Evaluating the potential and challenges of an uncertainty quantification method for long short-term memory models for soil moisture predictions

Kuai Fang[1,3], Daniel Kifer[2], Kathryn Lawson[1] and Chaopeng Shen[1]

[1]Department of Civil and Environmental Engineering, Pennsylvania State University, University Park, Pennsylvania, USA.

[2]Department of Computer Science and Engineering, Pennsylvania State University, University Park, Pennsylvania, USA.

[3]Department of Earth System Science, Stanford University, Stanford, USA

Key Points:

- With proper hyperparameters and training data, Monte Carlo Dropout with a data noise term can effectively estimate prediction error.
- The network-predicted data noise term responds to added noise while the network weight uncertainty term reacts to dissimilarity.
- The quality of both the data noise term and the network weight uncertainty term can be lowered by biased training data.

Corresponding author: Chaopeng Shen, cshen@engr.psu.edu

Abstract

Recently, recurrent deep networks have shown promise to harness newly available satellite-sensed data for long-term soil moisture projections. However, to be useful in forecasting, deep networks must also provide uncertainty estimates. Here we evaluated Monte Carlo dropout with an input-dependent data noise term (MCD+N), an efficient uncertainty estimation framework originally developed in computer vision, for hydrologic time series predictions. MCD+N simultaneously estimates a heteroscedastic input-dependent data noise term (a trained error model attributable to observational noise) and a network weight uncertainty term (attributable to insufficiently-constrained model parameters). Although MCD+N has appealing features, many heuristic approximations were employed during its derivation, and rigorous evaluations and evidence of its asserted capability to detect dissimilarity were lacking. To address this, we provided an in-depth evaluation of the scheme's potential and limitations. We showed that for reproducing soil moisture dynamics recorded by the Soil Moisture Active Passive (SMAP) mission, MCD+N indeed gave a good estimate of predictive error, provided that we tuned a hyperparameter and used a representative training dataset. The input-dependent term responded strongly to observational noise, while the model term clearly acted as a detector for physiographic dissimilarity from the training data, behaving as intended. However, when the training and test data were characteristically different, the input-dependent term could be misled, undermining its reliability. Additionally, due to the data-driven nature of the model, the two uncertainty terms are correlated. This approach has promise, but care is needed to interpret the results.

1 Introduction

1.1 Time series deep learning for hydrologic predictions

Recently, we have witnessed the rise of data-driven models, including those based on deep learning (DL), across various scientific disciplines (Shen et al., 2018; Schmidhuber, 2015; LeCun et al., 2015; Goodfellow et al., 2016). In hydrology, time series DL has been employed in predictions of soil moisture (Fang et al., 2017, 2018; Fang & Shen, 2020), water level in urban water networks (D. Zhang et al., 2018), streamflow (Kratzert et al., 2018; Feng et al., 2019), water table depth (J. Zhang et al., 2018), and weather (Wilson et al., 2018), among other applications. A defining characteristic of DL is the depth of the neural network which enables intermediate layers to perform *representa-*

tion learning – automatically deriving problem-relevant features which are then used to predict the outputs (Bengio, 2009). Provided that there is enough training data, this characteristic implies that few pre-processing steps and human-defined features are needed. In some tasks, the networks can engineer better features than human experts (Schmidhuber, 2015).

In our previous work, we showed that a recurrent DL approach, called long short-term memory (LSTM), could learn from the soil moisture dynamics measured by the Soil Moisture Active Passive (SMAP) mission (Fang et al., 2017). A model trained on only one year of data can make strong predictions for another year. Despite the large number of parameters, the DL model did not overfit and was more robust than regularized linear regression and autoregressive models. With 3 years of training data, LSTM could successfully predict multi-year trends in soil moisture for years not included in the training data (Fang et al., 2018). Despite SMAP's own limitations, this flexible model can be beneficial in a data fusion setting for long-term projections. There remains a substantial potential to utilize DL to improve accuracies for various hydrologic modeling applications with other variables of interest.

1.2 Uncertainties for data-driven models

Despite significant progress with DL models for hydrology, none of the above-mentioned studies addressed model uncertainties, here referring to the estimation of prediction errors. For many practical and scientific purposes, e.g. ensemble data assimilation (De Lannoy et al., 2007) and decision support (Lamontagne et al., 2018), it is as important to obtain the confidence of a prediction as to obtain the prediction itself (Beven, 1989; Pappenberger & Beven, 2006; Ajami et al., 2008). This is even more critical for hydrologic DL models, considering the alien nature of DL models to most hydrologic users. However, no big-data work so far in hydrology has reported uncertainty estimation methods for time series DL models.

Multiple classes of methods have arisen from Bayesian probability theory to estimate uncertainties, with different advantages and disadvantages. For example, the Markov Chain Monte Carlo (MCMC) method adaptively generates new samples that gradually approach the posterior distribution of model parameters (Vrugt et al., 2008). In the context of hydrologic modeling, these models are typically process-based ones with a low-

dimensional (¡10) parameter set. The uncertainty estimate is obtained from sampling parameter sets from this posterior distribution which is incrementally improved. Unfortunately, MCMC is intractable for DL models that have orders-of-magnitude more parameters. Aside from the computational cost, another difficulty of this approach is structural errors from the forward model, as such an approach assumes that the error comes from uncertainty in the model parameters only (and not from the structure of the model), but model structure is known to strongly control the errors (Butts et al., 2004).

Uncertainty for data-driven models is not a monolithic quantity. It consists of several distinct components that can be mathematically modeled as follows. Consistent with the machine learning literature, the target variable Y (e.g. soil moisture) is a function of the input X and some random noise whose distribution has dependence on X. In other words, $Y = f(X)+\epsilon_X$. This function f is unknown and furthermore, due to measurement error, we may have a noisy version $\widetilde{X}$ of the inputs (instead of the true X) (Kavetski et al., 2006). There exists some unknown function f^* that serves as the best predictor of Y given noisy input $\widetilde{X}$, i.e. $f^*(\widetilde{X}) \approx Y$. Now, since f^* is unknown, the goal of machine learning is to approximate it using a function g with parameters W (hence we write g_W). Neural networks are known as *universal approximators* (Hornik, 1991) which means that, under mild regularity conditions that depend on a chosen error metric, any function can be approximated to any desired level of accuracy by a sufficiently large neural network with the right choice of weight parameters W^*. However, since W^* is also unknown, it must be estimated from the data, leading to network weight uncertainty. The network g_W learned from the data has weights W that are different from W^* (network weight uncertainty). To summarize, we have 3 sources of error/uncertainty: data noise (predicting Y using f^*), model mis-specification error (approximating f^* with g_{W^*}), and network weight uncertainty (approximating g_{W^*} with g_W).

Of the three uncertainty terms mentioned above, without improvement in data quality, only the data noise cannot be reduced by collecting more data. However, data noise is often related to certain attributes that are known and is thus also input-dependent. For example, in our case of learning SMAP observations (Fang et al., 2017), SMAP observations are highly uncertain in regions with large vegetation water content (VWC). Hence, the magnitude of SMAP data noise could potentially be estimated based on precipitation and land cover types. The network weight uncertainty, on the other hand, results from insufficient training data and can be reduced by more data collection (and more

effort). As the amount of training data increases, the parameters are better constrained and the prediction uncertainty decreases. The mis-specification error is more pronounced with process-based models, which impose strong constraints on the function space. If these constraints differ from the actual physics, they could be inadequate or inappropriate for the modeling task, under which condition it could be said the model is *mis-specified*. For DL models, as long as the appropriate basic architecture is selected, the effect of mis-specified structure is minor as the constraints are universal approximators. The basic architecture of deep networks such as LSTM is so versatile that these networks can approximate a large range of problems, from speech recognition (Graves et al., 2013), to handwriting synthesis (Graves, 2013), to brain wave interpretation (Kumar et al., 2019), to improving health care (Miotto et al., 2017). Hence in practice the approximation error is dominated by data noise and network weight uncertainty.

Some may recognize that the data noise and network weight uncertainty terms are sometimes referred to as the *aleatoric* and *epistemic* uncertainties in the literature of machine learning and some other domains. For example, Kiureghian and Ditlevsen (2009) asserted that "Uncertainties are characterized as epistemic, if the modeler sees a possibility to reduce them by gathering more data or by refining models. Uncertainties are categorized as aleatory if the modeler does not foresee the possibility of reducing them". This categorization is simple to grasp and is in general agreement with the machine learning literature (Kendall & Gal, 2017; Senge et al., 2014; Depeweg et al., 2017), as well as some hydrology papers (Nearing, Mocko, et al., 2016; Gong et al., 2013; Behrouz & Alimohammadi, 2018). Data-driven modelers have become accustomed to highly noisy data and have regarded such noise (after due effort in data curation) as irreducible. On the other hand, their knowledge comes from the training data and hence they regard the parameter uncertainty (of a data-driven model) as *epistemic*. However, these definitions clash with some other definitions known to hydrology. On a philosophical level, it is quite difficult to clearly define the limit of what is knowable and what is unknowable, which can be witnessed by a series of historical debates (Beven, 2016; Nearing, Tian, et al., 2016). For example, some would regard noise with data (e.g. precipitation), and observations (e.g. soil moisture readings from SMAP), as epistemic (Beven, 2016), while to a machine learning scientist they would most likely be considered aleatoric. Because the purpose of this paper is largely to evaluate the methods that estimate errors with LSTM models, we avoided the controversial terms.

1.3 Background on Monte-Carlo dropout

Here we examine Monte Carlo dropout with a data noise term (MCD+N). The first part of MCD+N, proposed by Gal and Ghahramani (2016) (hereafter called GG16), can be interpreted as measuring the disagreement among ensemble members generated by applying dropout. The second part of MCD+N is a heteroscedastic input-dependent model for observational noise, proposed by Kendall and Gal (2017) (hereafter called KG17).

The foundational ideas are:

- Dropout (Srivastava et al., 2014) is a training technique that is used to prevent overfitting in deep networks - during each iteration of back-propagation, randomly selected units are ignored. It was originally interpreted as an efficient way of simulating an ensemble of deep networks. GG16 provided another interpretation, that dropout training of deep networks was an approximation of training Gaussian process (GP) models (Rasmussen & Williams, 2005). GG16 proposed the use of dropout during prediction to create random predictions and postulated that the variability of these predictions was a good measure of network weight uncertainty. This use of dropout is called Monte Carlo Dropout (MCD). It is worth noting that this term does not seek to approximate the bias of the network.
- An second output unit can be added to the deep network to be implicitly supervised. With a proper scoring function during training, this unit can be interpreted as an estimate of the variance of the network's prediction from its original output unit. The goal of the secondary unit is to measure data noise and model it as a function of the inputs.

GG16 revealed a new and surprisingly convenient path toward estimating uncertainty for DL models. A GP models data as multi-variate Gaussian distributions with covariance functions. Without the need for sampling, a GP model could directly prescribe the predictive distribution at a new point. Earlier work showed that with the right activation functions, a neural network with one or more hidden layers and a Gaussian prior on the weights would converge in distribution to a GP as the size of the hidden layers grows to infinity (Neal, 1996; Lee et al., 2018; Matthews et al., 2018). Extending along this avenue, GG16 developed a theoretical framework casting dropout (Srivastava et al.,

2014) as an approximate GP, where the sampling of the distribution could be achieved by applying dropout during model testing.

