Response letter of gmd-2020-211

Dear Editor,

Please find the revised version of our manuscript “Combining Ensemble Kalman Filter and Reservoir Computing to predict spatio-temporal chaotic systems from imperfect observations and models”, which we would like to resubmit for publication in *Geoscientific Model Development.*

Comments made by the reviewers were highly insightful. They allowed us to greatly improve the quality of the manuscript. We described the responses to the comments.

Each comment made by the reviewers is written in *italic* font. We numbered each comment as (n.m) in which n is the reviewer number and m is the comment number. In the revised manuscript, changes are highlighted in yellow in the single-column and double-spaced paper.

We trust that the revisions and responses are sufficient for this manuscript to be published in *Geoscientific Model Development*.

Sincerely,
Futo Tomizawa, Yohei Sawada

Responses to the comments of Topical Editor

(0.1) There has been a lot of recent activity in using ML for data assimilation; a more thorough review would help the authors better position this work in the current context. A few possible suggestions:

10.1007/s12551-020-00776-4

10.1137/20M1349965

10.1038/s42254-021-00314-5

→ Recent activity in using ML for NWP tasks was introduced in Section 1, but it was not discussed in the context of the general dynamical system theory. We cited more papers including ones proposed by the editor to reinforce our discussion.

> Lines 47-50: On the other hand, model-free prediction methods based on machine learning have received much attention recently. In the context of dynamical system theory, previous works have developed the methods to reproduce the dynamics by inferring it purely from observation data (e.g., Rajendra & Brahmajirao, 2020), or by combining a data-driven approach and physical knowledge on the systems (Karniadakis et al., 2021). In the NWP context, many previous studies have successfully applied machine learning to predict chaotic dynamics.
>
> Lines 512-514: As in the review by Karniadakis et al. (2021), methodologies to train the dynamics from noisy observational data by integrating data and physical knowledge are attracting attentions. In the NWP context, some studies proposed methods to combine data assimilation and machine learning to emulate the system dynamics from imperfect model and observations (e.g. Dueben & Bauer, 2018; Bocquet et al., 2019; Brajard et al., 2020; Bocquet et al., 2020), and these approaches are getting popular.

However, we have decided not to cite the second paper in the suggestion. It is a proposal of a novel data assimilation method, and it has significantly contributed to the data assimilation research. On the other hand, our proposed method in this work does not depend on specific data assimilation methods and many other methods including 4D-VAR or particle filter can be used instead of LETKF. Thus, we think that the paper is not significantly related to our paper.

Responses to the comments of Referee #4

(4.1) The construction and training of RC (and indeed of any NN) where the number of inputs/outputs equals the number of states of a geophysical model raises the question of dimensionality. The strategy works for toy models like L96, but does it scale to models with millions/billions of variables? The authors should discuss the scalability of the approach.

→ Previous studies such as Pathak, Hunt et al. (2018) showed that the parallelized reservoir computing can predict the dynamics of Kuramoto-Sivashinski system of up to 200 grid points with tractable computational cost. They also discussed the scalability of the parallelized reservoir computing scheme and applicability to the realistic NWP problems in the sequel study (Wikner et al., 2020). Although we discussed this point in the previous version of the manuscript, we have strengthened the discussion citing the latter paper.

> Lines 522-523: Although we tested our method only on 40-dimensional Lorenz 96 system, Pathak, Hunt et al. (2018) indicated that parallelized RC can be extended to predict the dynamics of substantially high dimensional chaos such as 200-dimensional Kuramoto-Sivashinski equation with small computational costs. Moreover, the applicability to the realistic NWP problems has also been discussed by their sequel study (Wikner et al., 2020). These studies imply that the findings of this study can also be applied to higher dimensional systems.

(4.2) RC-Obs, equation (17), needs to be explained in more detail. The training equation (14) seems to imply that the inputs and the outputs of the RC live in the same space. This is not the case with the first equation in (17) predicting X^f_{K+1}.

→ As the reviewer pointed out, the inputs and outputs of the RC must be in the same space, otherwise the prediction accuracy is degraded. Thus, in the equation (17), x^f_{k+1} has the same dimensionality as y^o_k, meaning that we can only predict the future state of the observable grid points. Since this point was not clarified in the previous version of the manuscript, we have added some descriptions explaining this point.

> Lines 257-259: As in equation (14), input and output of RC must be in the same space. Therefore, in this case, prediction variables $\boldsymbol{x}^f_k$ has the same dimensionality as $\boldsymbol{y}^o_k$, and the non-observable grid points are not predicted by this prediction scheme.

(4.3) Depending on how state variables are split, the parallelization of RC can be viewed as a form of localization, where the dynamics of groups of variables in the proximity of each other is trained separately from other groups. The authors need to explain this connection better, and eventually comparer against the predictions of a non-parallel RC.

→ We agree that the analogy between parallelization of RC and localization is not sufficiently discussed. We added the discussion on that analogy and then highlighted the difference between parallelized RC and ordinary RC.

> Lines 281-286: The strategy of parallelization is similar to the localization of data assimilation. As LETKF ignores correlations between distant grid points, parallelized reservoir computing assumes that the state variable of a grid point at the next time step depends only on the state variables of neighboring points. In contrast, ordinary RC assumes that the time evolution at one grid point is affected by all points in the state space, which may be inefficient in many applications in geoscience such as NWP.

(4.4) For eqn. (22), should the estimation depend on the initial state as well?

→ No, the estimation does not necessarily depend on the initial state. Even if we start the data assimilation procedure from a random initial value, we can obtain a good state estimation time series by truncating the head of the analysis time series by the appropriate amount as the "spin-up" duration. We have decided not to change the paper responding to this comment.

(4.5) Is the nonlinear output function f(r_k) in eqn (25) defined specifically for Lorenz96?

→ No, similar kinds of nonlinear transformation function is found to be suitable for other spatio-temporal chaotic systems such as Lorenz 63, Kuramoto-Shivasinski, and so on. (e.g., Pathak et al., 2017) On the other hand, other kinds of nonlinear function can also be suitable for the prediction of Lorenz 96 system (e.g., Chattopadhyay et al., 2019). We have clarified this point in the revised version of the manuscript.

> Lines 359-361: Note that the form of the transformation function can be flexible; one can use a different form of the function to predict Lorenz 96 (Chattopadhyay et al., 2019), or the same function can be used to predict other systems (Pathak et al., 2017).

(4.6) The language can be polished for a better presentation. For example, at line 231: "Although it is quadratic … it is differentiable" does not seem right. Please check throughout

→ Thank you for pointing that out. We checked the whole manuscript and revised the expressions.

Line 234: Since the objective function (14) is quadratic, it is differentiable. The optimal value can be obtained by just solving a linear equation as follows: