Neural Error Regression Diagnosis (NERD): A Tool for Model Bias Identification and Prognostic Data Assimilation

GAB ABRAMOWITZ

Macquarie University, Sydney, Australia

HOSHIN GUPTA

The University of Arizona, Tuscon, Arizona

ANDY PITMAN

Macquarie University, Sydney, Australia

YINGPING WANG, RAY LEUNING, AND HELEN CLEUGH

CSIRO Atmospheric Research, Melbourne, Australia

KUO-LIN HSU

University of California, Irvine, Irvine, California

(Manuscript received 28 January 2005, in final form 8 July 2005)

ABSTRACT

Data assimilation in the field of predictive land surface modeling is generally limited to using observational data to estimate optimal model states or restrict model parameter ranges. To date, very little work has attempted to systematically define and quantify error resulting from a model's inherent inability to simulate the natural system. This paper introduces a data assimilation technique that moves toward this goal by accounting for those deficiencies in the model itself that lead to systematic errors in model output. This is done using a supervised artificial neural network to "learn" and simulate systematic trends in the model output error. These simulations in turn are used to correct the model's output each time step. The technique is applied in two case studies, using fluxes of latent heat flux at one site and net ecosystem exchange (NEE) of carbon dioxide at another. Root-mean-square error (rmse) in latent heat flux per time step was reduced from 27.5 to 18.6 W m^{-2} (32%) and monthly from 9.91 to 3.08 W m^{-2} (68%). For NEE, rmse per time step was reduced from 3.71 to 2.70 μmol m^{-2} s^{-1} (27%) and annually from 2.24 to 0.11 μmol m^{-2} s^{-1} (95%). In both cases the correction provided significantly greater gains than single criteria parameter estimation on the same flux.

1. Introduction

Mathematical models of natural systems, primarily built to make predictions of systems' behavior, are usually tested using measurements of the variables predicted by the model. Any *systematic* procedure that further uses physical measurements to actually improve model simulation may be termed "data assimilation." Classically in the field of land surface modeling, data assimilation has meant model state estimation (e.g., soil moisture or soil temperature). Measurements of state variables are used, for example, to update a model's predicted values for a period leading up to the present before running the model forward in time in order to make a weather prediction.

Two vital assumptions are made in this type of configuration. The first is that the model's parameters, the time-independent variables that describe the conditions under which the model is operating, are correctly cho-

Corresponding author address: Gab Abramowitz, Department of Physical Geography, Macquarie University, NSW 2109, Australia.
E-mail: gabramow@els.mq.edu.au

sen. The second is that the model itself, the chosen representation and coupling of physical processes, is actually capable of making a prediction with the accuracy and precision that is required. It is now well recognized that when equations representing physical processes within a model are developed at different spatial and temporal scales to those at which the model is applied, many model parameters are not directly measurable. This leaves the modeler with little choice other than to choose "behavioral" parameter values: those whose resulting model output matches observed data well. This procedure, commonly called parameter calibration, is also in some sense a data assimilation technique. It too, however, makes the assumption that the model in question is capable of reproducing the natural system's behavior; all that is required are the "correct" model parameters.

In this paper we use observed data to critique the model itself. We will demonstrate, using a single land surface model, that systematic problems in model simulation resulting from model limitations (rather than parameter mis-prescription) are of far greater significance than the limitations in the accuracy and precision of the observational data used to validate the model. Indeed in the cases presented here, systematic errors resulting from model parameterization problems play a greater role in the model's inability to match observational data than the choice of parameter values. We make it clear that by "model parameterization" we refer concurrently to what others may refer to as "model structure" and "model physics."

We use the Commonwealth Scientific and Industrial Research Organisation (CSIRO) Biosphere Model (CBM) (Wang and Leuning 1998; Leuning et al. 1998), a land surface model developed at CSIRO Atmospheric Research, and examine the existence of systematic trends in the output error. If we can isolate, quantify, and predict such trends, then not only should we be able to correct them, but additionally gain insight into which parts of the model parameterization are ripe for improvement.

We do this by using an artificial neural network (ANN) to simulate model output error as a function of the model's inputs (meteorological forcing) and some outputs on a per-time-step basis (Fig. 1). As an example, the ANN may learn to simulate latent heat flux error (the ANN *output*) as a function of observed downward shortwave radiation, observed humidity, and modeled soil moisture (the ANN *inputs*). This involves a training phase (Fig. 1a) and a testing/simulation phase (Fig. lb). The training phase involves providing the ANN with a time series of input–output pairs from which it establishes the functional dependence (hence

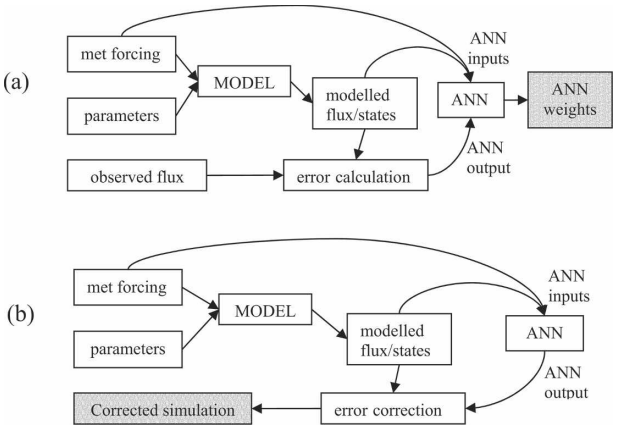

FIG. 1. Configuration for (a) training the ANN and (b) testing the ANN. Shaded boxes represent the goal of each phase. During the training period, the ANN is provided with a set of input–output pairs from which it will establish a functional relationship, recorded in the ANN weights. This relationship is tested by using the ANN to make a correction to model output using "unseen" data.

both ANN inputs and ANN output are directed toward the ANN in Fig. 1a). The end result is a set of ANN parameters or weights. During the testing or simulation phase (Fig. lb), these weights are used by the ANN to make a prediction of the model's error (based on the errors made by the model under similar conditions during the training phase). This prediction is used to make a correction to the model's output.

This approach differs from state-constraint techniques such as Kalman filtering in a number of ways. First, corrections to model states are correcting only for those parameterizations within the model that affect the specific state in question. The process described above corrects for all model parameterizations affecting model output. Second, while implementations of the Kalman filter usually assume zero model bias, this technique specifically attempts to capture the bias. Third, the use of a neural network means the bias relationships learned by the ANN from the testing set can be used to make prognostic corrections. That is, the technique has predictive capability. While the use of ANNs in the natural sciences is not new (see Maier and Dandy 2000) applications to model bias have been very limited (e.g., Martínez and Velázquez 2001; Tetko 2002).

To capture the systematic component of model output error, we need to make a careful choice of ANN. Here we use the regression-based Self-Organizing Linear Output (SOLO) ANN (Hsu et al. 2002), precisely because it simulates only the systematic part of the training data with which it is provided. If it is simply trained with noise, it will make a zero-value simulation.

In this paper, we show the prevalence of systematic error in CBM's output as well as the ability of the regression-based SOLO ANN to capture this error by making a (statistically based) correction to the model at every time step. We use an ensemble of model runs, derived from multiple-criteria parameter estimation, to show that this systematic error is not a result of poor parameter choices. The combination of these two processes defines the Neural Error-modeling Regression-based Diagnosis (NERD) tool.

To begin, we discuss the attribution of error in model output and how we minimize contributions to this error from all sources other than the model itself. We will then detail the datasets, land surface model, and neural network used for the experiment before outlining how they are used together.

2. Defining error

We wish to create conditions under which error in model output results primarily from the model's inherent inability to simulate the natural system. We can represent a deterministic, discrete time step model functionally as

$$Y_t = M(I_t, \phi, \zeta_{t-1}), \tag{1}$$

where Y_t are the model outputs for time step t, I_t the model inputs, ϕ the (time invariant) model parameters, and ζ_{t-1} the model states from the previous time step. We can then represent the simulation *error* as

$$E(\phi, I_t, O_t, \zeta_{t-1}, M) = Y_t - O_t, \tag{2}$$

where O_t are observations of the model outputs made concurrently with the model inputs. This equation suggests five possible sources of model output error: errors in ϕ, the model parameters; observational error in the input and model validation data, I_t and O_t; mis-prescription of initial states, ζ_{t-1}; model parameterization or structure inadequacy, M.

We now outline how we attempt to ensure that the systematic component of model output error, E, is due only to M, and not the other four sources. We additionally try to characterize systematic error in a way that is relatively insensitive to our choice of parameter set. We deal with each of the five possible error sources in Eq. (2) in turn.

a. Choosing model parameter values

We set about identifying parameter sets that are as close as possible to being "correct." While ideally this means we want the values that are those of the natural system, not all parameters are physically observable. This is often because the parameterization of physical processes included in the model has been developed at spatial and temporal scales different to those at which the model is applied. This leaves us with little choice when choosing parameter values other than to use intuition and physical reasoning, and/or to choose the values that make the model perform best. We will refer to parameter sets that make the model best match observations as *behavioral.* By definition the process of choosing values based on a model's posterior adherence to observations (commonly known as calibration) decreases the error in simulations. It does not, however, guarantee that parameter values so obtained are physically meaningful, nor that they would be successful in any other model (Franks et al. 1997). However, since models may include unmeasurable parameters, it seems we can do nothing better than to estimate them in this way. This is essentially what we do here with CBM.

As a starting point, we restrict the ranges of "unobservable" parameters to values within physical limitations and intuition based on observational experience. This restricted range, Φ, termed the *feasible parameter space*, forms the basis for our calibration. To calibrate, we attempt to find a parameter set, $\phi \in \Phi$, so that some objective function $f(\phi)$ is minimized, for example root-mean-square error (rmse) in latent heat. We use fixed initial states and ignore error resulting from observations and model inability, so that we minimize

$$f(\phi) = \left[\frac{\sum_{t=1}^{T} E(\phi, I_t, O_t, \zeta_{t-1}, M)^2}{T} \right]^{1/2} = \left[\frac{\sum_{t=1}^{T} E(\phi)^2}{T} \right]^{1/2} \quad \text{for} \quad \phi \in \Phi$$

in the case of rmse, where T is the total number of time steps. Methods that undertake such a search of the feasible parameter space for an objective function minimum are many, and have varying degrees of success depending on the complexity of the surface formed by the objective function in question (Gan and Biftu 1996; Vrugt et al. 2003b).