GG16's GP interpretation of dropout training is heuristic in the sense that it involves approximations whose accuracies were not quantified (and is a subject for debate (Osband et al., 2016)). Moreover, with respect to the GP argument, it has never been systematically shown in previous studies (Gal & Ghahramani, 2016; Kendall & Gal, 2017; Vandal et al., 2018) that the MCD estimate would predict a smaller error for an instance more similar to the training dataset, and a larger error for instances that are unlike the training data. One barrier was that for the tasks examined in many DL applications, it was difficult to define and visualize proximity. Hence, the effectiveness of the MCD ensemble to quantify similarity has yet to be evidenced.

The MCD+N method is appealing due to its simplicity and its support for arbitrary network architectures. The resulting uncertainty estimates also proved useful in an image segmentation task (Kendall & Gal, 2017). Consequently, the scheme has garnered an enormous amount of popularity, which can be witnessed by the high citation count of GG16 (cited 1620 times at the time of writing this article) and KG17. However, the limitations and properties of this method have not been adequately examined. Since the input-dependent uncertainty is estimated by the trained network, it is natural to question its accuracy in the event that the test data comes from a fundamentally different distribution than the training data the network is based on, i.e., the test data is *out of distribution*. Another question is whether the combined uncertainty estimate is of high quality given representative or unrepresentative training data. This work constitutes the first report on MCD+N in hydrology and perhaps also one of the most thorough evaluations of this scheme in DL, revealing both its potential and limitations.

1.4 Research questions

The goal of this paper is not to promote the MCD+N scheme but to use experiments to evaluate the quality and limitations of the scheme for the case of soil moisture predictions, which is the first hydrologic dataset encountered by this method. While satellites provide global-scale coverage of surface soil moisture, many other hydrologic data, e.g. streamflow and groundwater levels, are available only locally. Even with satellites, there are regions beyond the scope of satellite, e.g. high latitudes and areas covered with

dense vegetation canopy. Therefore, we are concerned with the quality of MCD+N estimates when the training data is biased in only part of the domain. We ask the following questions:

(1) When the training data is representative of the spatial domain, can the MCD+N uncertainty terms help us anticipate predictive error as measured by unbiased RMSE?

(2) Do the two uncertainty estimates behave as asserted, i.e., does the data noise term respond to stochasticity in the data and does the network weight uncertainty term respond to dissimilar cases?

(3) When a network directly predicts input-dependent uncertainty via a secondary output unit, is this estimate reliable for time series that are out of the training data distribution?

(4) How are these results affected by hyperparameters such as the dropout rate and priors on the input-dependent uncertainty output units?

It is worth mentioning that the goal of this paper is not to promote the MCD+N scheme but to use carefully-designed experiments to evaluate its quality.

2 Methods and datasets

As an overview, we trained a probabilistic time series DL model to learn the level-3 SMAP surface soil moisture product. The input to this DL model included climatic forcing data and constant geophysical attributes. In addition to the SMAP product, the network also estimates the input-dependent data noise. The network weight uncertainty is then estimated via the MCD procedure, which runs many forward realizations of the stochastic dropout masks during inference (making soil moisture predictions about a new instance).

2.1 SMAP and input data

The SMAP level 3 radiometer product (L3_SM_P, version 4) measures the global surface soil moisture since April 2015, with a moisture-dependent sensing depth that is less than 5 cm. The spatial resolution of L3_SM_P is 36 km, with a revisit time of 2 to 3 days. The DL model was trained with seven climatic forcing inputs: precipitation, temperature, radiation, humidity, pressure, and wind speed (two directions). We obtained

the forcing data from North American Land Data Assimilation System phase II (NLDAS2) (Xia et al., 2015). In addition, the DL model also used static geographic attributes, e.g. soil texture and attributes, from the World Soil Information (ISRICWISE) database (Batjes, 1995), and land surface characteristics from SMAP flags.

2.2 Time series deep learning

The LSTM model used the atmospheric forcing time series and static land surface characteristics described above as inputs. Each valid SMAP pixel over the continental United States (CONUS) was treated as a training instance. Spatial autocorrelation was not explicitly modeled but could be implicitly considered due to the spatial autocorrelation in the inputs. During training, we used a mini-batch size of 100. A mini-batch bundles a small number of training instances together to perform weight updates via variations of stochastic gradient descent (typical deep learning training algorithms cycle over mini-batches while performing updates). The loss function is summed over the mini-batch. This procedure allows for more effective use of the memory of the Graphical Processor Units (GPUs).

Because surface soil moisture has short memory, each instance in the mini-batch is 30 days of data randomly taken from the available training data of a randomly selected SMAP pixel. 500 epochs were performed for a training job for our CONUS-scale experiment. An epoch has approximately the same number of forward runs as the number of instances. In our case, each epoch contains around 888 mini-batches.

Recurrent Neural Networks make use of sequential information by updating hidden states based on both inputs of the current time step and network states of previous time steps. By implementing a *memory cell* and *gates*, LSTM addressed the *vanishing gradient* issue that has prevented effective training for vanilla recurrent networks (Hochreiter & Schmidhuber, 1997). While there are several versions of LSTM units, we use the one specified by the following equations:

$$\text{(input transformation)} \quad x^{(t)} = ReLU(W_{xx}x_0^{(t)} + b_{xx}) \tag{1}$$

$$\text{(input node)} \quad g^{(t)} = \tanh(\mathcal{D}(W_{gx})x^{(t)} + \mathcal{D}(W_{gh})h^{(t-1)}) + b_g \tag{2}$$

$$\text{(input gate)} \quad i^{(t)} = \sigma(\mathcal{D}(W_{ix})x^{(t)} + \mathcal{D}(W_{ih})h^{(t-1)}) + b_i \tag{3}$$

$$\text{(forget gate)} \quad f^{(t)} = \sigma(\mathcal{D}(W_{fx})x^{(t)} + \mathcal{D}(W_{fh})h^{(t-1)}) + b_f \tag{4}$$

$$\text{(output gate)} \quad o^{(t)} = \sigma(\mathcal{D}(W_{ox})x^{(t)} + \mathcal{D}(W_{oh})h^{(t-1)}) + b_o \tag{5}$$

$$\text{(cell state)} \quad s^{(t)} = \mathcal{D}(g^{(t)}) \odot i^{(t)} + s^{(t-1)} \odot f^{(t)} \tag{6}$$

$$\text{(hidden gate)} \quad h^{(t)} = \tanh(s^{(t)}) \odot o^{(t)} \tag{7}$$

$$\text{(output layer)} \quad f^{(t)} = W_{hy}h^{(t)} + b_y \tag{8}$$

The superscript t refers to the time step. For a time step t, the vector of raw inputs is $x_0^{(t)}$, the state of the hidden cells is denoted by $h^{(t)}$, the state of memory cells is denoted by $s^{(t)}$, and the output of the network by $f^{(t)}$. $ReLU$ refers to Rectified Linear units (Glorot et al., 2011). In this equation, σ and $tanh$ refer to sigmoid and hyperbolic tangent functions, respectively, and they are used as the activation function in the network. $\odot$ represents point-wise multiplication. The W's and b's are the trainable connection weights and constant bias parameters in the network, which are shared by all time steps. $\mathcal{D}$ is the Dropout operator (Srivastava et al., 2014), which randomly sets some of the network connections to zero in order to reduce overfitting. During each iteration, the dropout mask is randomly initialized and remains the same for all time steps. More details of dropout are provided in Section 2.3.2.

2.3 Probabilistic LSTM Model

Overall, the uncertainty of the model is comprised of an input-dependent data noise term (Section 2.3.1) and a network weight uncertainty term (Section 2.3.2), following Kendall and Gal (2017). We let the DL network learn and predict the variance of the input-dependent uncertainty based on inputs to LSTM. Network weight uncertainty results from insufficient training data, and according to GG16, is estimated by Monte Carlo Dropout.

2.3.1 Input-dependent data noise

It is well known that SMAP observations are highly uncertain in regions with high vegetation water content (VWC) due to instrumental limitations. This kind of uncer-

tainty can be captured based on many input variables such as vegetation cover and temperature. However, instead of manually prescribing a model for the error, we let the network estimate it and provide it as an output, following KG17. For a model prediction f, the corresponding observation and error vectors are y and $\epsilon = y - f$, respectively. We assume the errors come from a Gaussian distribution, with a variance σ_x^2 that is dependent on the input data x: $\epsilon \sim \mathcal{N}(0, \sigma_x^2)$ and $y \sim \mathcal{N}(f, \sigma_x^2)$. Given n data points (regardless of space or time) $\mathbf{y} = \{y_1, ..., y_n\}$ and corresponding model predictions $\mathbf{f} = \{f_1, ..., f_n\}$ and standard deviations $\sigma_\mathbf{x} = \{\sigma_{x,1}, ..., \sigma_{x,n}\}$, the likelihood function is

$$p(\mathbf{y}|\mathbf{f}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_{x,i}^2}} \exp[-\frac{(y_i - f_i)^2}{2\sigma_{x,i}^2}] \tag{9}$$

We ask the LSTM model to output an estimate variance, $\hat{\sigma}_x^2$, for σ_x^2. For numerical stability, the network will predict $s = \log(\hat{\sigma}_x^2)$. Hence, the LSTM model will have two nodes at the output layer: $(\mathbf{f}, \mathbf{s}) = F^W(x)$, where F^W is the trained LSTM model and W is the weight in the network. There is no directly supervising data for s. Rather, it is implicitly supervised by the regression task. As the network cannot reduce random errors that cannot be predicted based on the inputs, it is forced to learn the error magnitude. For N SMAP pixels (N is the mini-batch size during training), each with T time steps, the loss function $\mathcal{L}$ to be minimized is the negative logarithm of Equation 9 across the data points:

$$\mathcal{L} = \frac{1}{2} \sum_{i=1}^{N} \sum_{t=1}^{T} \mathbf{1}_{i,t}[(y_{i,t} - f_{i,t})^2 \exp(-s_{i,t}) + s_{i,t}] \tag{10}$$

where i and t are the spatial and temporal indices, respectively, and $\mathbf{1}_{i,t}$ is 1 when there is a valid SMAP observation and 0 when there is not. Naturally, the $s_{i,t}$ term also serves as a regularization term to prevent the training from unreservedly decreasing the $\exp(-s_{i,t})$ term to minimize the loss function.

2.3.2 MCD for network weight uncertainty

Each weight update step consists of a forward pass (in which the prediction of the network is computed) and a back-propagation pass (in which this information is used to compute an approximate gradient for updating the weights). In the dropout method, a randomly chosen set of nodes is ignored for each weight update step (the ignored nodes do not affect the prediction in the forward pass). The choice of which nodes to keep and which to (temporarily) drop is implemented via the *dropout mask*.

