haario, H. (2012) Ensemble prediction and parameter estimation system: the concept. *Quarterly Journal of the Royal Meteorological Society*, **138**, 281–288.

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

Laine, M., Solonen, A., Haario, H. and Järvinen, H. (2012) Ensemble prediction and parameter estimation system: the method. *Quarterly Journal of the Royal Meteorological Society*, **138**, 289–297.

Leutbecher, M. and Palmer, T. N. (2008) Ensemble forecasting. *Journal of Computational Physics*, **227**, 3515–3539.

Lewis, J. M. (2005) Roots of ensemble forecasting. *Monthly Weather Review*, **133**, 1865–1885.

Lorenz, E. N. (1996) Predictability: A problem partly solved. In *Proc. Seminar on predictability*, vol. 1.

Palmer, T. and Hagedorn, R. (2006) *Predictability of weather and climate*. Cambridge University Press.

Särkkä, S. (2013) *Bayesian filtering and smoothing*, vol. 3. Cambridge University Press.

Shemyakin, V. and Haario, H. (2018) Online identification of large-scale chaotic system. *Nonlinear Dynamics*, 1–15.

Solonen, A. and Järvinen, H. (2013) An approach for tuning ensemble prediction systems. *Tellus A: Dynamic Meteorology and Oceanography*, **65**, 20594.

Storn, R. and Price, K. (1997) Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, **11**, 341–359.

Wilks, D. S. (2005) Effects of stochastic parametrizations in the Lorenz'96 system. *Quarterly Journal of the Royal Meteorological Society*, **131**, 389–407.

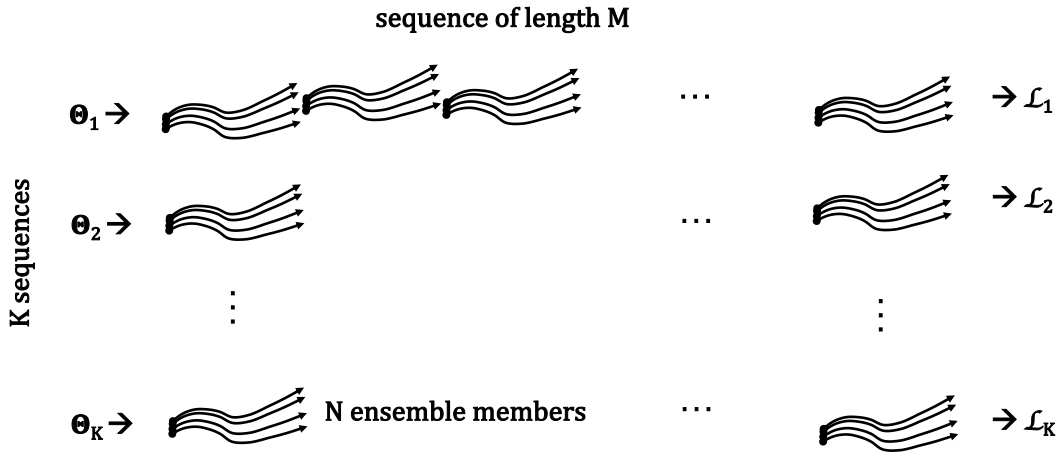



**FIGURE 1** An illustration of one assessment step, and how the $K$ times $M$ times $N$ forecasts are thought to be arranged. For each sequence $S_{M,k}$, the parameter vector $\boldsymbol{\Theta}_k$ is the input and the likelihood value $\mathcal{L}_k$ the corresponding output.