These procedures choose parameter values for our multiple-output model that minimize error in only *one* of its outputs. Many single output criteria search algorithms deal with this problem by declaring that the "ideal" parameter set is the one that minimizes, for example, the weighted sum of the objective functions of each model output,

$$f(\phi) = w_1 f_1(\phi) + \cdots + w_n f_n(\phi) \quad \text{for} \quad \phi \in \Phi. \tag{3}$$

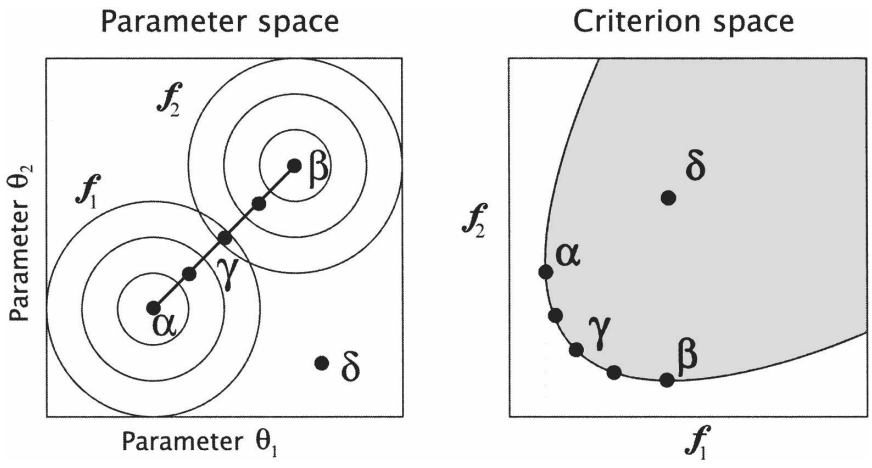

FIG. 2. The parameter and criteria space in a two-criteria, two-dimensional parameter calibration setup. The dark line between the two criteria's minima, α and β, represents the noninferior or pareto set, while the concentric circles in the parameter space represent level curves for the two criteria objective functions. The shaded region represents the projection of the parameter space into the criterion space, and γ is the "compromise point" defined in Eq. (12) (modified after Gupta et al. 2002).

Here n is the number of model outputs for which we have validating observational data, and the w_i are weights. This presupposes, however, that the units of each objective function term are commensurable. For example, that a unit decrease in rmse of latent heat flux (watts per meter squared) is comparable to a unit increase in rmse of soil temperature (kelvin). It should be clear that these two quantities, even if weighted, are not objectively comparable.

Multiple criteria calibration techniques seek to avoid this problem by acknowledging that we ideally wish to find ϕ so that

$$\mathbf{F}(\phi) = [f_1(\phi), \ldots, f_n(\phi)] \quad \text{for} \quad \phi \in \Phi \tag{4}$$

is minimized, in the sense that all the f_i are minimized simultaneously, regardless of their unit of measure. In common practice, of course, no such parameter set exists. Instead we are left with as many "optimal" ϕ as there are criteria f_i. This leads us to define an optimal *region* of the feasible parameter space rather than a single point (Gupta et al. 1999). This region, Φ', has the property that of any two distinct points within it, one always performs better than the other in at least one of the criteria, but never all. That is, for all ϕ_a, $\phi_b \in (\Phi' \subset \Phi)$

$$f_i(\phi_a) < f_i(\phi_b) \quad \text{and} \quad f_j(\phi_a) > f_j(\phi_b) \quad \text{for some} \quad 1 \le i, j \le n. \tag{5}$$

This "noninferior" region of the parameter space is commonly called the *pareto set*. It is illustrated for a two-dimensional parameter space with two model output error criteria in Fig. 2, where the dark line represents the pareto set. The points α and β are the minima of the two criteria objective functions f_1 and f_2, respectively. In the parameter space diagram, the two sets of concentric circles represent level curves for the two objective functions. The shaded region represents the projection of the parameter space onto the criterion space.

In this paper, the multiple-criteria technique we employ to obtain a pareto set for CBM is the Multi-Objective Shuffled Complex Evolution Metropolis algorithm (MOSCEM-UA) (Vrugt et al. 2003a), which essentially combines the Multi-Objective Complex Evolution (MOCOM-UA) (Yapo et al. 1998) and Shuffled Complex Evolution Metropolis (SCEM-UA) (Vrugt et al. 2003b) methods. Details of the nature of the specific search algorithms employed can be found in Vrugt et al. (2003a) and Vrugt et al. (2003b), and a more general discussion of the benefits of the multiple-criteria approach can be found in Gupta et al. (1999).

For a given model, the calibration process provides us with a collection of parameter sets, the pareto set, each member of which contains parameter values that are both realistic and behavioral with respect to at least one of the model outputs for which we have measurements. Our stated goal was to ensure that errors in output from a given model could not be attributed to parameter misprescription. Which point should we then choose from the pareto set to run the model with? Ideally the answer is all of them, since we have no grounds to declare any one point universally "better" than another. That is, if we wish to characterize the nature of

the systematic component of model output error at a particular site in a way that is *independent* of a given parameter set, we must include analysis of model runs using *all* realistic and behavioral parameter values. In practice we are limited by finite computing power, so that "all" needs to become a reasonably small, manageable number while still adequately representing the range of parameter sets within the pareto set. We will discuss how we have selected such subsets in our experimental setup in section 6.

b. Error in state initialization and observations

We now look at the other two sources of error described in Eq. (2), error in observed data and model initialization, which may cause systematic error in model output not originating from model parameterization weaknesses.

State initialization issues are commonly dealt with by what is referred to as model spinup. This involves running the model on the simulation dataset repeatedly until the model states reach equilibrium, at which point we begin recording model output. Using a spinup period usually ensures that model performance is insensitive to initial state values and this was indeed the case for the experiments conducted here (see section 6). There is, however, another way that we might interpret "initial state error."

Equations (1) and (2) represent model behavior for a particular time step during a simulation. If for a moment we ignore error arising from observational (I_t and O_t) and parameter (ϕ) uncertainty, then model output error *not* arising from model inability comes from the states of the previous time step, ζ_{t-1}. Even though we have employed a model spinup period, and hence the value of ζ_{t-1} is insensitive to the *first* time step's state values, ζ_1, there is no reason to believe that ζ_{t-1} will be as measured on site. It is commonly accepted that model states, such as soil moisture and temperature, may "drift" from observed values. The passing of state values from time step to time step, therefore, represents an internal feedback mechanism, since ζ_{t-1} is a function not only of initial state values, but the model inability, parameter value, and input data errors from every time step since the first. This may influence the nature of any systematic error in model output.

This problem will be dealt with in part by ensuring that model states are used as ANN inputs. That is, state values will partially form the set of conditions from which the ANN will be trained to recognize model error. We will discuss this in more detail, together with other possible approaches to dealing with the problem in section 8.

Issues of accuracy in model input (meteorological) and validation (flux) data are not dealt with explicitly in this paper. As we will see after discussing the structure of the SOLO ANN in section 5, this is unlikely to influence the results presented here *unless* they are of a systematic nature. Systematic problems in observational data, where known, need to be dealt with individually and are outside the scope of this paper.

In the following sections we outline the datasets, land surface model, calibration algorithm, and type of neural network used in this paper. Quite some time will be spent discussing the workings of the neural network, as its structure is vital to the success of the NERD process. We then detail how these elements are combined during the training and testing phases of the neural network.

3. Datasets

To illustrate the technique we use two datasets. The first was collected at Cabauw in the Netherlands (51°58′N, 4°56′E) and is described in detail by Beljaars and Bosveld (1997). The site consists mainly of short grass divided by narrow ditches, with no obstacle or perturbation of any importance within a distance of about 200 m from the measurement site. Climate in the area is characterized as moderate maritime with prevailing westerly winds. Variables available, at 20-m height in 30-min intervals for the year 1987, include downward shortwave radiation, downward longwave radiation, air temperature, wind, specific humidity, sensible heat flux, latent heat flux, ground temperature, net radiation, and ground heat flux. These data were used by the Project for the Intercomparison of Land surface Parameterization Schemes (PILPS) (Henderson-Sellers et al. 1995) as both atmospheric forcing and observed flux data, in an evaluation of the performance of a suite of land surface schemes (Chen et al. 1997). As part of this experiment a default parameter set was provided, which we will use here to help quantify the gains made by parameter estimation.

The second was collected at the Harvard Forest site in Massachusetts (42°32′N, 72°10′W). This cool moist temperate deciduous forest site consists of a mixture of hardwoods and conifers, with vegetation height around 25 m near the 30-m measurement tower. The measurement site has an elevation of around 300 m, with mainly sandy loam soils. Hourly averages for the years 1992–99 of the following variables were used: air temperature, downward shortwave radiation, wind speed, relative humidity, rainfall, surface soil temperature, CO_2 flux, latent heat flux, and sensible heat flux. Downward longwave radiation was synthesized using the Swinbank approximation (Swinbank 1963). Unlike Cabauw, for simulations at Harvard Forest, we used time-dependent

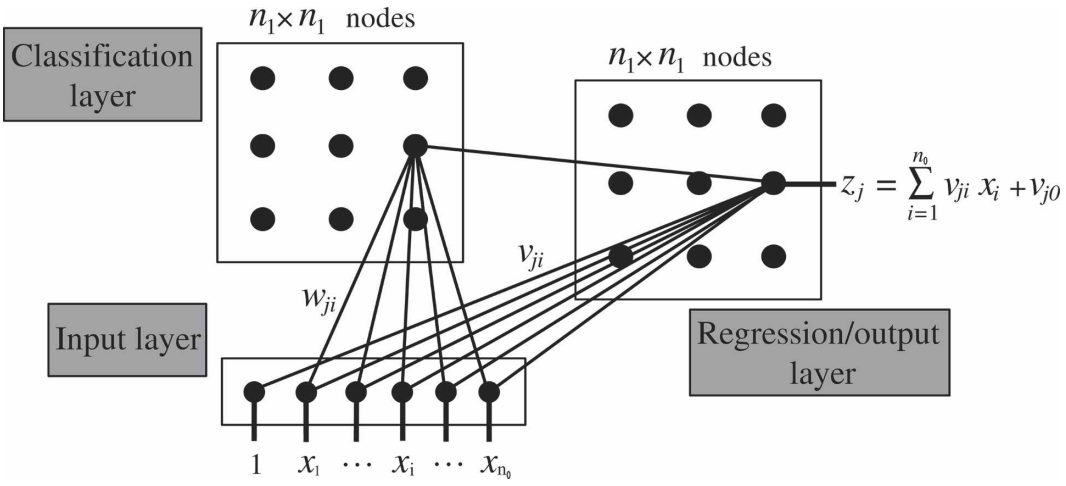

FIG. 3. The three-layer structure of the SOLO neural network. The input layer, classification layer, and weights w_{ji} together form a SOFM, while the output layer performs a node-by-node multiple linear regression with parameters v_{ji}. The input variables $\{x_1, \ldots, x_{n_0}\}$ (e.g., air temperature and wind speed) are normalized, in this case using their maximum possible ranges. After Hsu et al. (2002).

leaf area index, derived from on-site measurements. The carbon flux measurements used here are discussed in Barford et al. (2001). (For a list of publications and details of site instrumentation see http://www-as.harvard.edu/chemistry/hf/.)