GG16 proposed the use of dropout during the test step (inference) to generate random predictions. MCD runs M forward realizations, $f^{\widehat{W}_j}, j \in 1, ..., M$, with each set of weights $\widehat{W}_j$ obtained by randomly sampling dropout masks at the same locations where dropout is applied during training. In contrast, the normal use of dropout during inference would turn the dropout operators into a multiplication operation with constant scalars related to the dropout rate, with all connections enabled. The average of the MCD realizations becomes the overall prediction, and their variance is interpreted as a measure of uncertainty. GG16 recommended that MCD only be used for networks that are also trained using dropout. The mean and variance of the MCD ensemble for a prediction f are:

$$E[\mathbf{f}] \approx \frac{1}{M}\sum_{m=1}^{M} f^{\widehat{W}_m}(x) \tag{11}$$

$$\sigma^2_{mc}[\mathbf{f}] \approx \frac{1}{M}\sum_{m=1}^{M} f^{\widehat{W}_m}(x)^2 - E[\mathbf{f}]^2 \tag{12}$$

MCD can be interpreted intuitively from an ensemble simulation perspective, just like dropout training (Srivastava et al., 2014). Each realization of the dropout mask forms a sub-network. The random predictions arising from multiple randomly chosen masks can then be viewed as predictions coming from an ensemble of related sub-networks. These sub-networks would be in stronger agreement (hence smaller variance) in regions where the input space is well conditioned by known data points. Further away from the training data, the sub-networks may diverge more significantly. Nevertheless, it is very challenging to formally prove this intuition.

The primary contribution of GG16 was that they noted connections between dropout training and variational Bayesian inference of GP (an overview of their arguments and a discussion of issues can be found in Appendix A). Their main argument was that if variational inference was conducted with respect to network weights, with a special set of variational distributions, it would approximately lead to the same loss function as dropout training with mini-batching, as described in Equation 10. In this way, each realization with a set of randomly sampled dropout masks is equivalent to sampling from the posterior variational distribution. Although the approximation error was generally not quantified, this connection inspired their proposal of using MCD as an estimate of model uncertainty (since this is what the posterior distribution of a GP corresponds to). In computer vision tasks, GG16 and KG17 found that MCD was useful as an uncertainty mea-

sure – the estimated uncertainty tended to be large when the prediction of the network was inaccurate.

2.3.3 Combining uncertainties

In their analysis of the connections between GP and MCD in deep networks, GG16 noted that the variance of the posterior distribution depends on the variance of the prior as well as the dropout retention rate β. These factors suggest that the network weight uncertainty term needs to be calibrated. GG16 suggested linearly scaling the model uncertainty term to match the predictive error magnitude, i.e.,

$$\sigma_{mc}^2(f_{i,t}) \approx \alpha \left\{ \frac{1}{M} \sum_{m=1}^{M} f_{i,t}^{\widehat{W}_m}(x)^2 - \left[\frac{1}{M} \sum_{m=1}^{M} f_{i,t}^{\widehat{W}_m}(x) \right]^2 \right\} \tag{13}$$

Another option is to find β^*, the optimum β value, to best capture the correct uncertainty magnitude, i.e.,

$$\sigma_{mc}^2(f_{i,t}) \approx \frac{1}{M} \sum_{m=1}^{M} f_{i,t}^{\widehat{W}_m(\beta^*)}(x)^2 - \left[\frac{1}{M} \sum_{m=1}^{M} f_{i,t}^{\widehat{W}_m(\beta^*)}(x) \right]^2 \tag{14}$$

Here $f_{i,t}^{\widehat{W}_m(\beta^*)}(x)$ is the prediction for input x when the network uses the weight parameters $\widehat{W}_m$ obtained by applying dropout with rate β to the trained network.

Given $y \sim \mathcal{N}(f, \sigma_x^2)$ and the model uncertainty as calculated in Equation 12, the total uncertainty variance is σ_{comb}^2:

$$\sigma_{comb}^2 = \sigma_{mc}^2 + \sigma_x^2 \tag{15}$$

where (i, t) are dropped for brevity.

The hyperparameter β^* or α, depending on which calibration method was chosen, needs to be tuned. For the scope of this work, we chose to tune β^* as it is a simpler procedure, and we found a constant β^* to be sufficient for improving the quality of the uncertainty. We used the first year of the SMAP data as training data, and the second year as the validation data for hyperparameter tuning. Hyperparameters were adjusted so that the estimated combined error σ_{comb}^2 matched the predictive error in the spatial regions where the model was trained. To avoid over-tuning, we did a lazy search (meaning without sophisticated searching) for a uniform β^* value in all layers and locations, although we recognize that β^* could, in theory, be different from location to location. The third year of SMAP data was used as a test dataset entirely for the purpose of evaluation.

2.4 Evaluation of the uncertainty quality

In all of our experiments, we used the level-3 SMAP surface soil moisture product over the CONUS as the training target. As mentioned earlier, we used the first year of data (2015/04 - 2016/03) as the training data, the second (2016/04 - 2017/03) for validation and hyperparameter tuning, and the third (2017/04 - 2018/03) as the test data for the evaluation of metrics. The quality of uncertainty was evaluated by both the predictive errors and the cumulative distribution of the likelihood function. For the predictive errors, we compared the magnitude of σ_{comb}, the standard deviation of the combined errors, to that of the unbiased root-mean-square error (ubRMSE) when predicting SMAP surface soil moisture in the test period. We also calculated the Pearson's correlation coefficient (R) between ubRMSE and σ_{comb}.

Similar to KG17 and Vandal et al. (2018), we calculated an error exceedance likelihood, $p_{ee}(|e| > |y - f|; \sigma^2) = 1 - \frac{\text{erf}(-|y-f|)}{2\sigma}$, $e \sim \mathcal{N}(0, \sigma^2)$, which is the self-assessed chance that an error of this magnitude ($|y - f|$) or worse could happen, given an uncertainty estimate σ^2. By this definition, if the uncertainty estimate is perfect, for a large error marked with a 0.01 exceedance likelihood, we expect to see that it is exceeded roughly 1% of the time. Similarly, for an error estimate exceeded 40% of the time, we expect to see a calculated error exceedance likelihood of 0.4. As a result, when the cumulative distribution function (CDF) of p_{ee} is plotted (called the calibration plot in KG17), we would like to see it being close to a one-to-one line. We further calculated d, the maximum distance of the CDF from the 1:1 line, also called the Kolmogorov-Smirnov distance between two empirical CDFs. d thus serves as a succinct measure of the quality of the uncertainty estimate. A d value of 0 would mean a perfect uncertainty quality, while a d value close to 0.5 would suggest very poor quality. The error exceedance likelihoods calculated using σ_x, σ_{mc}, and σ_{comb} as σ^2 are referred to as p_x, p_{mc}, and p_{comb}, respectively. Evaluating p_{ee} separately with these variances helps us to understand how each component of the uncertainty estimate works.

2.5 Training experiments and evaluations

2.5.1 CONUS-scale generalization test

We trained a LSTM model over the entire CONUS from 2015/04 to 2016/03, with spatial downsampling done by picking 1 pixel from every patch of 2 x 2 pixels. To eval-

uate the overall quality of the uncertainty estimation, we ran both a temporal test and a regular spatial test. In the temporal generalization test, the model was tested on the same pixels as the training set but with the third year of data (2017/04 to 2018/03). In the regular spatial generalization test, the model was tested on the same period as the training set, but with the neighboring pixel in the diagonal direction, which was not part of the model's training data.

2.5.2 Noise perturbation experiments

According to the theory discussed by KG17, the input-dependent data noise term could directly detect observation error, while the model parameter uncertainty could not. To test this theory, we examined how the input-dependent data noise (σ_x) and network weight uncertainty (σ_{mc}) each responded to noise introduced to the learning target. Here we prescribed an independent zero-mean Gaussian relative noise value with variance σ^2_{noise}, which was added to the observation data as

$$y_{noise} = y + \mathcal{N}(0, \sigma^2_{noise}) \tag{16}$$

Ten independent models were trained by adding different levels of noise as $\sigma_{noise} \in \{0.1, 0.2, ..., 1.0\}$. The results of the noise perturbation experiments are presented in Section 3.2.

2.5.3 Spatial extrapolation experiments

As discussed earlier, a primary objective of uncertainty analysis is to measure the model confidence when making predictions for new and potentially unfamiliar instances. For example, a GP assigns high posterior uncertainty to instances that are dissimilar from the training data and low posterior variance to instances that are similar. Ideally, a neural network trained with dropout would exhibit similar behavior.

Thus we tested how the proposed uncertainty estimates respond to instances similar to (or dissimilar from) the training dataset with two sets of experiments. Similarity, defined as the proximity between instances in a space spanned by inputs that are relevant to the prediction target, can be difficult to judge, so here we use geographic proximity and ecoregion hierarchy as proxies. Based on US Environmental Protection Agency (EPA) Ecoregions, which are areas where ecosystems are generally similar (McMahon et al., 2001), we divided the entire CONUS into 17 sub-regions of relative similar sizes. To achieve this, we broke the largest ecoregion into several smaller ones and merged the

smallest ecoregions into bigger ones. The ecoregions are hierarchical, i.e., ecoregions under the same level-1 or level-2 codes will be more similar to each other than the ones with different level-1 or level-2 codes. These ecoregions represent a wide diversity of landscapes, land covers, soils, and climates over the CONUS.

In the first set of experiments, we trained a LSTM model on each of the ecoregions using year one data, adjusted hyperparameters on these training ecoregions using year two data, and examined standard deviations for data noise (σ_x), $network weight uncertainty(\sigma_{mc})$, and combined uncertainty σ_{comb} when the model was tested in other regions with year three data. Our hypothesis was that if MCD indeed captures the network weight uncertainty, then σ_{mc} should be small in regions similar to the training region and large in dissimilar regions. For comparison, we also attempted a different division strategy, 18 level-2 hydrologic cataloging units (HUC2), and show the results in the Appendix.

In the second set of experiments, we trained the models on several combinations of ecoregions. Some of these ecoregion combinations are dispersed throughout different parts of the CONUS (hence were more likely to be representative of the background testing data), while three of the combinations were clustered towards only part of the CONUS (hence were more likely to be biased). These tests allowed us to examine whether useful uncertainty measures could be produced using a small subset of available data.

3 Results and Discussion

3.1 Uncertainty quality

We first examined the impacts of the dropout retention rate β on uncertainty estimates and predictive error. The network weight uncertainty was clearly a function of β, and we found $\beta \approx 0.4$ to be an approximate value that enabled both accurate predictions and high-quality uncertainty estimates during the validation period (Appendix B, Figure B.1). This was the case for either CONUS-scale models or regional-scale models. To avoid fine tuning, we used $\beta = 0.4$ for all of our evaluations. This result also suggests that it is useful to calibrate the network weight uncertainty before using it to anticipate errors.

The spatial patterns of both data noise (σ_x) and model uncertainty (σ_{mc}) agreed more or less with the predictive metric of unbiased root-mean-square error (ubRMSE), and were larger in the eastern CONUS than in the western CONUS (Figure 1 maps).