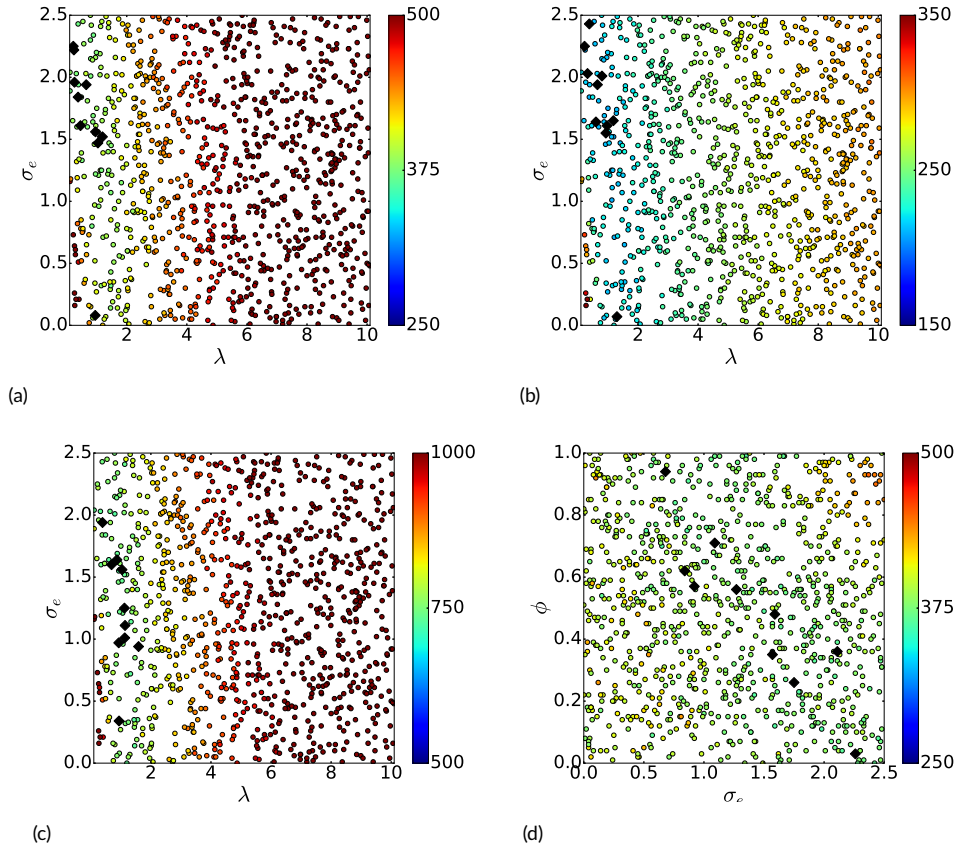



**FIGURE 2** Mapping of the parameter space using ensemble size $N = 20$ and sequence length $M = 10$; (a)-(c) show the -2log likelihood as a function of parameters $\lambda$ and $\sigma_e$ with $\phi = 0.5$ for experimental set-ups with different temporal distribution of observations (a) d$t$=0.2 time units, (b) d$t$=0.4 units, and (c) d$t$=0.1 time units. (d) shows the mapping of the $(\sigma_e, \phi)$-space with $\lambda = 1$ for d$t$=0.2 time units. The ten smallest -2log likelihood values are marked with black diamonds. Note the different color scales in (a)-(d).