4. The CSIRO Biosphere Model

The CBM was developed by CSIRO (Australia). It uses a single-layer, two-leaf canopy model that consists of two parts: 1) a radiation submodel that calculates the photosynthetically active radiation, near-infrared radiation, and thermal radiation absorbed by sunlit and shaded leaves and 2) a coupled model of stomatal conductance, photosynthesis, and partitioning of absorbed net radiation into sensible and latent heat (Leuning 1995; Leuning et al. 1995, 1998; Wang and Leuning 1998). The soil component uses a six-layer structure to compute heat conduction and Richards' equation to calculate moisture transport, and includes soil freeze and thaw cycles. The snow model computes the temperature, snow density and thickness of three snowpack layers.

CBM took part in the PILPS C1 experiment (www.pilpsc1.cnrs-gif.fr), which compared land surface model performance using data collected at the Loobos, Netherlands, pine forest site. CBM's demonstrated competence in this experiment (results at www.pilpsc1.cnrs-gif.fr) suggests that the results presented here should be applicable to other models. When we speak of "the model" in the experiments considered here, we mean CBM.

5. The SOLO neural network

An ANN may be thought of as a mathematical function that, through an iterative process, adjusts its own constants or parameters to fit a given set of data. Most commonly, ANN operation is split into two phases, one to train the ANN and one to test or use it for prediction. This process, providing the ANN with a fixed set of input/output pairs from which it establishes the desired functional relationships, is known as "supervised training." Figure 1a represents the supervised training phase, and Fig. 1b represents the testing phase.

For our purpose of modeling systematic trends in model output error we chose the SOLO neural network (Hsu et al. 2002). Our primary reason for doing so is that the structure of the SOLO map ensures that only systematic trends in training data are captured; there is little risk of modeling noise in data, often an issue with overtraining in feed-forward ANNs. We will discuss this and other reasons for our choice in more detail after we have outlined the SOLO map's structure and operation. The description below follows from Hsu et al. (2002).

The SOLO map consists of three layers, shown in Fig. 3: an input layer, an input classification layer, and a regression or output layer. The input layer, given n_0 input variables (such as air temperature or wind speed), consists of $n_0 + 1$ nodes. The unit input that forms the extra node is used only in the regression stage of operation. Both the classification layer and output layer are square matrices of $n_1 \times n_1$ nodes. Joining the ith *nonunit* input node to the jth classification layer node is the weight w_{ji}, for all $i = 1, \ldots, n_0$ and

$j = 1, \ldots, n_1 \times n_1$. These weights, w_{ji}, together with the input and classification layers form a Self Organizing Feature Map (SOFM) (Kohonen 1989), which operates in the following way.

We wish to classify a set of input data into groups, where each group is represented by a classification layer node. These data are a collection of n_0-dimensional vectors, with each member of the collection representing an observation/model time step (though not necessarily chronologically ordered—see section 6). To do this, we first normalize each input variable (here we use the range of possible values for each variable, as described in section 6). We then define the distance, d_j, between a given input vector $\mathbf{x} = (x_1, \ldots, x_{n_0})$ and the jth classification layer node to be

$$d_j = \left[\sum_{i=1}^{n_0} (x_i - w_{ji})^2 \right]^{1/2}. \tag{6}$$

Each input vector $\mathbf{x}$ belongs to the group or node to which its distance is shortest. We refer to the node, c, for which $d_c = \min(d_j)$ for all $j = 1, \ldots, n_1 \times n_1$ as the winner node for $\mathbf{x}$. Training the SOFM is then a matter of choosing the w_{ji} to spread the input data amongst the classification layer nodes.

To begin, the weights w_{ji} are randomly initialized. Then, for a given input vector $\mathbf{x}$, we adjust all the classification layer nodes within a (square) neighborhood, Λ_c, of the winner node for $\mathbf{x}$:

$$w_{ji} = \begin{cases} w_{ji} + \eta[x_j - w_{ji}] & \text{if} \quad j \in \Lambda_c \\ w_{ji} & \text{otherwise.} \end{cases}$$

In this equation, η is the *learning rate* or size of adjustment. If $\eta = 1$, all the nodes in Λ_c will have a zero distance from $\mathbf{x}$; if $\eta = 0$ the distances remain unchanged. We first adjust the w_{ji} for all vectors in the training set using a large value of $0 < \eta < 1$ and large neighborhood size Λ and then repeat the process many times with η and Λ reduced progressively. As training progresses, nodes in the classification layer will become associated with data-rich regions of the input space (Hsu et al. 2002). This ensures that if the data occupy only a small proportion of the range by which they've been normalized, they are still well spread amongst the classification layer nodes. Weight adjustment ceases when the distribution of input vectors amongst the classification layer nodes stabilizes.

At this point, with SOFM training complete and all input vectors associated with nodes of the classification layer, a direct link is made between the jth node of the classification layer and jth node of the output/regression layer (see Fig. 3). By this, we mean each node in the regression layer is associated with the subset of the input data belonging to the jth classification layer node. A linear regression is then performed between this subset of the input data and its associated output data (remembering for this training period we provided the SOLO map with a set of input–output pairs). The weights between each input layer node (*including* the unit input node) and the jth regression layer node, $\{v_{ji} | i = 0, \ldots n_0\}$, are the parameters of this regression (see Fig. 3).

Once such regression parameters have been established, so that we have now trained the weights v_{ji} as well as w_{ji}, the SOLO map training is complete. Given an input vector $\mathbf{x} = (x_1, \ldots, x_{n_0})$ with winner node c in the classification layer, the output from a trained SOLO network will be

$$z = \begin{cases} \sum_{i=1}^{n_0} v_{ji} x_i + v_{j0} & \text{for} \quad j = c \\ \varnothing & \text{otherwise.} \end{cases} \tag{7}$$

Here, Ø implies that no calculation is performed, so that only one of the regression matrix nodes, the winner node for $\mathbf{x}$, gives an output. We can also see that the input layer unit node provides us with the constant term in the regression.

Finding the regression parameters, v_{ji}, however, is not trivial. For a given node in the regression layer, let the number of input–output pairs supplied by SOFM training be p. We then need to solve a set of linear equations for $\theta = (v_{j0}, v_{j1}, \ldots, v_{jn_0})$,

$$\mathbf{Z} = \mathbf{X}\theta + \boldsymbol{\epsilon}, \tag{8}$$

where $\boldsymbol{\epsilon}$ is a $p \times 1$ vector of estimation errors with zero mean, $\mathbf{Z}$ is the $p \times 1$ vector of p output data for training, and $\mathbf{X}$ is a $p \times (n_0 + 1)$ matrix containing p rows of n_0-variable training data. This in turn requires us to solve the normal equations (see derivation in appendix A),

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\theta = \mathbf{X}^{\mathrm{T}}\mathbf{Z}. \tag{9}$$

Ideally, we would then simply solve $\theta = (\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathrm{T}}\mathbf{Z}$ to find the regression parameters. However, it is not always the case that $\mathbf{X}^{\mathrm{T}}\mathbf{X}$ is invertible; there may be significant correlation between the input variables x_i. This issue is dealt with by recasting Eq. (8) in terms of the principal components of $\mathbf{X}$. Let $\mathbf{Y}$ be the $p \times (k + 1)$ matrix of the $k \leq n_0$ principal components of $\mathbf{X}$ defined by

$$\mathbf{Y} = \mathbf{X}\mathbf{C}, \tag{10}$$

where $\mathbf{C}$ is the $(n_0 + 1) \times k$ transformation matrix with eigenvectors derived from the covariance matrix of $\mathbf{X}$ satisfying $\mathbf{C}^T\mathbf{C} = \mathbf{C}\mathbf{C}^T = \mathbf{I}$ (see appendix B). We then have

$$\mathbf{Z} = \mathbf{X}\theta + \epsilon = \mathbf{Y}\mathbf{C}^T\theta + \epsilon = \mathbf{Y}\psi + \epsilon \tag{11}$$

and are now guaranteed that a solution to the principal component analog to Eq. (9), $\psi = (\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{Z}$, exists, since the variables in $\mathbf{Y}$ are orthogonal. By choosing only a subset of the principal components of $\mathbf{X}$, those which explain the most variance, we simultaneously reduce instability in regression parameter estimation and speed network training by reducing the dimension of the matrices involved. For the work considered here, we ensure that the above transformation preserves most of the variance in $\mathbf{X}$ by keeping the ratio $\Sigma_{i=1}^{k}\lambda_i / \Sigma_{i=1}^{n_0}\lambda_i > 95\%$ at each node.

The regression structure of the SOLO map ensures no correction will be made if there is no systematic trend in the model's output error. In this case, the gradient and intercept regression parameters for each node will be zero. This makes it ideal for use as a bias correction model. For the same reason, noise in observational data should not affect the input–output relationships established, provided we have enough data for training. Additionally, the regression structure eliminates the many potential problems encountered with other ANNs that use error space search algorithms to find optimal network parameters (e.g., problems with local minima and overtraining). It is also computationally more efficient than either multilayer feedforward or recurrent neural networks (Hsu et al. 2002). In the sections to follow, when we speak of "the ANN" we mean the SOLO map, and by SOFM "resolution" we mean the number of nodes, n_1^2, in the SOFM.