In particular, the northern central CONUS and northeast and northwest coastal regions had large ubRMSE along with large σ_x. The eastern half of the CONUS, in general, had larger annual precipitation than the western half. The magnitudes of soil moisture fluctuations, and consequently the magnitudes of measurement errors, were larger. In the northern CONUS, forest land cover is prominent and a larger fraction of precipitation falls as snow, so the SMAP signal is adversely impacted by large vegetation water content (VWC) (O'Neill et al., 2016). Soil moisture cannot be accurately sensed below freezing conditions, which further reduces the amount of training data available (Fang et al., 2018). As a result, the northeastern and northwestern (along the Rocky mountains) forests had the highest ubRMSE. The lowest errors were found on the Great Plains and in the southeastern CONUS, due to arid conditions and reduced forest cover, with associated low VWC. The predicted σ_x automatically captured these spatial patterns. A belt-like region with large errors was found along the Mississippi River, which descends along curved state boundaries into the Gulf of Mexico in the south. This large noise may be associated with (i) signal leakage from the Mississippi River; or (ii) extensive irrigation due to cultivated crops along the Mississippi, but, interestingly, σ_x captured it nonetheless.

On scatter plots of these results, we note a high Pearson's correlation coefficient value (R=0.84) between ubRMSE and σ_{comb} with a small under-estimation bias (Figure 1c). For the regular spatial generalization test, the correlation was still around 0.79 (Figure 1i). The relationship between σ_{comb} and ubRMSE was heteroscedastic, with more spread toward the wetter range. In addition, we found that σ_x was larger than σ_{mc} in both cases, but the two terms were correlated (Figure 1f, Figure 1l).

These results suggest that for cases of temporal prolongation or mild spatial extrapolation, it is possible to anticipate model predictive errors using σ_{comb}, while using either σ_x or σ_{mc} alone would result in under-estimation of the error. In particular, we can anticipate that if the predicted σ_{comb} is below 0.03, the actual model error will be closely bounded to the range of $0-0.03$. When σ_{comb} is larger than 0.05, however, we should anticipate large errors, even though ubRMSE may be coincidentally small. The results suggest that we can use the σ_{comb} map to identify regions where SMAP does not function properly. In addition, as observed by Pan, Cai, Chaney, Entekhabi, and Wood (2016), the low uncertainty in the southeast coastal plains is noteworthy. The small error indicates that SMAP has a reasonable value in this region.

The calibration plots of error exceedance likelihoods (Figure 2) show the quality of each uncertainty-estimating component. p_{mc} in both panels lies above the 1:1 line toward the left end (e.g. for a p_{mc} of 0.2, a cumulative frequency of $\tilde{0}.39$ is obtained), which means that large predictive errors occurred more frequently than anticipated. Hence, the pattern means that σ_{mc} alone under-estimated the uncertainty toward the large-error range. On the other hand, if we had only considered σ_x, the uncertainty would be slightly under-estimated. In both validation and temporal tests, σ_{comb} was closer to the one-to-one line than either individual component. Since the validation period was employed to identify the optimal β, p_{comb} was almost perfect. In the test period, there was a slightly bigger gap between p_{comb} and the 1:1 line, but the difference still remained small, with a KolmogorovSmirnov distance of 0.027.

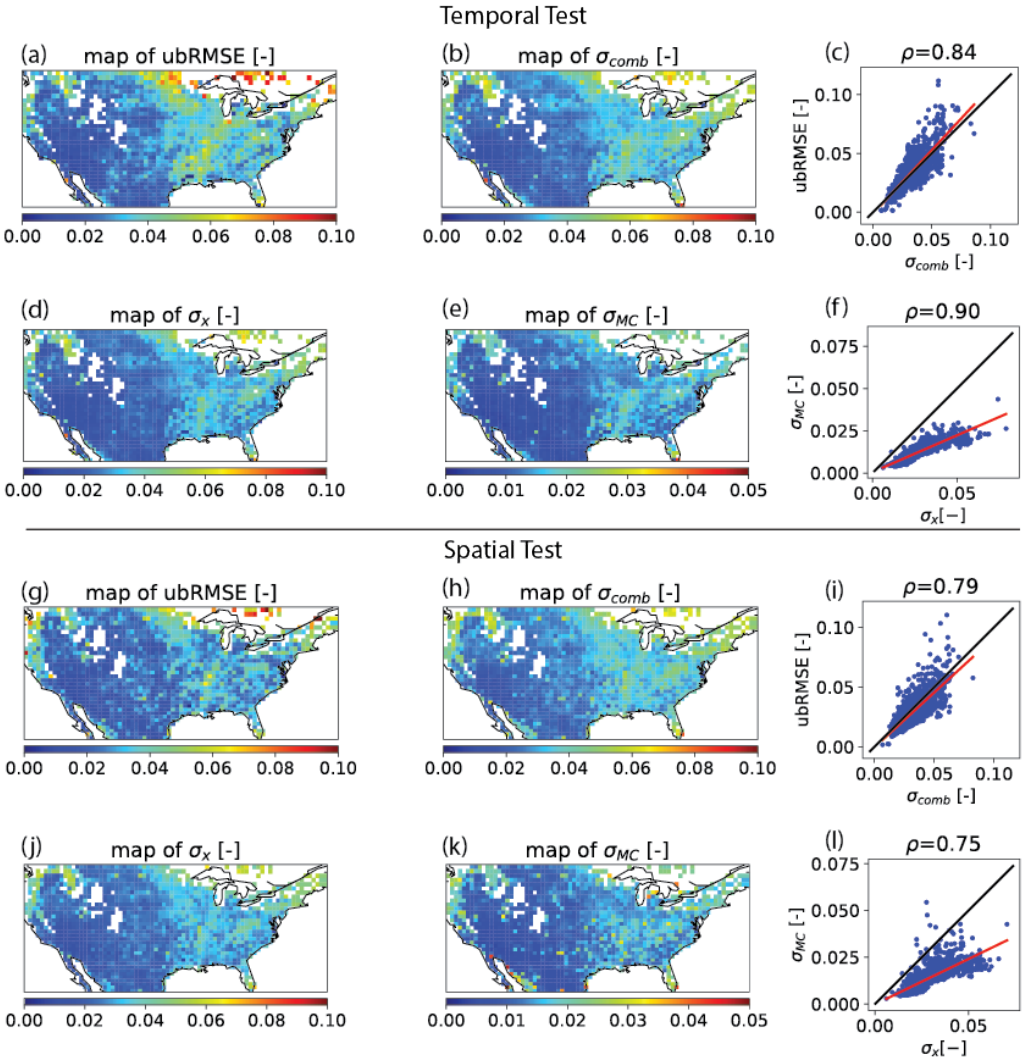


Figure 1. Model error and uncertainty estimates of temporal and spatial generalization tests over the CONUS. The top two rows (a-f) show temporal test results, and the bottom two rows ((g)-(l)) show spatial test results. For each of these tests, the left two columns show maps of model test error (unbiased root-mean-square error, $ubRMSE$) and three uncertainty estimates: data noise (σ_x), network weight uncertainty (σ_{mc}), and combined uncertainty (σ_{comb}). Note that the plots of σ_{mc} ((e), (k)) have a narrower numeric range for the same color range as the other uncertainty estimates, as the range of σ_{mc} is smaller than those of the others. For the two maps in each row, the one-to-one comparison is shown on the right column, with each point corresponding to one pixel on the maps, red lines representing lines of best fit, and black lines representing $y = x$.

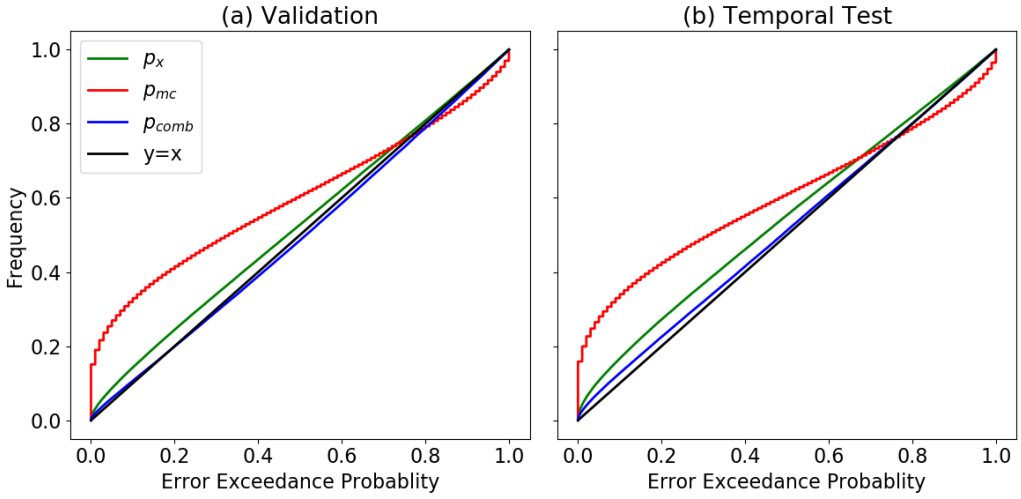


Figure 2. Calibration plots of error exceedance likelihoods computed using network weight uncertainty (p_{mc}), data noise (p_x), and combined error (p_{comb}) for the (a) validation set (2016/04-2017/03) and (b) test set (2017/04-2018/03) of the CONUS-scale temporal generalization test. x-axes are estimated error exceedance likelihoods (p_{ee}) based on the different variances given, and y-axes are the cumulative frequencies, so these curves are the cumulative distribution functions (CDFs) of p_{ee}, given an uncertainty estimate. The left end of the x-axis represents large errors, and the right end represents smaller errors. An ideal uncertainty estimate would produce a CDF that is identical to a 1:1 plot (black lines). The uncertainty qualities, d values (maximum distance of the CDF from the 1:1 line, section 2.4), of p_x, p_{mc}, and p_{comb} were 0.045, 0.230, and 0.015 for the validation set, and 0.072, 0.241, and 0.027 for the temporal test, respectively.

3.2 Responses of uncertainty estimates to noisy targets

The observation that the two uncertainty estimates were correlated needed further investigation. Were they correlated because they partially measured the same type of uncertainty, or because the presence of different uncertainties themselves were correlated in the SMAP prediction task? In other words, were they correlated because regions with smaller amounts of training data (leading to larger network weight uncertainties) also tended to have higher data uncertainties? We thus added noise into the observations to increase the apparent data uncertainty. In the ideal case, this would cause σ_{mc} to remain unchanged and σ_x to increase by the same amount as the noise.

When the model was trained on the whole CONUS without added noise, the median ubRMSE was around 0.03, smaller than the design accuracy of SMAP. When we added Gaussian random noise, test error and estimated uncertainties all increased. σ_{comb} maintained roughly the same magnitude as $ubRMSE$, with a slight under-estimation (Figure 3a). σ_x responded much more strongly to noise than σ_{mc}, which shows that the proposed data noise scheme is effective at estimating random noise with the target. LSTM could not predict the random noise, and the part that was uncapturable was correctly attributed to the data noise term, especially toward the high noise levels. This result shows that this decomposition of uncertainty could be reasonable at least when the training data are representative.