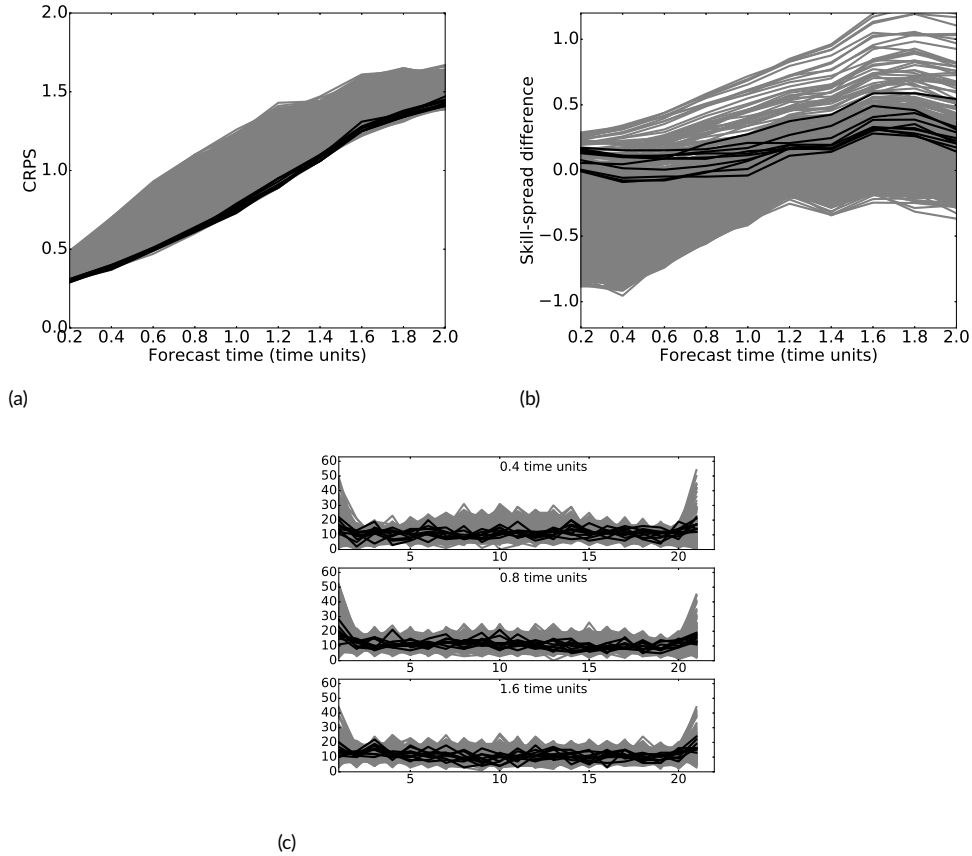



**FIGURE 3** Corresponding verification metrics of the mapping from Figure 2a, where the gray lines show all different parameter vectors and the black lines correspond to the ten smallest -2log likelihood, marked with black diamonds in Figure 2a. (a) CRPS, (b) skill-spread difference, and (c) rank histogram after 0.4, 0.8, and 1.6 time units.

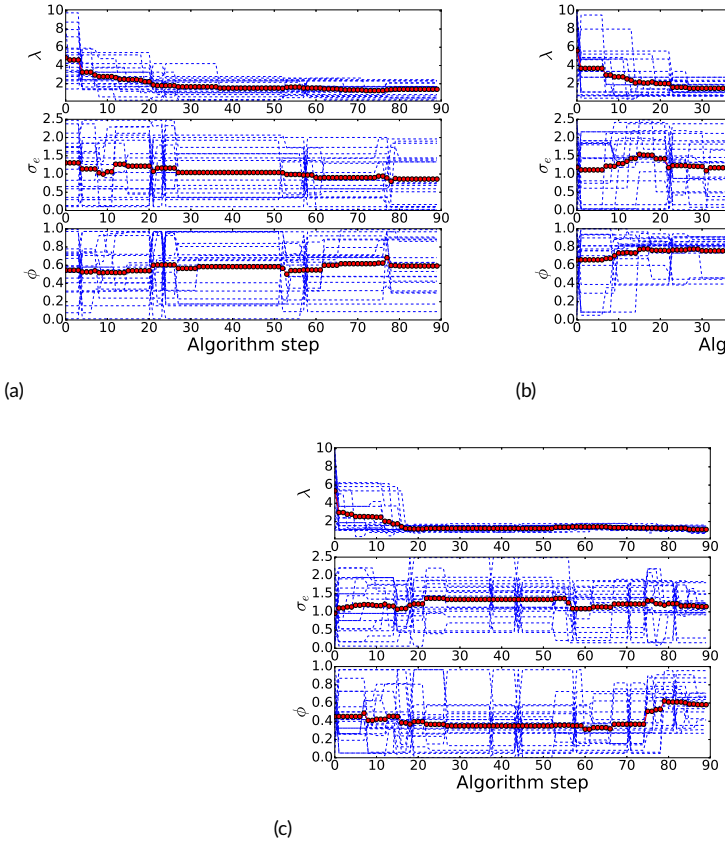



**FIGURE 4** The evolution of the DE algorithm for test L for different temporal distributions of observations: output frequency every (a) 0.2 time units, (b) 0.4 time units, and (c) 0.1 time units. The forecast length is 2.0 time units for all experiments. The red dots show the mean parameter value and the blue dashed lines show the parameter values suggested by DE.