6. Experimental setup

To demonstrate the NERD process, we make a correction to CBM's simulation output at the two observational sites described in section 3. In both cases we correct only a single model output flux, although it should be clear that extending the SOLO map architecture to deal with several outputs is relatively simple. The processes of training and testing the ANN, described below, are shown schematically in Figs. 1a and 1b, respectively.

a. Case 1: Latent heat correction at Cabauw

For the first case, we perform a correction to the predictions of latent heat flux by CBM using the Cabauw data. The first step in this process is to select the parameter set or sets required to run CBM. In section 2a, we discussed the benefits of using a finite collection of points from the pareto set for this purpose, to characterize model systematic error in a parameter-independent way. In this case, we perform a multiple-criteria calibration using all 17 520 time steps of the Cabauw data, using rmse in latent and sensible heat flux as the two objective function criteria. From the resulting pareto set, we select five representative parameter sets: one at each of the calibration criterion's minima, and three others evenly spaced between these (Fig. 2). By "evenly spaced" we mean rmse distance defined in the following way: if α and β are the two points in the parameter space where f_1 and f_2, the two criteria functions, have minima, then the point, γ, in the pareto set with rmse closest to

$$[f_1(\gamma), f_2(\gamma)] = \left[\frac{f_1(\alpha) + f_1(\beta)}{2}, \frac{f_2(\beta) + f_2(\alpha)}{2}\right] \tag{12}$$

is the midpoint between α and β (see Fig. 2).

In addition to these five parameter sets, we use two default parameter sets for reference purposes. One of these was provided by Beljaars and Bosveld (1997) for the PILPS phase 2a experiment (Chen et al. 1997); the other was our choice of default parameters for Cabauw, using generic vegetation and soil type.

To demonstrate the processes involved in using these seven parameter sets we first consider the simplest configuration. CBM is run with a single default parameter set for the entire year of Cabauw forcing data. We then have 17 520 time steps of model output, observations of latent heat flux (provided with the Cabauw meteorological forcing), and meteorological forcing. Alternate time steps are allocated to training and testing sets, giving two 8760 time step sets.

During both the training and testing phases for this experiment, input variables to the ANN are observed downward shortwave radiation (S), air temperature (T), specific humidity (Q), and wind speed (W) together with modeled latent heat flux (L) and top layer soil moisture content (M). For training, output is simply latent heat flux error (L), as defined in Eq. (2), while for testing, output is the ANN simulation of this error. These input variables were chosen based on physical reasoning as well as an analysis of the correlation between model inputs and latent heat residual. Table 2 suggests there are significant relationships between the six inputs chosen and latent heat error. The training and testing sets therefore, are a series of 8760 input–output pairs:

$$[\mathrm{IN}|\mathrm{OUT}] = \left[\begin{array}{cccccc|c} S_1^{\mathrm{obs}} & T_1^{\mathrm{obs}} & Q_1^{\mathrm{obs}} & W_1^{\mathrm{obs}} & L_1^{\mathrm{mod}} & M_1^{\mathrm{mod}} & L_1^{\mathrm{err}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_k^{\mathrm{obs}} & T_k^{\mathrm{obs}} & Q_k^{\mathrm{obs}} & W_k^{\mathrm{obs}} & L_k^{\mathrm{mod}} & M_k^{\mathrm{mod}} & L_k^{\mathrm{err}} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ S_{8760}^{\mathrm{obs}} & T_{8760}^{\mathrm{obs}} & Q_{8760}^{\mathrm{obs}} & W_{8760}^{\mathrm{obs}} & L_{8760}^{\mathrm{mod}} & M_{8760}^{\mathrm{mod}} & L_{8760}^{\mathrm{err}} \end{array}\right]. \quad (13)$$

Using the nomenclature of section 5, we have input vectors $\mathbf{x}_k = (x_{1k}, \ldots, x_{n_0}k) = (S_k^{\mathrm{obs}}, T_k^{\mathrm{obs}}, Q_k^{\mathrm{obs}}, W_k^{\mathrm{obs}}, L_k^{\mathrm{mod}}, M_k^{\mathrm{mod}})$ and output vectors $z_k = L_k^{\mathrm{err}}$. For this experiment, the order of elements in the set of training pairs $\{(\mathbf{x}_k, z_k), k = 1, 8760\}$ is not important since all variables are for an individual model time step; each member is treated independently. This means we do not necessarily need continuous data, although in this case the data are continuous. To use the SOFM in the SOLO ANN, the inputs need to be normalized, and for this we use the ranges prescribed by the ALMA convention (http://www.lmd.jussieu.fr/~polcher/ALMA/convention_3.html). These ranges are the theoretical global limits for each variable.

This configuration is used for CBM runs with each of the five parameter sets chosen from the pareto set and the two default parameter sets, leaving us with seven trained ANNs and seven respective testing sets.

The second configuration examines the systematic trends in CBM's error in a *parameter independent* way by using the same ANN inputs and output as mentioned above but utilizing all five pareto point runs. That is, $5 \times 17\,520/2 = 43\,800$ input–output pairs are provided for ANN training, and 43 800 provided for testing. In terms of the matrix in Eq. (13) above, this simply involves increasing the length of each column by a factor of 5.

For each of the seven model runs used in these eight experiments, a 5-yr spinup period was used to remove sensitivity to initial state values. To be certain of its success, five initial state value sets were used for all model runs with each of the seven parameter sets. These state sets had soil moisture values ranging from wilting point to above soil saturation as well as soil temperature values ranging from 0° to 20°C. In every case, after the 5-yr spinup period, variation in rmse in latent heat amongst runs with a fixed parameter set but different initial state values was three orders of magnitude less than the variation between runs with different parameter sets.

b. Case 2: Carbon correction at Harvard Forest in dynamic conditions

We demonstrate the broad applicability of the NERD process by making a correction to carbon fluxes at a different site. The experimental setup is similar to the first case, but is additionally designed to explore the validity, in a dynamic environment, of the statistical correction. We stress that we are primarily using the NERD methodology as a tool to identify systematic model weakness, but the question of how fundamental this weakness is, relative to changes in climate system behavior, remains open. To investigate this question, the 8 yr of Harvard Forest data described in section 3 are used to make a correction to net ecosystem exchange (NEE) predictions. The first 4 yr of data are used both to select parameter values *and* train the ANN. The second 4 yr are used to test the relationships so established.

In selecting parameter sets in this case, we perform a three criteria calibration, using rmse latent heat, sensible heat, and NEE on the first 4 yr of Harvard Forest data. From the pareto set obtained we make use of four parameter sets: the three which minimize each of the criteria as well as one "compromise" point, δ, defined as follows. If α, β, and γ are the three points in the parameter space which minimize the three criteria f_1, f_2, and f_3 respectively, then define δ as the point with rmse, $[f_1(\delta), f_2(\delta), f_3(\delta)]$, closest to

$$\left\{\frac{f_1(\alpha) + \min[f_1(\beta), f_1(\gamma)]}{2}, \frac{f_2(\beta) + \min[f_2(\alpha), f_2(\gamma)]}{2}, \frac{f_3(\gamma) + \min[f_3(\alpha), f_3(\beta)]}{2}\right\}. \quad (14)$$

We again include a default model run using generic vegetation and soil type parameter values.

Inputs to the ANN in this case are (observed) downward shortwave radiation, surface air temperature, and leaf area index together with (modeled) latent heat flux, top layer soil temperature, and net ecosystem ex-

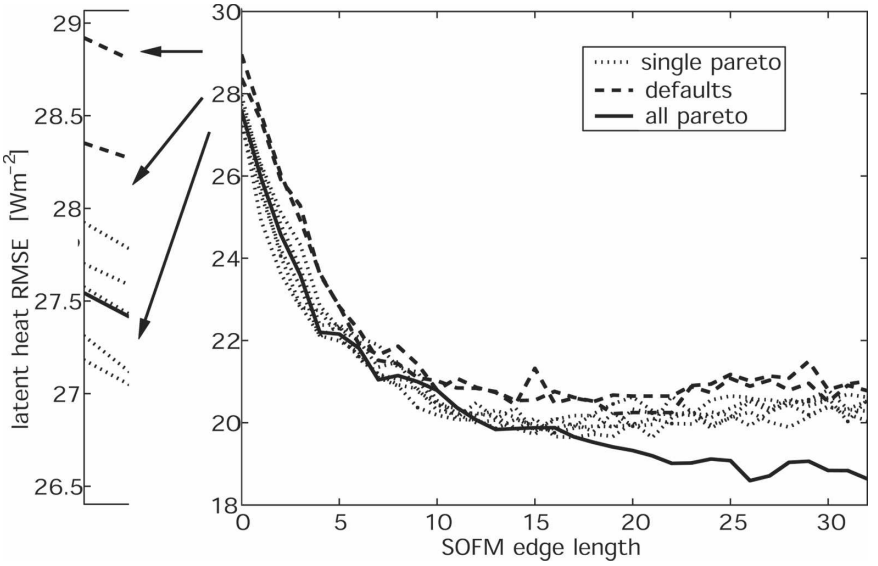

FIG. 4. Rmse of corrected model simulation vs the size of the self-organizing feature map used to make the correction for latent heat correction at Cabauw. Dotted lines represent ANNs trained and tested on model output generated using a single pareto point, dashed lines use a single default parameter set, and the solid line uses an ensemble of model runs using all five pareto points. Zero edge length (the *y* axis) is the uncorrected model simulation.

change. Output is error in NEE. We again make a correction on a per-time-step basis.

7. Results

We first address the *existence* of systematic model error. (The dotted lines in Figs. 7 and 8 show the average daily and monthly values of the two fluxes predicted by CBM at the two sites; the solid lines represent observations.) March, July, and November were chosen as evenly separated months that include the middle of the Northern Hemisphere summer. Results in both figures are an average of the ensemble of all pareto point runs (and testing years in the Harvard Forest case). We can see that CBM consistently underrepresents latent heat flux at Cabauw. Harvard Forest NEE was also underpredicted during the winter months, with CBM predicting virtually no net emission of CO_2. Had we any doubt, it should now be clear that the model has a detectable systematic bias.

Each of the eight experiments outlined in case 1 of section 6 using Cabauw data are represented by a line in Fig. 4. This plot shows the per-time-step rmse of model simulations corrected by the ANN for a range of self-organizing feature map (classification layer) resolutions. The *x* axis represents the SOFM edge length, n_1, as shown in Fig. 3, and the rmse value at zero resolution is simply the rmse of the uncorrected model run. No attempt has been made here to distinguish between runs generated by the two default parameter sets (dashed) or runs generated by the five individual pareto set runs (dotted), rather we consider them as two behavioral groups. The solid line represents the performance of the ANN trained on the ensemble pareto set runs.