We note in Figure 3a that σ_{mc} also increased with noise, albeit gradually. This observation is consistent with the spatial patterns shown in Figure 1 and the correlation between the two uncertainty terms, and is not in conflict with the meaning of the two terms. Unsurprisingly, significant observational noise led to reduced useful supervising data and thus more ambiguous network weights. Even though σ_{mc} can, in theory, be reduced by the addition of more data, when noise is significant, the demand for data is amplified. As a result, the resulting training data is not sufficient at high noise levels.

We wanted to see how the quality of two uncertainty estimates changed with the noise in observational data. As Figure 3b and c show, the quality of σ_x increased with noise, as the data noise component could explain more of the total uncertainty. The network weight component, on the contrary, was less and less important with respect to the total error. This observation agrees with the naming of the data noise term.

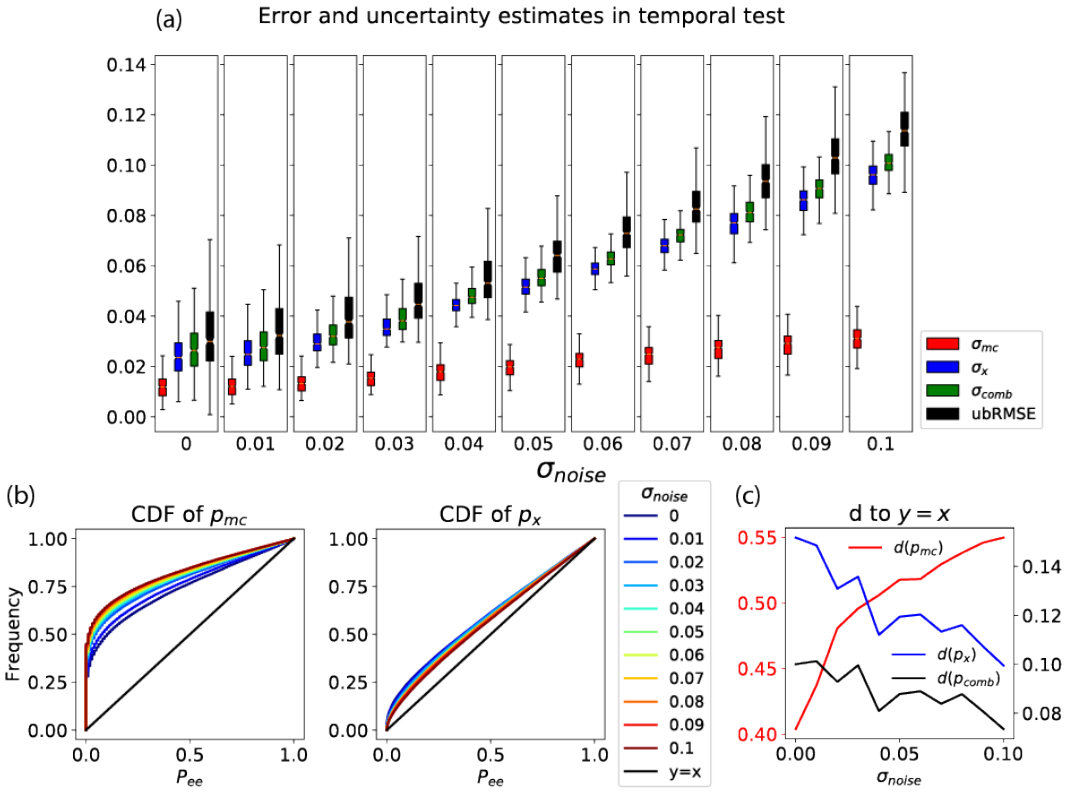


Figure 3. Performance of model trained by noise-added observations. (a) shows matrices of uncertainty estimates for network weight uncertainty (σ_{mc}), input-dependent data noise (σ_x), and combined uncertainty (σ_{comb}), as well as test error ($ubRMSE$). (b) shows calibration plots of error exceedance likelihoods for different noise levels (p_{mc}, p_x). (c) shows the uncertainty quality, d (the maximum distance between each CDF and the one-to-one line), varied with noise added to observations. $d(p_{mc})$ is plotted using the left y-axis while $d(p_x)$ and $d(p_{comb})$ are plotted using the right y-axis.

3.3 Response of uncertainty estimates to dissimilarity

The results in Sections 3.1- 3.2 were obtained from models trained on the entire CONUS. In the following sections we show results from models trained over parts of the CONUS, which explore how the uncertainty terms respond to out-of-training instances. We questioned whether the network parameter uncertainty adequately captured dissimilarity.

Overall, we see a clear influence of geographic proximity on network weight uncertainty, σ_{mc}, as a result of spatial autocorrelation in the attributes. When we tested mod-

els that were only trained on a single level-2 ecoregion, σ_{mc} was smallest inside the training region, somewhat larger in neighboring regions, and much larger further away (Figure 4). We only show models trained on four of the level-2 ecoregions here, but other cases behaved similarly. We show several results in Figure C.1 in Appendix C with similar results when using HUC2 as training regions. These results provided the clearest visual evidence so far that MCD does detect dissimilarity.

However, spatial distance itself was not the causal factor for autocorrelation. There is a visible contrast along the eastern edge of the training ecoregion in Figure 4b. This gradient shows where the Great Plains descends to the central plains, and also the divide between the drier western half and the wetter eastern half. Some pixels immediately adjacent to the east of the training ecoregion had much larger σ_{mc} than the western neighboring pixels, which suggests the model used precipitation and temperature as important factors in deciding similarity in terms of soil moisture dynamics.

It is important to remember that σ_{mc} also depends on the training data, so while it tends to be reciprocal, it may not always be. For example, when the model was trained on ecoregion 8.3 (Southeastern Plains, 4a), it regarded the the western coastal regions and some parts of the southwestern hot desert (parts of ecoregion 10.2, which is the red-highlighted training region selected in Figure 4d) as being similar, and regarded the northern high plains (including ecoregion 9.4 and 10.1, which are training regions highlighted in Figure 4c and d, respectively) as being dissimilar. As expected, models trained on ecoregion 9.4 and 10.1 (results shown in Figure 4c and d) also identified ecoregion 8.3 (training region in 4a) as being dissimilar. However, the model trained on ecoregion 10.2, most of which was found to be similar to ecoregion 8.3 by the model in Figure 4a, regarded the ecoregion 8.3 as dissimilar. This might be due to the more homogeneous environment of ecoregion 10.2 (hot desert). When a model is trained here, it has limited knowledge of what soil moisture may do in a wetter environment. When the model was trained in ecoregion 8.3 (wetter and relatively more diverse), it was trained on data with larger gradients in rainfall and appeared to be more confident to predict in ecoregion 10.2.

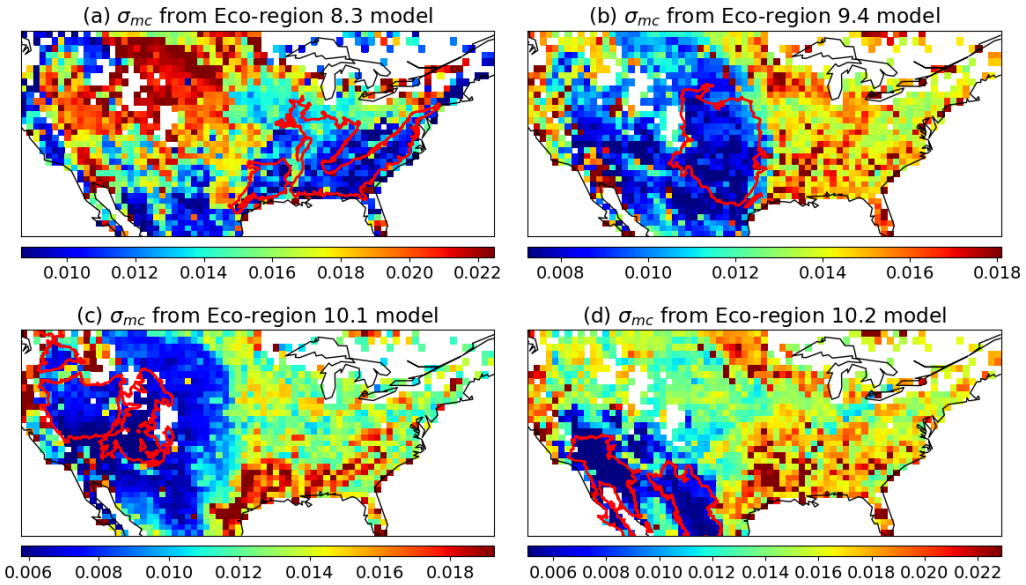


Figure 4. Maps of network weight uncertainty (σ_{mc}) when the LSTM model was trained on single level-2 ecoregions. The training region for each model instance is highlighted by the red polygon. The four selected ecoregions are a) 8.3 Southeastern Plains; b) 9.4 South-central Semiarid Prairies; c) 10.1 Cold Deserts; d) 10.2 Warm Deserts

The responses to similarity can become more clear via bar plots based on the ecoregion hierarchy (Figure 5), where the model was trained on one level-2 ecoregion and tested on another one belonging to the same level-1 ecoregion (the close ecoregion), and another one belonging to a different level-1 (the far ecoregion). In all three cases, σ_{mc} was much larger for the far ecoregions as compared to the close ones. Similar to what was suggested in Figure 4, σ_{mc} correctly provided warnings for instances that were dissimilar to the training region, and could discern that one region was more dissimilar than another.

In contrast, σ_x was not controlled by ecoregion similarity, but represented a prediction of the error based on the inputs, especially precipitation. The predictions seemed to be largely correct when we qualitatively examined Figure 5, although they may not be quantitatively perfect. In case (a), σ_x was smaller for both close and far ecoregions than for the training ecoregion (Figure 5a). Here the model was trained in the northeastern region, which has heavy forest cover and more months in a year with frozen soil, and thus large measurement error. It was tested in ecoregion 10.2, which has much drier conditions, and should therefore have smaller errors. This was reflected in the smaller

σ_x for ecoregion 10.2, but we would have expected the σ_x to be even smaller than the actual estimate. In case (b), σ_x was similar for the training and the close ecoregions, and larger for the coastal ecoregion of 11.1 (Figure 5b). Ecoregion 11.1 has larger rainfall than the inland regions and thus larger error, which was correctly captured by σ_x. In case (c), the model was trained in a drier region and tested in ecoregion 8.4, which is both different (higher σ_{mc} expected) and much wetter (higher σ_x expected). Therefore, σ_x and σ_{mc} seemed to indeed reflect different parts of the uncertainty and agreed with our expectation in terms of the general patterns, but quantitatively the quality could be limited by the training data (Figure 5c). We show similar results from HUC2 training regions in Figure C.2 in Appendix C.