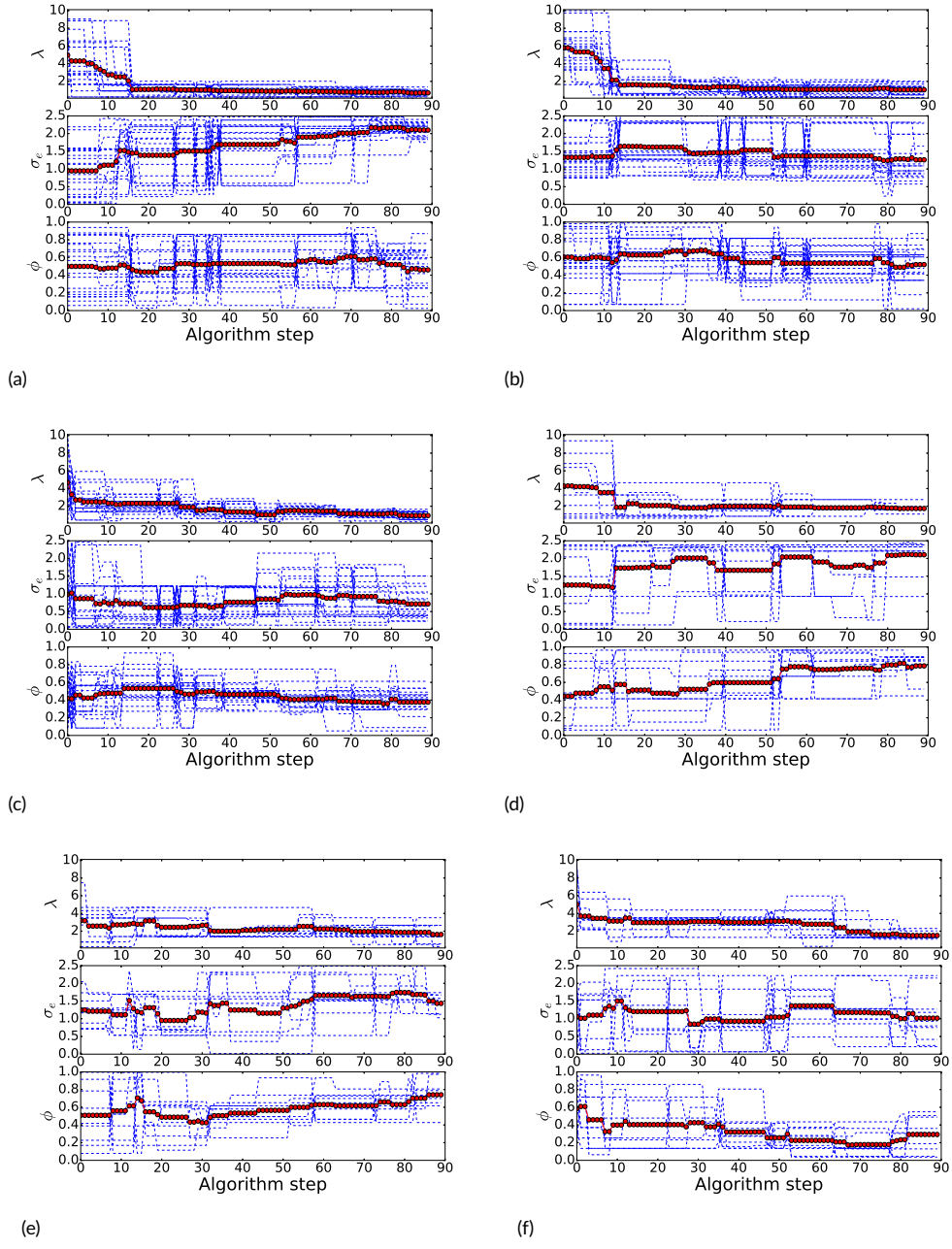
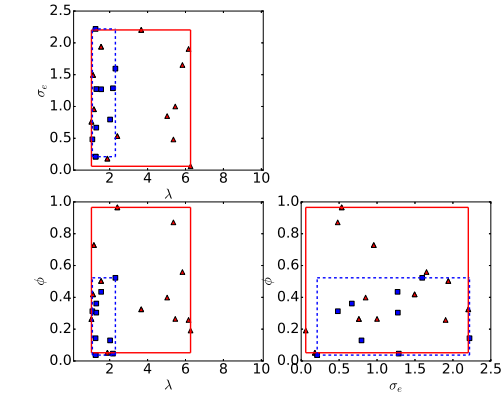
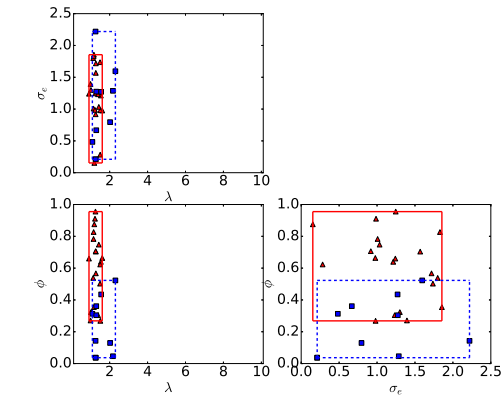