From this figure we see that a correction made by the ANN using a SOFM with edge length 32 (implying a $32 \times 32 = 1024$ node SOFM) that has been trained and tested on all five pareto set runs can decrease the simulation rmse for latent heat from 27.5 to 18.6 W m^{-2} (32%). If we look at the *y* axis or zero-resolution line (enlargement in Fig. 4) we see how effective parameter calibration is in this case. The difference between the worst-performing default parameter set and the best-performing noninferior parameter set is around 1.74 W m^{-2}, or about 6%. From the best-performing default to the worst-performing noninferior point is 0.44 W m^{-2}, or around 1.5%. Even a single unit SOFM ("1" on the *x* axis in Fig. 4) gives a 7% improvement in the rmse. That is, making a linear correction to the latent heat flux of CBM at Cabauw based on the six ANN inputs gives a correction of a similar size to parameter calibration.

The analogous plot for CO_2 flux correction at Harvard Forest shows a similar trend (Fig. 5). It represents the per-time-step rmse performance of NEE prediction by CBM for the 4-yr testing period (1996–99) for a range of ANN complexity. The best-performing correc-

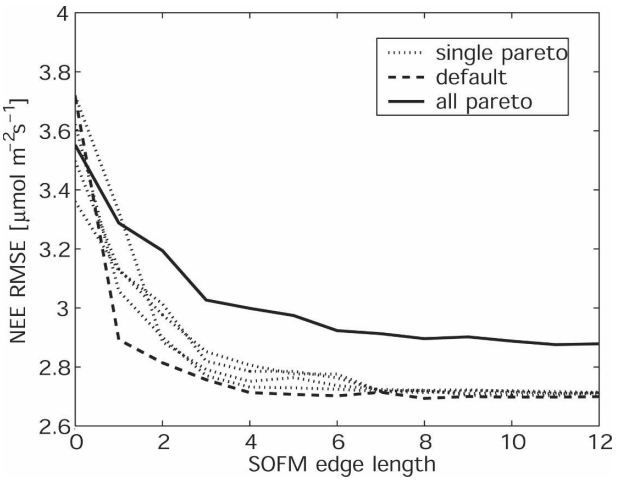

FIG. 5. Rmse of corrected model simulation vs the size of the self-organizing feature map used to make the correction for NEE correction at Harvard Forest. Dotted lines represent ANNs trained and tested on model output generated using a single pareto point, dashed lines use a single default parameter set, and the solid line uses an ensemble of model runs using all five pareto points. Zero edge length (the *y* axis) is the uncorrected model simulation.

tion came from the ANN trained and tested on the default parameter set, a 12^2 node SOFM reducing the per-time-step rmse from 3.71 to 2.70 μmol m^{-2} s^{-1} (27%). The all-pareto point experiment resulted in a drop from 3.55 to 2.88 μmol m^{-2} s^{-1} (18%).

Parameter calibration in this case reduced NEE per-time-step rmse from 3.71 to 3.36 μmol m^{-2} s^{-1} (9%) for the NEE minimum in the pareto set. The sensible heat minimum in the pareto set resulted in a marginally higher per-time-step rmse than that resulting from the default parameter set.

It appears that the nature of model systematic error in latent heat flux at Cabauw is not wholly parameter dependent. In Fig. 4, the all-pareto point experiment (which used *five* model runs) performed better than any of the single pareto point experiments, suggesting there is information about model weakness using one parameter set when running the model with another. The nature of the model's systematic error was therefore best generalized by the ANN trained using multiple parameter sets. This was not the case for the carbon correction at Harvard Forest, however. The reason for this is most likely that calibration process that generated the Harvard Forest pareto set used *three* criteria, instead of the two used at Cabauw. The result was a larger pareto set and consequently a larger range of model behavior for the ANN to capture.

For further analysis, unless otherwise stated, we use results from all-pareto trained ANNs. In the Cabauw case, we use a 32^2 node SOFM, trained and tested on the five pareto point model runs and for Harvard Forest, a 12^2 node SOFM trained and tested on the four pareto point model runs.

We now consider rmse on longer time scales. Figures 6a and 6b show rmse for a range of averaging window sizes. Half-day averages to 20-day averages are plotted for Cabauw and up to 40-day averages for Harvard Forest. Results are shown for all-pareto-point model (solid) and corrected model (dashed) runs as well as default model (dash–dot) and corrected model (dotted) runs. This gives us an indication of the relative effectiveness of parameter estimation and the NERD correction. If we dispense with parameter estimation altogether and simply implement NERD using default parameter sets only, results in the Cabauw case are only marginally worse and the Harvard Forest case, significantly better. A summary of the improvements is shown in Table 1, which suggests reductions in rmse are achieved both by increasing averaging time and applying the NERD correction. Note that the relative size of the NERD correction increases with increasing averaging time. While daily carbon flux rmse is reduced by 53% in the default simulations, dropping from 2.66 to 1.25 μmol m^{-2} s^{-1}, the annual reduction is 95%, a drop from 2.24 to 0.11 μmol m^{-2} s^{-1}.

The remainder of Fig. 6 represents the results of scatterplots of modeled versus observed values for both fluxes. Ideally, we want a unit gradient and zero offset for the least squares linear regression lines for such plots, regardless of whether we consider a scatter based on per-time-step, daily, weekly, or monthly averages. The gradient of such regression lines (Figs. 6c,d) as well as the square of the correlation coefficient, r^2 (Figs. 6e,f), are shown for a range of averaging window sizes at both sites. The solid line represents the gradient of model versus observed, and the dashed line represents *corrected* model versus observed. The shaded gray regions surrounding each line represent the 95% confidence intervals on the gradient estimates, which naturally broaden as we consider longer-term averages and the sample size shrinks.

The most striking result here is the correction of simulated CO_2 at Harvard Forest. While the gradient of model simulation versus observation converges to a value around 0.7 (with increasing size of averaging window), the corrected simulation is unbiased at 10-day or greater averages (where the 95% confidence interval includes the unit gradient). Correlation between observed and modeled CO_2 at Harvard Forest was also significantly improved by the correction. The Cabauw case was not so dramatic. While observed–modeled correlation was clearly bettered at all time scales by the

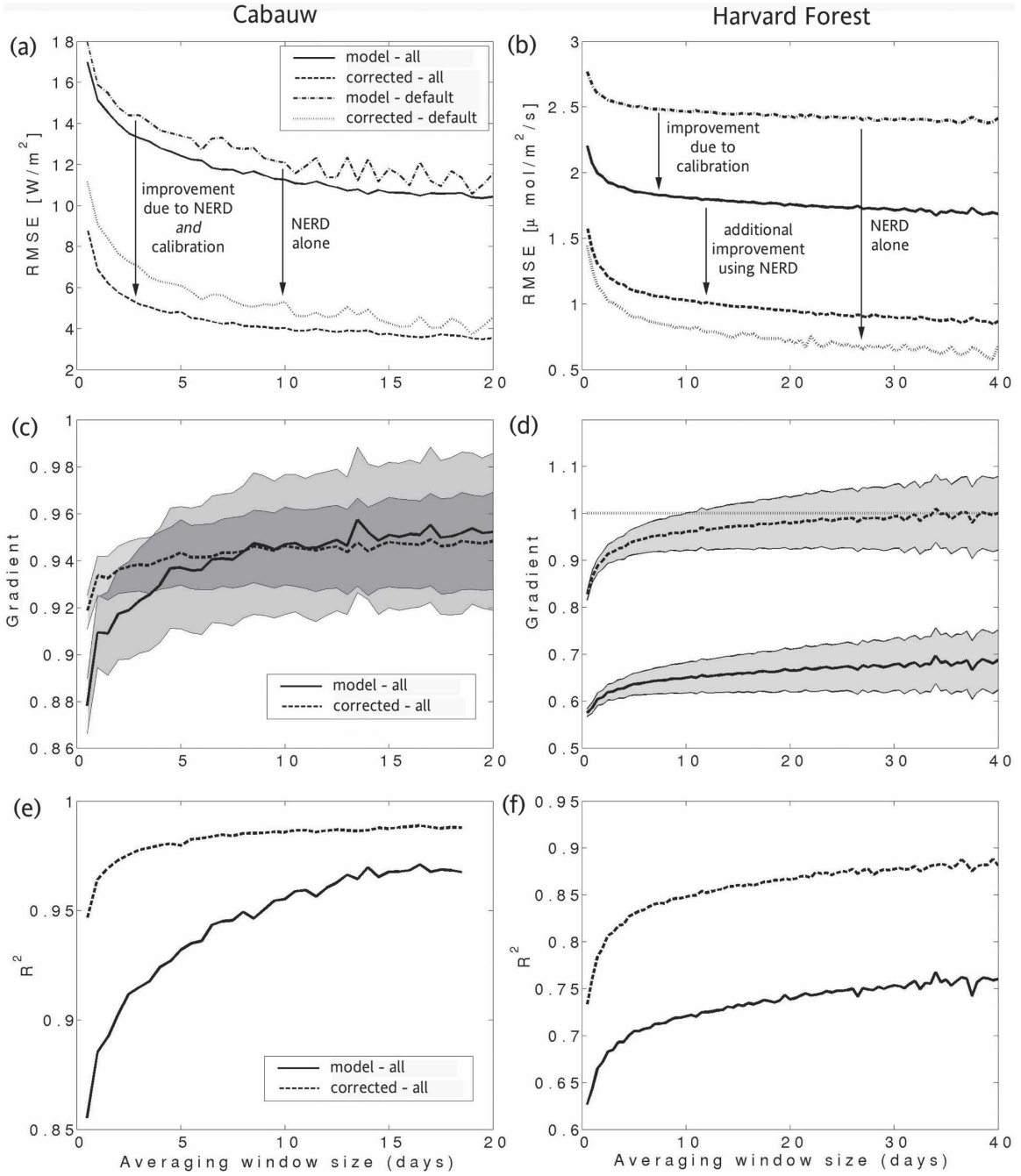

FIG. 6. (a), (b) Root-mean-square error; (c), (d) gradient of least squares regression of model and corrected model vs observed; and (e), (f) Pearson correlation coefficient, r^2, for latent heat at Cabauw and net ecosystem carbon exchange at Harvard Forest. The x axis represents the window averaging size; the shaded area around the lines represents the 95% confidence interval on the gradient estimates. Results are shown for ensemble model runs using pareto parameter sets; (a) and (b) additionally include default parameter set results.

correction, the corrected model versus observed regression gradient was only better for 8-day averages or less.