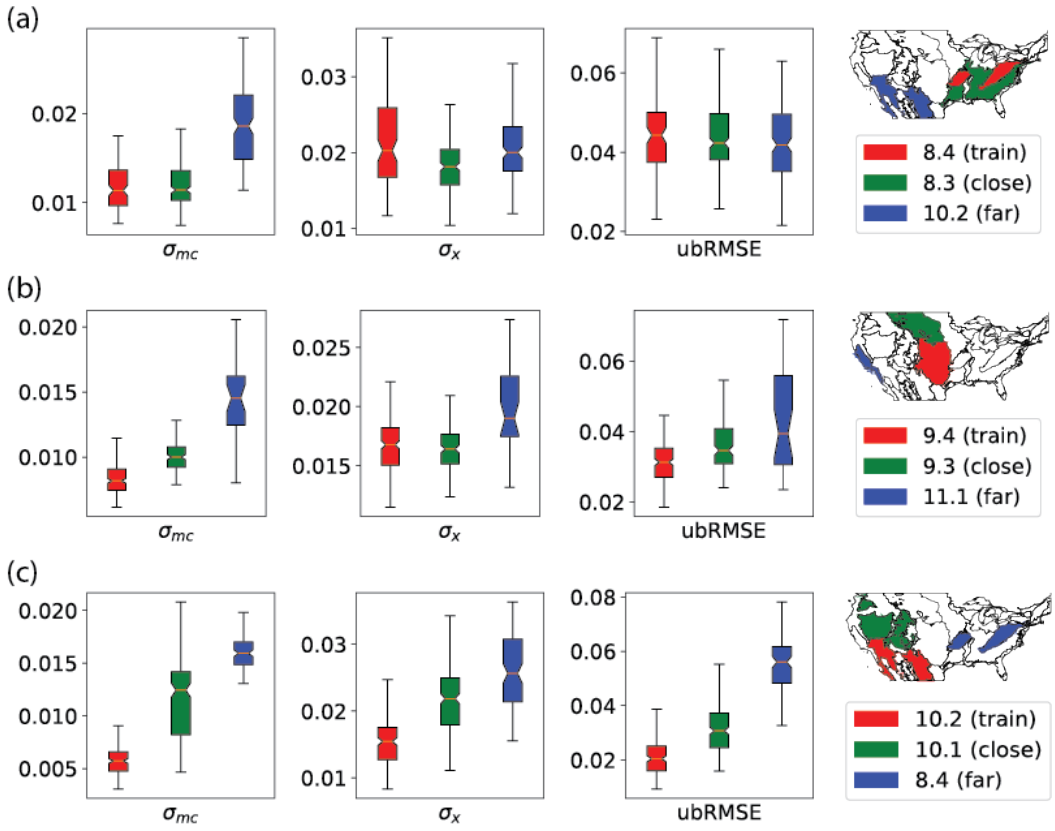


Figure 5. Metrics of performance when we trained the model in one level-2 ecoregion, and tested in two other level-2 ecoregions: one similar to the training region (from the same level-1 ecoregion), one farther away (from a different level-1 ecoregion). Performance metrics are network weight uncertainty (σ_{mc}), input-dependent data noise (σ_x), and test error ($ubRMSE$).

As σ_x is dependent on the training region, to further explore its limitations we investigated the performance of models when they were trained on several ecoregion combinations and tested on the rest of the CONUS. When the ecoregion combinations spanned across the CONUS, occupying a variety of landscapes in the CONUS domain (blue bars on Figure 6a), the estimated uncertainties were of higher quality. When the chosen ecoregions were clustered in only part of the CONUS domain (grouped as AB, CD, or EF, shown in Figure 6b), the estimated uncertainties were of much lower quality (higher d values). The combination EF had the lowest uncertainty quality, as these two regions are clustered together in the western arid landscape. Due to this aridity, the model trained there predicts small soil moisture fluctuations and also small σ_x when tested on other regions, resulting in significant under-estimation of the data noise term. We also noticed that whenever region F (warm deserts) was included in a combination in place of region E (cold deserts), the quality tended to be lower. This is presumably because the aridity of E is less extreme than F. As a result, including F instead of E expands the coverage of the training data in terms of the aridity scenarios.

This result can be explained by the fact that the data noise term was a trained output from the network, and was thus also conditioned by the training data. It provides direct evidence that σ_x could be misled by a strongly biased or unrepresentative training set. It is worth noting that the more representative sets (first three combinations) only sampled a fraction of the domain and are still far from representing the wide diversity of soil, land cover, and terrain combinations over the CONUS. However, they did provide more variety in the training data, and so it follows that σ_x reported by a model trained on one of these more varied datasets was more representative than σ_x reported by a model trained on a more biased training dataset.

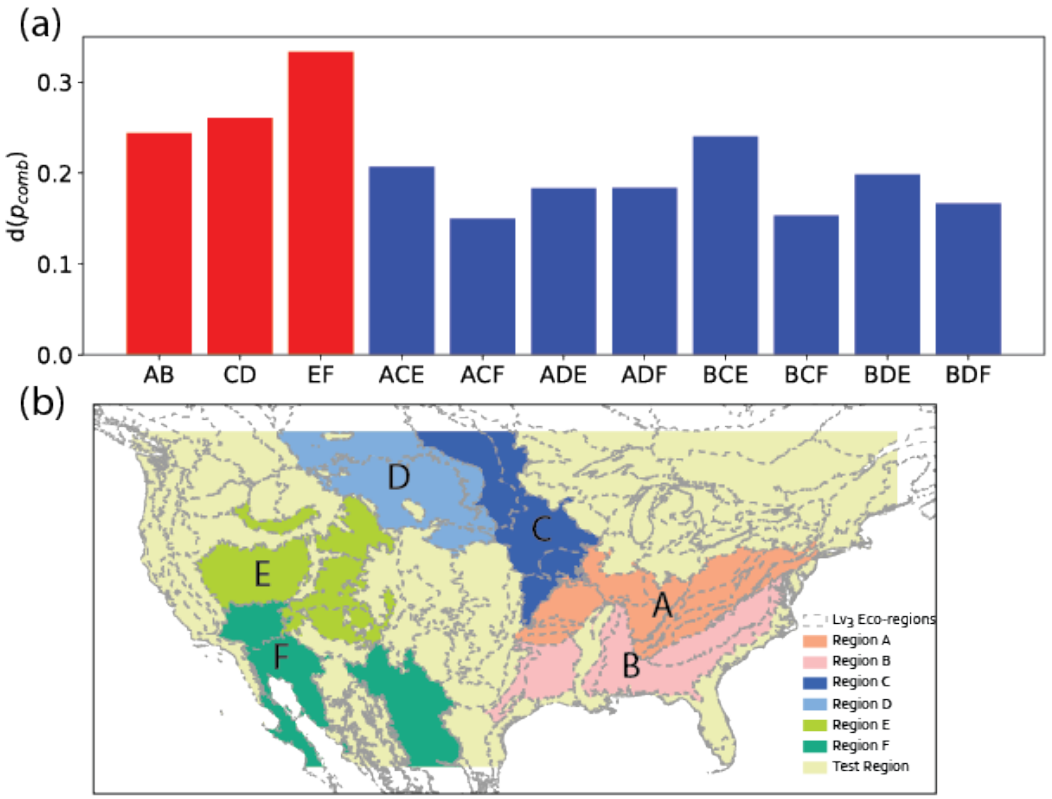


Figure 6. Evaluation of uncertainty quality (smaller d for higher quality) when models were trained on different combinations of ecoregions. The metrics were calculated in common regions of the CONUS that were outside of the training set. (a) Quality metric for combined error exceedance likelihoods ($d(p_{comb})$, the lower the better) of 11 combinations of regions, where 3 red bars show region combinations that are spatially clustered (AB, CD, EF) and 8 blue bars show region combinations that are spatially dispersed. Letters denote which regions are combined (e.g. ACE refers to a combination of regions A, C, and E). (b) Map of regions, some of which are composed of multiple level-3 ecoregions. A: ecoregions 8.3.1, 8.3.2, 8.3.3, 8.3.4, and 8.4; B: ecoregions 8.3.4, 8.3.5, 8.3.6, 8.3.7, and 8.3.8; C: ecoregion 9.2; D: ecoregion 9.3; E: ecoregions 10.1.4, 10.1.5, 10.1.6, 10.1.7, and 10.1.8; F: ecoregion 10.2.

3.4 Further discussion, limitations, and future work

The data noise term σ_x, which is essentially a trained, network-predicted error model, is shown to be a powerful technique with important implications for hydrology to simplify our workflow. Its quality and clear response to data noise suggest the plausibility of training such error models with very loose specifications of data noise. In the past,

a wealth of research has been dedicated to modeling error, e.g., specify error structures and adjustments for heteroscedasticity and autocorrelation (Evin et al., 2013; Götzinger & Bárdossy, 2008; Smith et al., 2015). The proposed procedure greatly relaxes the assumptions we need to make to obtain error models. The complex, possibly nonlinear, and potentially time-varying dependencies of the error on input terms can hardly be prescribed by experts. We can conveniently delegate such estimation to the deep learning algorithm itself, with the requirement that the training data must be representative.

The uncertainty with respect to climate or weather projections, a large and challenging research topic, has not been quantified here. For short-term forecast problems, the impacts of weather prediction error could potentially be assessed using weather forecasts from the past as atmospheric forcing data inputs to the model. As with other DL models, however, this work does not assume the forcings or the target observations to be perfect. The Artificial Intelligence community has worked extensively with data "*in the wild*", i.e. large but low-quality datasets, and DL models appear to deliver good performance even if there is significant noise (Izadinia et al., 2015; Stadelmann et al., 2018; Huang et al., 2016). What *will* mislead models are systematic errors.

The MCD+N method is simple to implement, but a lot remains to be understood. Although the two uncertainty terms were computed using very different methods and our experiments show they measure different uncertainty sources, their high level of correlation shows that they are not orthogonal, i.e. independent, quantities. Although perhaps unsatisfying, the correlation is consistent with their definitions and the proposed GP interpretation of network weight uncertainty (which was called the epistemic uncertainty in KG17). For data-driven models, knowledge comes from training data. When the training data has large amounts of noise, the knowledge of the model is negatively impacted, as reflected by the network weight uncertainty. In other words, noise in training data makes the model less certain of its own predictions. To further complicate our understanding, the correlation between network weight and data noise uncertainties also reflects the overall pattern of moisture variation and SMAP accuracy as functions of annual precipitation over the CONUS. Regions with high annual precipitation and high percentages of precipitation as snow also have high percentages of forest cover, and therefore high vegetation water content, which is known to lead to large uncertainty in SMAP measurements. Other datasets without these associations could help to disentangle the

effects of these factors. Even entangled, however, these factors are good estimators of prediction error and are thus still useful.

It could be hypothesized that the correlation between network weight and data noise uncertainties will be lower if we have a much larger dataset, as the data quantity could compensate for the quality, as shown in studies using noisy data "*in the wild*". However, as this is merely the first paper in hydrology to examine the MCD+N scheme, we leave the testing of this hypothesis to future work with more data quantity and diversity.