**FIGURE 5** Similar as Figure 4, but with different experimental set-ups of ensemble size ($N$), sequence length ($M$), and population size ($K$). The output frequency is 0.1 time units. (a) test I: $N = 10$, $M = 10$, $K = 20$, (b) test H: $N = 10$, $M = 5$, $K = 20$, (c) test G: $N = 10$, $M = 1$, $K = 20$, (d) test C: $N = 5$, $M = 10$, $K = 10$, (e) test B: $N = 5$, $M = 5$, $K = 10$, and (f) test A: $N = 5$, $M = 1$, $K = 10$.




(a)




(b)

**FIGURE 6** Comparison of test L (red solid line and red triangles) and test A (blue dashed line and blue squares) when (a) the total amount of forecast runs is 4000 and (b) both tests are run 80 algorithm steps. The markers show the location of the parameter vectors in three 2D-projections of the parameter space. The lines mark the approximate distribution of the parameters based on the minimum and maximum of the parameters.

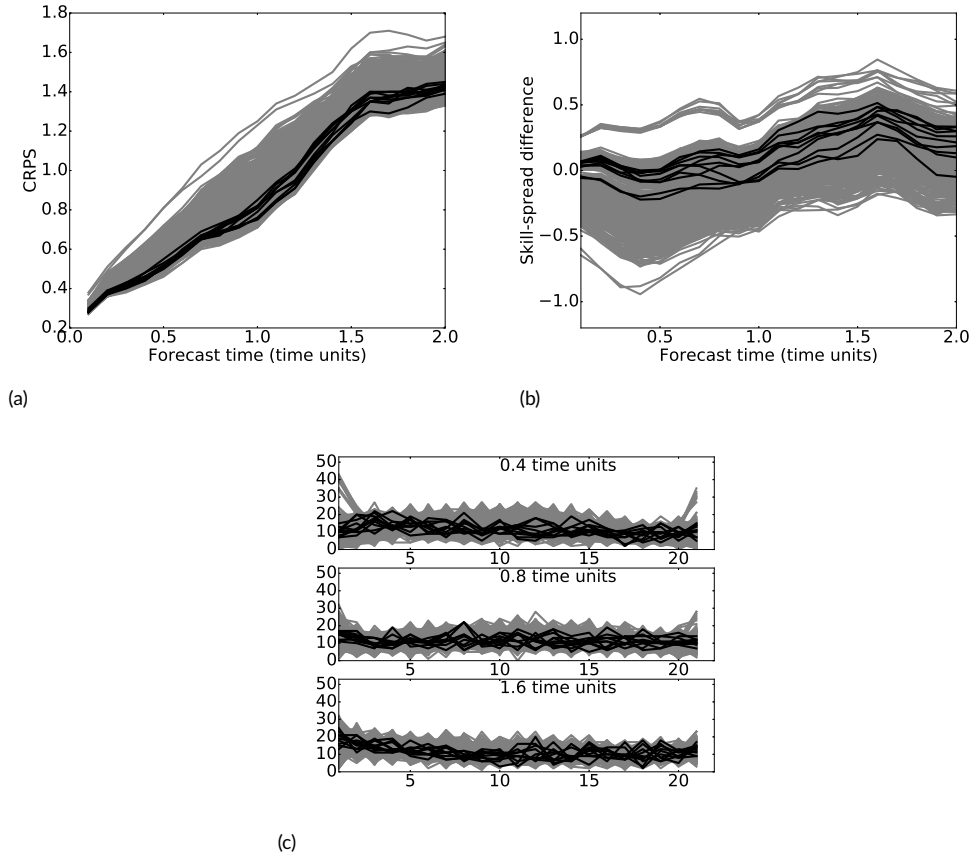



**FIGURE 7** Similar as Figure 3, but for the verification metrics of test A. Marked in black the parameter vectors from the last step of the algorithm (step 89), in gray all parameters suggested by DE (steps 0-88). (a) crps (b) spread-skill (c) rank histogram.