We now briefly look at the impact of the corrections on the diurnal and annual cycles of the two fluxes. Figure 7 shows the ensemble average day for three separate months during the simulation at both sites. Figure 8 shows average monthly flux values at both sites. The model underestimated the latent heat flux at Cabauw during each month and significantly so in March, while

TABLE 1. Daily, weekly, monthly, and annual decrease in rmse for latent heat flux at Cabauw and NEE carbon flux at Harvard Forest due to the NERD correction. Results are shown for the correction utilizing all pareto-point runs (all) and default model parameter set runs (def). Annual values for Cabauw are omitted due the brevity of the dataset.

Cabauw latent heat rmse (W m^{-2})				
	Model (all)	Model + NERD (all)	Model (def)	Model + NERD (def)
Daily	15.13	6.87	15.89	9.05
Weekly	11.75	4.23	13.30	5.63
Monthly	9.91	3.08	11.07	3.96
Harvard Forest NEE rmse (μmol m^{-2} s^{-1})				
	Model (all)	Model + NERD (all)	Model (def)	Model + NERD (def)
Daily	2.07	1.41	2.66	1.25
Weekly	1.83	1.06	2.48	0.86
Monthly	1.71	0.90	2.40	0.64
Annually	1.35	0.52	2.24	0.11

the application of the NERD process removed almost all of this bias (Fig. 7). A similar result was obtained for NEE at Harvard Forest, with NERD able to remove both positive and negative biases in model predictions. Systematic errors in modeled monthly mean latent heat fluxes were largely eliminated by NERD at Cabauw (Fig. 8), but the correction led to a systematic positive bias in NEE at Harvard Forest, in contrast to the negative biases in winter and autumn from the model alone.

8. Discussion

The results in section 7 demonstrated that the NERD process led to significant improvements in model performance at all time scales for most of the measures we considered. The ANN successfully identified and corrected systematic bias in model output for calibrated and default parameter sets.

Reasons for choosing one parameter set over another when making a NERD correction are not yet clear. In the Cabauw case, gains made by parameter calibration were preserved by the NERD correction; the separation of model performance using default parameter values versus pareto parameter values remained intact regardless of SOFM resolution in the correcting ANN (Fig. 4). At Harvard Forest, however, the default parameter set, which had considerably higher rmse for uncorrected model runs, consistently outperformed any of the pareto point model runs once the ANN correction was applied (Fig. 5). The use of multiple pareto parameter sets effectively gave several times as much training data with which to generalize the model's systematic error, which at Cabauw resulted in the superior performance of the all-pareto correction. This again was not true at Harvard Forest. A possible resolution of this issue could be the use of an all-pareto ANN that additionally includes selected model parameters as inputs.

We now consider possible improvements to the technique. Table 2 shows the Pearson correlation coefficient (r) between ANN inputs and model error before and after ANN correction. It also shows the "P value," a measure designed to gauge the significance of the correlation. It represents the probability of getting the given correlation by random chance, with the hypothesis of no correlation. Traditionally 5% (0.05) or less is deemed significant, implying the hypothesis is false. A zero P value here implies a value less than 10^{-100}. Table 2 suggests that although the ANN has significantly reduced systematic error, it has by no means done a comprehensive job. At Cabauw, the ANN largely removed the significant correlations between the pairwise error in latent heat fluxes and air temperature, humidity, and modeled latent heat flux, but the relatively minor decrease in the pairwise correlation of the other three inputs and latent heat error show that the ANN did not adequately capture this dependence. This problem is even clearer at Harvard Forest, where even after correction all ANN input–output P values were less than 10^{-10}. This suggests that the improvements made by NERD correction could be better.

A possible reason for this problem is what we might call *dimension resolution* in the SOFM. If we set all ANN input variables to be constant except one, say temperature, what does the temperature-output relationship look like? That is, since we are considering a piecewise linear approximation of the input–output relationship, how many linear "pieces" are used to resolve the temperature dimension? If we consider a simple 2×2 SOFM with only two input variables, x_1 and x_2, for any fixed value of x_1, we cannot expect an x_2-output graph to be any more complex than a two-piece linear approximation. This leads us to define a *dimension resolution number, k*, such that if our ANN has n_0 input variables and n_1^2 SOFM nodes,

$$k = (n_1^2)^{1/n_0}.$$

In the Cabauw case, $k = (1024)^{1/6} \approx 3.2$, and for Harvard Forest $k = (144)^{1/6} \approx 2.3$. To improve this situation we could use the principal components of the ANN inputs instead of the inputs themselves, but that is not explored in this paper.

One issue mentioned in section 2b that could complicate results was model state drift. Ideally we would

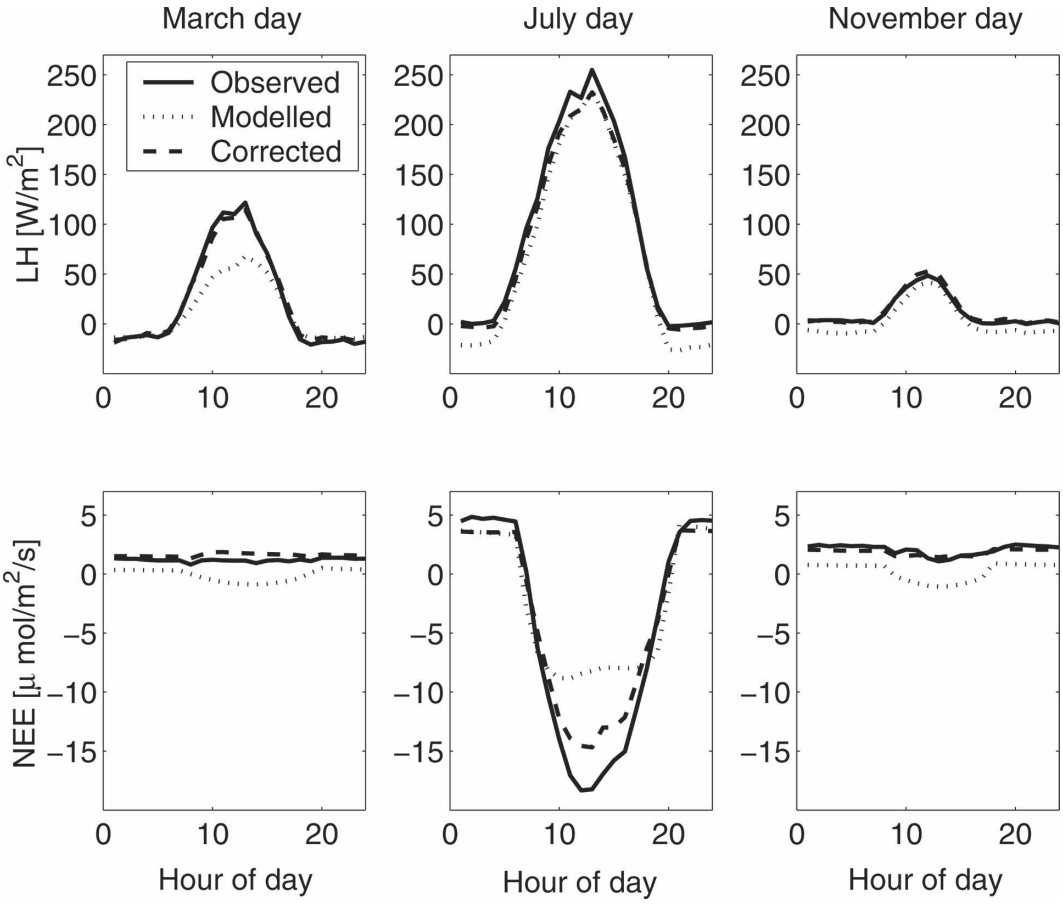

FIG. 7. Average day fluxes for March, July, and November for (top) Cabauw latent heat flux and (bottom) Harvard Forest net ecosystem carbon flux. Corrections were made using a 32^2 node SOFM for Cabauw and a 12^2 node SOFM for Harvard Forest. Results are an average of an ensemble of pareto set model runs.

like the ANN to learn only *first-order* model error, without the complication of internal feedback mechanisms. That is, the state values of the previous time step in Eq. (1) would ideally be observed states so that (for the moment ignoring observational errors) the error term defined in Eq. (2) would have no dependence on the model's behavior in previous time steps. If the ANN were to be trained this way, however, during the testing period it would have to make a correction to the model states (which had been replaced by observations during training). If it did not, we would expect model states to again drift to equilibrium values, potentially a very different environment from the one in which the ANN was trained. The main limiting factor for such an approach is the relatively limited amount of observed state data. The issue was mitigated to some extent by including model states as ANN inputs. Also, in both the Cabauw and Harvard Forest cases described above, we have some evidence to indicate that model states were reasonably realistic. Top-layer soil temperature, the only state variable available for both of the datasets, was easily within 1 K of the observed value after spinup in both cases.