Due to its data-driven nature, the data noise uncertainty estimate is still conditioned by data, making it vulnerable to biased training data. This observation exposes an inherent limitation with any purely data-driven method, which is that it is difficult to assess the quality of data based only on the data itself. Future integration of knowledge or process-based models could potentially reduce this barrier. For example, process-based models could be constructed to introduce physics relationships that were not adequately represented in the training data. How to properly combine two classes of models is an active area of research (Karpatne et al., 2017; Shen et al., 2018), and other methods such as Stein variational gradient descent training (Liu & Wang, 2016; Mo et al., 2018) could also be considered.

MCD seemed to have automatically identified similarities in the inputs (atmospheric forcing data, soils, slope, land cover), which manifested as smaller network weight uncertainties for neighboring regions. These similarities are not entirely based on geographic proximity. Compared to geostatistical methods such as Kriging (a GP that parameterizes covariance functions over geographic distance), input-parameterized similarity facilitates physical interpretation and relieves us from the burden of identifying and tuning appropriate forms and parameters of covariance functions. An immediate next step could be to examine the most important physical input parameters that were employed by the MCD dissimilarity detector, to determine whether the network has made a physically-meaningful selection of attributes.

The theory behind the success of MCD needs further development, but this is one intuitive explanation for how it works: A deep network is composed of neurons. Each neuronal unit has inputs $x_1, \ldots, x_k$, corresponding weights $w_1, \ldots, w_k$, a bias term b, and an activation function $\mathbf{g}$. The output of the unit is $\mathbf{g}(b+\sum_i x_i w_i)$. During training with dropout, the neuron only uses a Bernoulli random sample of its inputs to create an out-

put, such that a random subset of the terms in the summation are removed. Thus the unit is conditioned to produce approximately the same output from different subsets of its input; otherwise training would not be stable. In other words, the neuronal unit learns about redundancies in its inputs that occur during training, and takes advantage of them so that different subsets of its inputs can produce approximately the same output. When the testing data are not represented by the training data, the characteristics of the inputs to the neuronal unit change. The same types of redundancies that held in the training data would not be expected to hold in the testing data. Hence, the random summations would no longer result in similar outputs, causing an observable increase in variability. Future work could test this intuition and further improve the MCD formulation. As a side note, this redundancy requirement would be a very powerful constraint, which could ensure that a trained neural system produces robust outcomes.

Uncertainty estimation has long been a focus in hydrology and other domains. However, very often the quality of the uncertainty estimate has not been thoroughly evaluated. Our results show that there could be many subtleties and limitations with state-of-the-art uncertainty estimates. For example, one could employ the MCD+N method for a model to produce an uncertainty estimate for a new instance, without realizing the limitations of the data noise term when this new instance is outside of the training data distribution. More importantly, an improper uncertainty estimate could provide a false sense of reliability. Therefore, we recommend carefully evaluating the uncertainty estimate before applying it in a production setting.

4 Conclusions

Uncertainty estimation is an essential task for hydrology, but it is new for hydrologic time series deep learning. Our evaluation with soil moisture predictions shows that MCD+N can indeed help to estimate model error. MCD+N proposed an input-dependent data noise term and a network weight uncertainty term, which are new concepts for hydrology. While the two terms were correlated for a CONUS-scale model, our experiments showed they indeed primarily targeted different uncertainty sources. The proposed data noise term is essentially a data-driven error model that greatly simplifies error quantification, without the need for explicit assumptions. Most observational noise was correctly attributed to the data noise term in our experiments. Additionally, our results provided the first strong supporting evidence that Monte Carlo dropout does act as a dissimilar-

ity detector, while the data noise term does not. These *work-as-intended* behaviors gives us some confidence that MCD+N is a useful tool. However, uncertainty estimation is not a replacement for data acquisition. We showed that both terms are dependent on the training data. If the training data are not representative, not only will the error increase noticeably, but the quality of the data noise estimate may also deteriorate. Fortunately, we only need a small set of data covering the input space to serve as a representative training set. To improve the uncertainty quality, we should strive to include extreme cases in the training set. The MCD+N scheme had promise, but should not be used with blind trust.

Acknowledgments

All data used in this study, including forcing data from NLDAS-2 [1], land surface characteristics (including soil texture from ISRIC-WISE [2], land cover from NLCD [3], and NDVI from [4]), and SMAP measurements, are available from public sources. The LSTM code can be openly downloaded from the open-source repository [5]. KF was sponsored partially by the Biological and Environmental Research program from the U.S. Department of Energy under contract DE-SC0016605. CS was supported by the National Science Foundation under grant EAR #1832294, and a seed grant from the Penn State Institutes of Energy and the Environment.

References

Ajami, N. K., Hornberger, G. M., & Sunding, D. L. (2008, nov). Sustainable water resource management under hydrological uncertainty. *Water Resources Research*, *44*(11). Retrieved from `http://doi.wiley.com/10.1029/2007WR006736` doi: 10.1029/2007WR006736

Batjes, N. H. (1995). *A Homogenized Soil Data File for Global Environmental Research: A Subset of FAO, ISRIC, and NRCS profiles (Version 1.0)* (Tech. Rep. No. No. 95/10b). Wageningen: ISRIC.

Behrouz, M., & Alimohammadi, S. (2018, aug). Uncertainty Analysis of

[1] `https://hydro1.gesdisc.eosdis.nasa.gov/data/NLDAS/NLDAS_FORA0125_H.002/`

[2] `https://www.isric.org/projects/world-inventory-soil-emission-potentials-wise`

[3] `https://www.mrlc.gov/data/nlcd-2016-land-cover-conus`

[4] `https://ecocast.arc.nasa.gov/data/pub/gimms/3g.v1/`

[5] `https://github.com/mhpi/hydroDL`

781 Flood Control Measures Including Epistemic and Aleatory Uncertainties:
782 Probability Theory and Evidence Theory. *Journal of Hydrologic Engi-*
783 *neering*, *23*(8), 04018033. Retrieved from http://ascelibrary.org/
784 doi/10.1061/{\%}28ASCE{\%}29HE.1943-5584.0001675 doi: 10.1061/
785 (ASCE)HE.1943-5584.0001675

786 Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and*
787 *Trends® in Machine Learning*, *2*(1), 1–127. Retrieved from http://www
788 .nowpublishers.com/article/Details/MAL-006 doi: 10.1561/2200000006

789 Beven, K. (1989, jan). Changing ideas in hydrology The case of physically-based
790 models. *Journal of Hydrology*, *105*(1-2), 157–172. Retrieved from http://
791 linkinghub.elsevier.com/retrieve/pii/0022169489901017 doi: 10.1016/
792 0022-1694(89)90101-7

793 Beven, K. (2016, jul). Facets of uncertainty: epistemic uncertainty, non-
794 stationarity, likelihood, hypothesis testing, and communication. *Hydro-*
795 *logical Sciences Journal*, *61*(9), 1652–1665. Retrieved from http://
796 www.tandfonline.com/doi/full/10.1080/02626667.2015.1031761 doi:
797 10.1080/02626667.2015.1031761

798 Butts, M. B., Payne, J. T., Kristensen, M., & Madsen, H. (2004, oct). An
799 evaluation of the impact of model structure on hydrological modelling
800 uncertainty for streamflow simulation. *Journal of Hydrology*, *298*(1-4),
801 242–266. Retrieved from https://www.sciencedirect.com/science/
802 article/pii/S0022169404002471?casa{_}token=XRb1PqQkAFMAAAAA:
803 ltPq9pDTvI7kRav1465tObC86fnWDkhYp53VC2kB-QbERZ5kAPpPaB6Uo3bM8J-X{\
804 _}awVEHdlAQ doi: 10.1016/J.JHYDROL.2004.03.042

805 De Lannoy, G. J. M., Reichle, R. H., Houser, P. R., Pauwels, V. R. N., & Ver-
806 hoest, N. E. C. (2007, sep). Correcting for forecast bias in soil moisture
807 assimilation with the ensemble Kalman filter. *Water Resources Research*,
808 *43*(9). Retrieved from http://doi.wiley.com/10.1029/2006WR005449 doi:
809 10.1029/2006WR005449

810 Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., & Udluft, S. (2017, oct).
811 Decomposition of Uncertainty in Bayesian Deep Learning for Efficient and
812 Risk-sensitive Learning. In *Proceedings of the 35 th international confer-*
813 *ence on machine learning, stockholm, sweden, pmlr 80, 2018.* Retrieved from

http://arxiv.org/abs/1710.07283

Evin, G., Kavetski, D., Thyer, M., & Kuczera, G. (2013, jul). Pitfalls and improvements in the joint inference of heteroscedasticity and autocorrelation in hydrological model calibration. *Water Resources Research*, *49*(7), 4518–4524. Retrieved from http://doi.wiley.com/10.1002/wrcr.20284 doi: 10.1002/wrcr.20284

Fang, K., Pan, M., & Shen, C. (2018). The Value of SMAP for Long-Term Soil Moisture Estimation With the Help of Deep Learning. *IEEE Transactions on Geoscience and Remote Sensing*, *PP*(Dl), 1–13. Retrieved from https://ieeexplore.ieee.org/document/8497052/ doi: 10.1109/TGRS.2018.2872131

Fang, K., & Shen, C. (2020, jan). Near-real-time forecast of satellite-based soil moisture using long short-term memory with an adaptive data integration kernel. *Journal of Hydrometeorology*, JHM–D–19–0169.1. Retrieved from http://journals.ametsoc.org/doi/10.1175/JHM-D-19-0169.1 doi: 10.1175/JHM-D-19-0169.1

Fang, K., Shen, C., Kifer, D., & Yang, X. (2017, nov). Prolongation of SMAP to Spatiotemporally Seamless Coverage of Continental U.S. Using a Deep Learning Neural Network. *Geophysical Research Letters*, *44*(21), 11,030–11,039. Retrieved from http://doi.wiley.com/10.1002/2017GL075619 doi: 10.1002/2017GL075619

Feng, D., Fang, K., & Shen, C. (2019). Enhancing streamflow forecast and extracting insights using long short term memory networks that assimilate recent observations. *https://arxiv.org/abs/1912.08949*.