Perhaps the most serious criticism of the NERD process is that it is a statistical correction. One might well ask, if we believe that an ANN is capable of appropriately correcting the model, why not just use an ANN to model the land surface and dispense with the physically based model altogether? The answer is that we are modeling a dynamic climate system (assuming that our interest is long-term prognostic simulation). It is the case, whether we make a correction or not, that the model must incorporate enough of the natural system's physical processes that the mechanisms of climate change are captured. Additionally, there is not yet enough data to support a global statistical land surface model. Deciding whether a statistical correction to a physically based model is appropriate under dynamic conditions is very difficult, since we do not know exactly how dynamic the natural system actually is. We

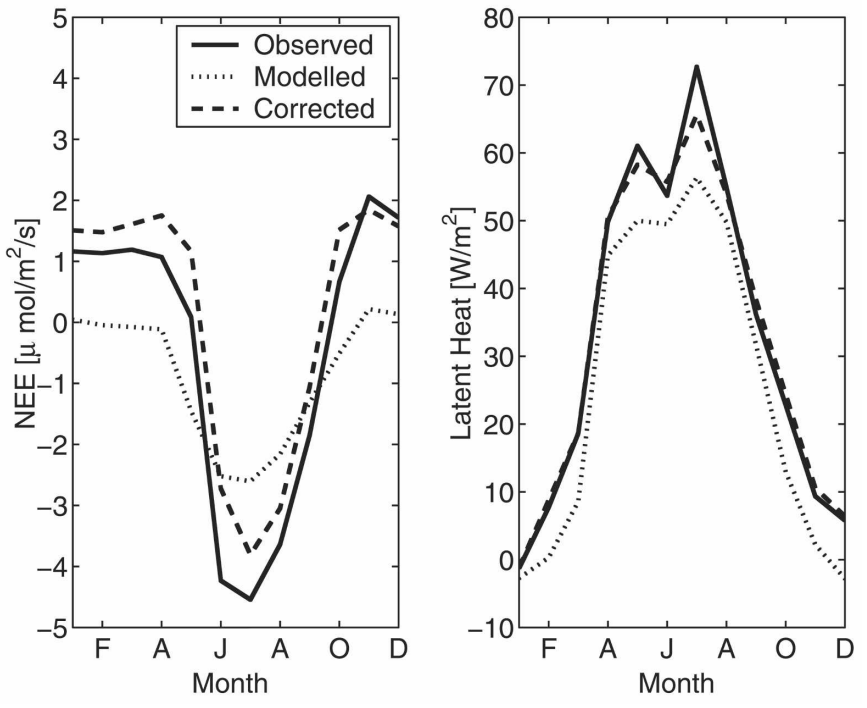

FIG. 8. Average monthly values for Harvard Forest net ecosystem carbon flux and Cabauw latent heat flux. Corrections were made using a 32^2 node SOFM for Cabauw and a 12^2 node SOFM for Harvard Forest. Results are an average of an ensemble of pareto set model runs.

must decide whether the anticipated modes of climate behavior are significantly different from those that were used to develop the statistical technique. That is, whether we have data today that includes the physical processes of climate in the future. Resolving these issues will take time and a great deal of high quality observational data, and the answer will probably be temporally and spatially dependent. They apply both to the NERD process and to parameter estimation, which has been performed using a short period of single-site observations to choose parameters for entire regions for long-term simulations (Sen et al. 2001). In the Harvard Forest case presented here however, since both parameter estimation and ANN training were per-

TABLE 2. Correlation between model residual and each of the variables selected as ANN inputs. Values of the square root of the Pearson correlation coefficient, r, as well P values (significance of the correlation) are shown for model runs before ("model") and after ("corr") NERD correction. Here P values less than 0.05 suggest significant correlation. Zero implies a value less than 10^{-100}. All pareto-point trained and tested ANNs were used with 32^2 and 12^2 node SOFMs for Cabauw latent heat flux and Harvard Forest CO_2 flux, respectively.

Cabauw latent heat correction						
	SW down	T air	Q air	Wind	Modeled LH	Modeled SM
Model r	−0.053	−0.012	0.023	0.206	−0.212	−0.042
Corr r	0.028	−0.003	0.008	0.041	−0.004	−0.040
Model P value	1.2×10^{-28}	0.010	9.5×10^{-7}	0	0	1.0×10^{-18}
Corr P value	3.1×10^{-9}	0.570	0.093	1.0×10^{-17}	0.460	1.2×10^{-16}
Harvard Forest net ecosystem exchange correction						
	SW down	T air	LAI	Modeled LH	Modeled ST	Modeled NEE
Model r	−0.103	−0.125	−0.181	−0.106	−0.156	0.050
Corr r	−0.041	0.029	0.053	0.020	0.034	−0.037
Model P value	0	0	0	0	0	3.1×10^{-79}
Corr P value	7.1×10^{-53}	1.9×10^{-27}	5.2×10^{-86}	1.3×10^{-13}	2.8×10^{-36}	2.3×10^{-44}

formed using 1992–95 data and the results used 1996–99 data, both processes seem appropriate.

This paper is intended as a simple demonstration of the ability of the NERD technique to capture (but not yet reveal) the nature of model error emanating from parameterization problems in the model. Future work will use NERD to identify weaker areas of model parameterization. Additionally, the statistical correction presented here will be extended to regional or global scales by including model parameters as inputs to the ANN as a mechanism for distinguishing between sites.

9. Conclusions

In this paper we have demonstrated the ability of the NERD process to remove a significant proportion of model error. That is, we have shown that an appropriately chosen artificial neural network can successfully identify and correct systematic trends in model output at different sites, for different variables, across a broad range of time scales. The magnitude of the correction in all cases presented here was considerably larger than that afforded by parameter calibration. For latent heat flux at the Cabauw site, the NERD process reduced per-time-step rmse from 27.5 to 18.6 W m^{-2} (32%) and monthly rmse from 9.91 to 3.08 W m^{-2} (68%). Net ecosystem carbon exchange (NEE) rmse at the Harvard Forest site was reduced from 3.71 to 2.70 μmol m^{-2} s^{-1} (27%) on a per-time-step basis and 2.24 to 0.11 μmol m^{-2} s^{-1} (95%) on annual time scales. This clearly shows that systematic error in model output does indeed exist.

We have also ensured that the gains made by the NERD correction compensate for inadequacies in model parameterization rather than problems resulting from inappropriate parameter values. The NERD tool was applied using model parameter sets that minimized error in latent heat, sensible heat, and net ecosystem carbon exchange both independently and simultaneously, as well as with default parameter sets.

This suggests that data quality is *not* a major limitation on the validation and development of land surface models. Indeed the use of observational data purely for parameter estimation at least in this case appears to be an underutilization of important information on model misbehavior, which the observational data contain. The NERD technique also dramatically enhances the breadth of data available for testing and improving land surface models since it does not require continuous observational data. Even single measurements of appropriate variables can contribute to neural network training or testing. It should be noted, however, that the work presented here represents a small sample size. It only made use of a single model and two observational sites.

Acknowledgments. The authors thank Yuqiong for providing the MOSCEM code and Steve Wofsy for helpful comments on the manuscript, together with the U.S. Dept. of Energy, Office of Science, and the U.S. National Science Foundation for Harvard Forest data and the Royal Netherlands Meteorological Institute for Cabauw data. Work here was supported by a CSIRO Postgraduate Scholarship and an Australian Postgraduate Award.

APPENDIX A

Finding the Regression Parameters

We wish to find the regression parameters, υ_{ji}, discussed in section 5. For a given node in the regression layer, let the number of input–output pairs supplied by SOFM training be p. We then need to solve a set of linear equations for $\theta = (\upsilon_{j0}, \upsilon_{j1}, \ldots, \upsilon_{jn_0})$,

$$\mathbf{Z} = \mathbf{X}\theta + \boldsymbol{\epsilon}, \tag{A1}$$

where $\boldsymbol{\epsilon}$ is a $p \times 1$ vector of estimation errors with zero mean, $\mathbf{Z}$ is the $p \times 1$ vector of p output data for training and $\mathbf{X}$ is a $p \times (n_0 + 1)$ matrix containing p rows of n_0-variable training data:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{1,1} & \ldots & x_{1,n_0} \\ 1 & x_{2,1} & \ldots & x_{2,n_0} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ 1 & x_{p,1} & \ldots & x_{p,n_0} \end{bmatrix}.$$

The column of ones in the matrix $\mathbf{X}$ is from the unit node in the input layer, and once multiplied by the υ_{j0} parameter, as suggested before, will form the constant term in the regression.

To minimize the error sum of squares, $\boldsymbol{\epsilon}^{\mathbf{T}}\boldsymbol{\epsilon}$, consider the vector $\mathbf{Z}$ as a sum of the vectors $\mathbf{X}\theta$ and $\boldsymbol{\epsilon}$ in a p-dimensional Euclidean space, as in Eq. (A1) and Fig. A1. The columns of $\mathbf{X}$, also p-dimensional vectors, span the *estimation space*, so that the product

$$\mathbf{X}\theta = [1, \mathbf{x}_1, \cdots, \mathbf{x}_{n_0}] \begin{bmatrix} \upsilon_{j0} \\ \upsilon_{j1} \\ \cdot \\ \cdot \\ \cdot \\ \upsilon_{jn_0} \end{bmatrix}$$

$$= \upsilon_{j0} + \upsilon_{j1}\mathbf{x}_1 + \cdots + \upsilon_{jn_0}\mathbf{x}_{n_0}$$

can define any vector in the estimation space for appropriate values of the υ_{ji}. From Fig. A1a it should be

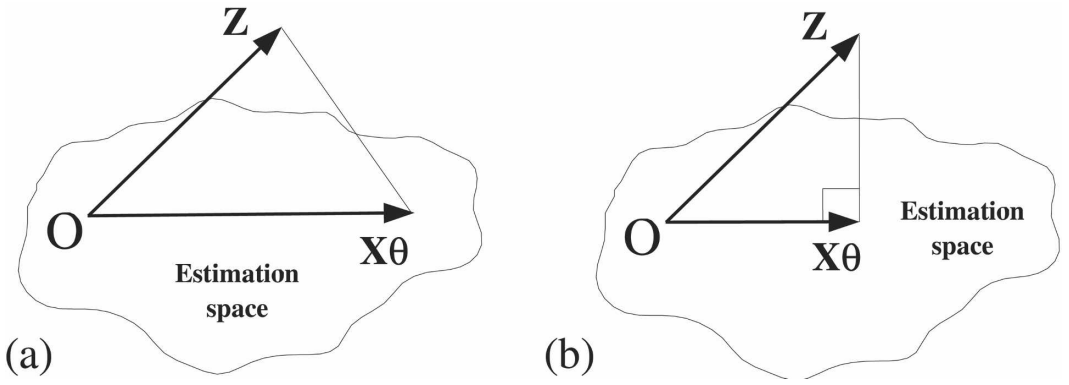

FIG. A1. The decomposition $\mathbf{Z} = \mathbf{X}\theta + \boldsymbol{\epsilon}$, where span$\{1, \mathbf{x}_1, \ldots, \mathbf{x}_{n_0}\}$ is termed the *estimation space* in (a) the general case and (b) the case where the choice of θ minimizes the length of $\boldsymbol{\epsilon} = \mathbf{Z} - \mathbf{X}\theta$.

clear that minimizing the error sum of squares $\boldsymbol{\epsilon}^{\mathrm{T}}\boldsymbol{\epsilon} = (\mathbf{Z} - \mathbf{X}\theta)^{\mathrm{T}}(\mathbf{Z} - \mathbf{X}\theta)$ is equivalent to minimizing the length of $\mathbf{Z} - \mathbf{X}\theta$. It should also be clear that the shortest distance from the point $\mathbf{Z}$ to the estimation space is the perpendicular distance, so that choosing $\mathbf{X}\theta$ to be the projection of $\mathbf{Z}$ onto the estimation space will minimize $\boldsymbol{\epsilon} = \mathbf{Z} - \mathbf{X}\theta$ (Fig. A1b). That is, the dot product of $\boldsymbol{\epsilon}$ and *every* vector in the estimation space will be zero. Since the columns of $\mathbf{X} = [1, \mathbf{x}_1, \cdots, \mathbf{x}_{n_0}]$ span the estimation space, for any $\mu \in \mathbb{R}^{n_0} + 1$

$$\begin{aligned}(\mathbf{X}\mu)^{\mathrm{T}}\boldsymbol{\epsilon} &= \mu^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\boldsymbol{\epsilon} \\ &= \mu^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}(\mathbf{Z} - \mathbf{X}\theta) \\ &= \mu^{\mathrm{T}}(\mathbf{X}^{\mathrm{T}}\mathbf{Z} - \mathbf{X}^{\mathrm{T}}\mathbf{X}\theta) \\ &= 0. \end{aligned} \tag{A2}$$

Since Eq. (A2) must be true for *any* value of μ, the *normal equations*

$$\mathbf{X}^{\mathrm{T}}\mathbf{X}\theta = \mathbf{X}^{\mathrm{T}}\mathbf{Z}$$

must hold.

APPENDIX B

Principal Components

We wish to find a solution to $\mathbf{X}^{\mathrm{T}}\mathbf{X}\theta = \mathbf{X}^{\mathrm{T}}\mathbf{Z}$ in the case that $(\mathbf{X}^{\mathrm{T}}\mathbf{X})^{-1}$ is not invertible because of collinearity between the n_0 variables (columns) of $\mathbf{X}$. To do this, we consider the principal component transformation of $\mathbf{X} = [1, \mathbf{x}_1, \cdots, \mathbf{x}_{n_0}]$, which will essentially remove the collinearity between the variables and express the data in terms of a coordinate system with perpendicular axes.

Since the covariance matrix of $\mathbf{X}$, $\Sigma = E(\mathbf{X}^{\mathrm{T}}\mathbf{X})$, is symmetric, it is orthogonally diagonalizable, so that

$$\begin{aligned}\Lambda &= \begin{bmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_1 & \ldots & 0 \\ \cdot & \cdot & \ddots & \cdot \\ \cdot & \cdot & & \cdot \\ 0 & 0 & \ldots & \lambda_{n_0} \end{bmatrix} \\ &= \mathbf{C}^{\mathrm{T}}\Sigma\mathbf{C} \\ &= \mathbf{C}^{\mathrm{T}}E(\mathbf{X}^{\mathrm{T}}\mathbf{X})\mathbf{C} \\ &= E(\mathbf{C}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{C}). \end{aligned} \tag{B1}$$

This suggests that the transformation of $\mathbf{X}$

$$\mathbf{Y} = \mathbf{X}\mathbf{C} \tag{B2}$$

provides the change of coordinates we require, since the covariance matrix of $\mathbf{Y}$, $E(\mathbf{Y}^{\mathrm{T}}\mathbf{Y}) = E(\mathbf{C}^{\mathrm{T}}\mathbf{X}^{\mathrm{T}}\mathbf{X}\mathbf{C}) = \Lambda$, is diagonal. That is, the covariance between any two variables (columns) in $\mathbf{Y}$ is zero, and the variance of each of these variables is given by the set of eigenvalues $(\lambda_1, \ldots, \lambda_{n_0})$.

We can also see that the transformation has preserved the total variance of $\mathbf{X}$:

$$\mathrm{trace}(\Sigma) = \mathrm{trace}(\mathbf{C}\Lambda\mathbf{C}^{\mathrm{T}}) = \mathrm{trace}(\Lambda) = \sum_{i=1}^{n_0} \lambda_i. \tag{B3}$$

The principal components of $\mathbf{X}$, $\mathbf{Y}$ are ordered by variance, so that $\lambda_1 \geq \cdots \geq \lambda_{n_0}$. By reducing the number of principal components to $k \leq n_0$ while still requiring that

$$\left(\sum_{i=1}^{k} \lambda_i \Big/ \sum_{i=1}^{n_0} \lambda_i\right) \times 100\% > 95\%, \tag{B4}$$

we ensure stable regression parameters at each SOLO node. Using this transformation, we have

$$\mathbf{Z} = \mathbf{X}\theta + \boldsymbol{\epsilon} = \mathbf{Y}\mathbf{C}^{\mathrm{T}}\theta + \boldsymbol{\epsilon} = \mathbf{Y}\psi + \boldsymbol{\epsilon}, \tag{B5}$$

which, using the result from appendix A, requires us to solve $(\mathbf{Y}^T\mathbf{Y})\psi = \mathbf{Y}^T\mathbf{Z}$. This time, however, we have ensured that the columns of $\mathbf{Y}$, the principal components of $\mathbf{X}$, are orthogonal, so that

$$\psi(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{Z}. \tag{B6}$$

Our original regression parameters can then be recovered:

$$\theta = \mathbf{C}\psi = \mathbf{C}(\mathbf{Y}^T\mathbf{Y})^{-1}\mathbf{Y}^T\mathbf{Z} = \mathbf{C}\Lambda^{-1}\mathbf{C}^T\mathbf{X}^T\mathbf{Z}$$
$$= \sum_{i=1}^{k} \lambda_i^{-1} e_i e_i^T \mathbf{X}^T\mathbf{Z}, \tag{B7}$$

where e_i is the eigenvector of Σ with the ith largest eigenvalue.

REFERENCES

Barford, C. C., and Coauthors, 2001: Factors controlling long- and short-term sequestration of atmospheric CO_2 in a mid-latitude forest. *Science,* **294,** 1688–1691.

Beljaars, A. C. M., and F. C. Bosveld, 1997: Cabauw data for the validation of land surface parameterization schemes. *J. Climate,* **10,** 1172–1193.

Chen, T. H., and Coauthors, 1997: Cabauw experimental results from the Project for Intercomparison of Land-Surface Parameterization Schemes. *J. Climate,* **10,** 1194–1215.

Franks, S. W., K. J. Beven, P. F. Quinn, and I. R. Wright, 1997: On the sensitivity of soil–vegetation–atmosphere transfer (SVAT) schemes: Equifinality and the problem of robust calibration. *Agric. For. Meteor.,* **86,** 63–75.

Gan, T. Y., and G. F. Biftu, 1996: Automatic calibration of conceptual rainfall–runoff models: Optimization algorithms, catchment conditions, and model structure. *Water Resour. Res.,* **32,** 3513–3524.

Gupta, H. V., L. A. Bastidas, S. Sorooshian, W. J. Shuttleworth, and Z. L. Yang, 1999: Parameter estimation of a land surface scheme using multicriteria methods. *J. Geophys. Res.,* **104,** 19 491–19 503.

——, S. Sorooshian, T. Hogue, and D. Boyle, 2002: Advances in automatic calibration of watershed models. *Calibration of Watershed Models,* Q. Duan et al., Eds., Water Science and Application Series, Vol. 6, Amer. Geophys. Union, 9–28.

Henderson-Sellers, A., A. J. Pitman, P. K. Love, P. Irannejad, and T. H. Chen, 1995: The Project for Intercomparison of Land Surface Parameterization Schemes (PILPS): Phases 2 and 3. *Bull. Amer. Meteor. Soc.,* **76,** 489–503.

Hsu, K.-L., H. V. Gupta, X. Gao, S. Sorooshian, and B. Imam, 2002: Self-Organizing Linear Output map (SOLO): An artificial neural network suitable for hydrologic modeling and analysis. *Water Resour. Res.,* **38,** 1302, doi:10.1029/2001WR000795.

Kohonen, T., 1989: *Self-Organization and Associative Memory.* Springer-Verlag, 312 pp.

Leuning, R., 1995: A critical appraisal of a combined stomatal-photosynthesis model for c3 plants. *Plant Cell Environ.,* **18,** 339–355.

——, F. M. Kelliher, D. G. G. de Pury, and E.-D. Schulze, 1995: Leaf nitrogen, photosynthesis, conductance and transpiration: Scaling from leaves to canopies. *Plant Cell Environ.,* **18,** 1183–1200.

——, F. X. Dunin, and Y. P. Wang, 1998: A two-leaf model for canopy conductance, photosynthesis and partitioning of available energy. ii. Comparison with measurements. *Agric. For. Meteor.,* **91,** 113–125.

Maier, H. R., and G. C. Dandy, 2000: Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environ. Modell. Software,* **15,** 101–124.

Martínez, M. A., and M. Velázquez, 2001: A new method to correct dependence of MSG IR radiances on satellite zenith angle, using a neural network. *Proc. 2001 EUMETSAT Meteorological Satellite Data Users' Conf.,* Antalya, Turkey.

Sen, O. L., L. Bastidas, W. Shuttleworth, Z. Yang, and S. Sorooshian, 2001: Impact of field calibrated vegetation parameters on GCM climate simulations. *Quart. J. Roy. Meteor. Soc.,* **127,** 1199–1224.

Swinbank, W. C., 1963: Longwave radiation from clear skies. *Quart. J. Roy. Meteor. Soc.,* **89,** 339–348.

Tetko, I. V., 2002: Associative neural network. *Neural Process. Lett.,* **16,** 187–199.

Vrugt, J., H. V. Gupta, L. A. Bastidas, W. Bouten, and S. Sorooshian, 2003a: Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resour. Res.,* **39,** 1214, doi:10.1029/2002WR001746.

——, ——, W. Bouten, and S. Sorooshian, 2003b: A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour. Res.,* **39,** 1201, doi:10.1029/2002WR001642.

Wang, Y. P., and R. Leuning, 1998: A two-leaf model for canopy conductance, photosynthesis and partitioning of available energy. i. Model description. *Agric. For. Meteor.,* **91,** 89–111.

Yapo, P. O., H. V. Gupta, W. Bouten, and S. Sorooshian, 1998: Multi-objective global optimization for hydrologic models. *J. Hydrol.,* **204,** 83–97.