Gal, Y., & Ghahramani, Z. (2016, jun). Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. *Proceedings of The 33rd International Conference on Machine Learning*, *48*. Retrieved from http://arxiv.org/abs/1506.02142

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. In G. Gordon, D. Dunson, & M. Dudík (Eds.), *Proceedings of the fourteenth international conference on artificial intelligence and statistics* (pp. 315–323). PMLR. Retrieved from http://proceedings.mlr.press/v15/glorot11a.html

Gong, W., Gupta, H. V., Yang, D., Sricharan, K., & Hero, A. O. (2013). Estimating epistemic and aleatory uncertainties during hydrologic modeling: An information theoretic approach. *Water Resources Research*, *49*(4), 2253–2273. Retrieved from https://agupubs.onlinelibrary.wiley.com/doi/full/10.1002/wrcr.20161 doi: 10.1002/wrcr.20161

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. The MIT Press. Retrieved from https://www.deeplearningbook.org/

Götzinger, J., & Bárdossy, A. (2008, dec). Generic error model for calibration and uncertainty estimation of hydrological models. *Water Resources Research*, *44*(12). Retrieved from http://doi.wiley.com/10.1029/2007WR006691 doi: 10.1029/2007WR006691

Graves, A. (2013, aug). Generating Sequences With Recurrent Neural Networks. *arXiv:1308.0850*. Retrieved from http://arxiv.org/abs/1308.0850

Graves, A., Mohamed, A.-r., & Hinton, G. (2013, may). Speech recognition with deep recurrent neural networks. In *2013 ieee international conference on acoustics, speech and signal processing* (pp. 6645–6649). Vancouver, Canada: IEEE. Retrieved from http://ieeexplore.ieee.org/document/6638947/ doi: 10.1109/ICASSP.2013.6638947

Hochreiter, S., & Schmidhuber, J. (1997, nov). Long Short-Term Memory. *Neural Computation*, *9*(8), 1735–1780. Retrieved from http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735 doi: 10.1162/neco.1997.9.8.1735

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/089360809190009T doi: 10.1016/0893-6080(91)90009-T

Huang, W., He, D., Yang, X., Zhou, Z., Kifer, D., & Giles, C. L. (2016). Detecting Arbitrary Oriented Text in the Wild with a Visual Attention Model. In *Proceedings of the 2016 acm on multimedia conference - mm '16* (pp. 551–555). New York, New York, USA: ACM Press. Retrieved from http://dl.acm.org/citation.cfm?doid=2964284.2967282 doi: 10.1145/2964284.2967282

Izadinia, H., Russell, B. C., Farhadi, A., Hoffman, M. D., & Hertzmann, A. (2015).

Deep Classifiers from Image Tags in the Wild. In *Proceedings of the 2015 workshop on community-organized multimodal mining: Opportunities for novel solutions* (pp. 13–18). ACM. doi: 10.1145/2814815.2814821

Karpatne, A., Atluri, G., Faghmous, J. H., Steinbach, M., Banerjee, A., Ganguly, A., ... Kumar, V. (2017, oct). Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering*, *29*(10), 2318–2331. Retrieved from `http://ieeexplore.ieee.org/document/7959606/` doi: 10.1109/TKDE.2017.2720168

Kavetski, D., Kuczera, G., & Franks, S. W. (2006, mar). Bayesian analysis of input uncertainty in hydrological modeling: 2. Application. *Water Resources Research*, *42*(3). Retrieved from `http://doi.wiley.com/10.1029/2005WR004376` doi: 10.1029/2005WR004376

Kendall, A., & Gal, Y. (2017, mar). What Uncertainties Do We Need in Bayesian Deep Learning for Computer Vision? *Advances in Neural Information Processing Systems 30*, *16*(4), 5574—-5584. Retrieved from `http://arxiv.org/abs/1703.04977`

Kiureghian, A. D., & Ditlevsen, O. (2009, mar). Aleatory or epistemic? Does it matter? *Structural Safety*, *31*(2), 105–112. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167473008000556` doi: 10.1016/J.STRUSAFE.2008.06.020

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, *22*(11), 6005–6022. Retrieved from `https://www.hydrol-earth-syst-sci.net/22/6005/2018/` doi: 10.5194/hess-22-6005-2018

Kumar, S., Sharma, A., & Tsunoda, T. (2019, dec). Brain wave classification using long short-term memory network based OPTICAL predictor. *Scientific Reports*, *9*(1). doi: 10.1038/s41598-019-45605-1

Lamontagne, J. R., Reed, P. M., Link, R., Calvin, K. V., Clarke, L. E., & Edmonds, J. A. (2018, mar). Large Ensemble Analytic Framework for Consequence-Driven Discovery of Climate Change Scenarios. *Earth's Future*, *6*(3), 488–504. Retrieved from `http://doi.wiley.com/10.1002/2017EF000701` doi: 10.1002/2017EF000701

LeCun, Y., Bengio, Y., & Hinton, G. (2015, may). Deep learning. *Nature*, *521*(7553), 436–444. Retrieved from http://www.nature.com/articles/nature14539 doi: 10.1038/nature14539

Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., & Sohl-Dickstein, J. (2018). Deep Neural Networks as Gaussian Processes. In *International conference on learning representations.* Retrieved from http://arxiv.org/abs/1711.00165

Liu, Q., & Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems* (pp. 2378–2386).

Matthews, A. G. d. G., Rowland, M., Hron, J., Turner, R. E., & Ghahramani, Z. (2018). Gaussian Process Behaviour in Wide Deep Neural Networks. In *International conference on learning representations* (pp. 1–36). Retrieved from http://arxiv.org/abs/1804.11271

McMahon, G., Gregonis, S. M., Waltman, S. W., Omernik, J. M., Thorson, T. D., Ffreehouf, J. A., ... Keys, J. E. (2001, apr). Developing a Spatial Framework of Common Ecological Regions for the Conterminous United States. *Environmental Management*, *28*(3), 293–316. Retrieved from http://link.springer.com/10.1007/s0026702429 doi: 10.1007/s0026702429

Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2017, may). Deep learning for healthcare: Review, opportunities and challenges. *Briefings in Bioinformatics*, *19*(6), 1236–1246. doi: 10.1093/bib/bbx044

Mo, S., Zhu, Y., Zabaras, J., Nicholas, Shi, X., & Wu, J. (2018). Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. *Water Resources Research*.

Neal, R. M. (1996). Priors for Infinite Networks. In *Bayesian learning for neural networks* (pp. 29–53). New York, NY: Springer New York. Retrieved from http://link.springer.com/10.1007/978-1-4612-0745-0_2 doi: 10.1007/978-1-4612-0745-0_2

Nearing, G. S., Mocko, D. M., Peters-Lidard, C. D., Kumar, S. V., & Xia, Y. (2016, mar). Benchmarking NLDAS-2 Soil Moisture and Evapotranspiration to Separate Uncertainty Contributions. *Journal of Hydrometeorology*, *17*(3), 745–759. Retrieved from http://journals.ametsoc.org/doi/10.1175/

946 JHM-D-15-0063.1 doi: 10.1175/JHM-D-15-0063.1

947 Nearing, G. S., Tian, Y., Gupta, H. V., Clark, M. P., Harrison, K. W., & Weijs,
948 S. V. (2016, jul). A philosophical basis for hydrological uncertainty. *Hydrolog-*
949 *ical Sciences Journal*, *61*(9), 1666–1678. Retrieved from http://dx.doi.org/
950 10.1080/02626667.2016.1183009 doi: 10.1080/02626667.2016.1183009

951 O'Neill, P., Chan, S., Njoku, E., Jackson, T., & Bindlish, R. (2016). *SMAP L3*
952 *Radiometer Global Daily 36 km EASE-Grid Soil Moisture, Version 4.* Boul-
953 der, Colorado USA: NASA National Snow and Ice Data Center Distributed
954 Active Archive Center. Retrieved from https://nsidc.org/data/SPL3SMP/
955 versions/4 doi: 10.5067/OBBHQ5W22HME

956 Osband, I., Blundell, C., Pritzel, A., & Van Roy, B. (2016, feb). Deep Exploration
957 via Bootstrapped DQN. *NIPS 2016 Bayesian Deep Learning Workshop*, 26–28.
958 Retrieved from http://arxiv.org/abs/1602.04621

959 Pan, M., Cai, X., Chaney, N. W., Entekhabi, D., & Wood, E. F. (2016, sep). An
960 initial assessment of SMAP soil moisture retrievals using high-resolution model
961 simulations and in situ observations. *Geophysical Research Letters*, *43*(18),
962 9662–9668. Retrieved from http://doi.wiley.com/10.1002/2016GL069964
963 doi: 10.1002/2016GL069964

964 Pappenberger, F., & Beven, K. J. (2006, may). Ignorance is bliss: Or seven reasons
965 not to use uncertainty analysis. *Water Resources Research*, *42*(5), 1–8. Re-
966 trieved from http://doi.wiley.com/10.1029/2005WR004820 doi: 10.1029/
967 2005WR004820

968 Rasmussen, C. E., & Williams, C. K. I. (2005). *Gaussian Processes for Machine*
969 *Learning (Adaptive Computation and Machine Learning Series).* The MIT
970 Press.

971 Schmidhuber, J. (2015, jan). Deep learning in neural networks: An overview. *Neu-*
972 *ral Networks*, *61*(10), 85–117. Retrieved from https://linkinghub.elsevier
973 .com/retrieve/pii/S0893608014002135 doi: 10.1016/j.neunet.2014.09.003

974 Senge, R., Bösner, S., Dembczyński, K., Haasenritter, J., Hirsch, O., Donner-
975 Banzhoff, N., & Hüllermeier, E. (2014, jan). Reliable classification: Learning
976 classifiers that distinguish aleatoric and epistemic uncertainty. *Information*
977 *Sciences*, *255*, 16–29. Retrieved from https://www.sciencedirect.com/
978 science/article/pii/S0020025513005410 doi: 10.1016/J.INS.2013.07.030

Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F.-j., ... Tsai, W.-P. (2018, nov). HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community. *Hydrology and Earth System Sciences*, *22*(11), 5639–5656. Retrieved from https://www.hydrol-earth-syst-sci.net/22/5639/2018/ doi: 10.5194/hess-22-5639-2018

Smith, T., Marshall, L., & Sharma, A. (2015, sep). Modeling residual hydrologic errors with Bayesian inference. *Journal of Hydrology*, *528*, 29–37. Retrieved from https://www.sciencedirect.com/science/article/pii/S0022169415004011?casa{_}token=qzNGTPKWZooAAAAA:jyFlvInezUh78OkFBSn-7tVwF9PdX2EErKiKTOlpjEKrKyfBIEhPqcQdge1Tzb3gC7tJz0uZJQ doi: 10.1016/J.JHYDROL.2015.05.051

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html doi: 10.1214/12-AOS1000

Stadelmann, T., Amirian, M., Arabaci, I., Arnold, M., Duivesteijn, G. F., Elezi, I., ... Tuggener, L. (2018, sep). Deep Learning in the Wild. In *Annpr 2018* (pp. 17–38). Springer, Cham. Retrieved from http://link.springer.com/10.1007/978-3-319-99978-4{_}2 doi: 10.1007/978-3-319-99978-4_2

Vandal, T., Kodra, E., Dy, J., Ganguly, S., Nemani, R., & Ganguly, A. R. (2018). Quantifying Uncertainty in Discrete-Continuous and Skewed Data with Bayesian Deep Learning. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining - kdd '18* (pp. 2377–2386). New York, New York, USA: ACM Press. Retrieved from http://dx.doi.org/10.1145/3219819.3219996 doi: 10.1145/3219819.3219996

Vrugt, J. A., ter Braak, C. J. F., Clark, M. P., Hyman, J. M., & Robinson, B. A. (2008, dec). Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation. *Water Resources Research*, *44*(12), 1–15. Retrieved from http://doi.wiley.com/10.1029/2007WR006720 doi: 10.1029/2007WR006720

Wilson, T., Tan, P.-n., & Luo, L. (2018, nov). A Low Rank Weighted Graph Con-

volutional Approach to Weather Prediction. In *2018 ieee international conference on data mining (icdm)* (pp. 627–636). IEEE. Retrieved from https://ieeexplore.ieee.org/document/8594887/ doi: 10.1109/ICDM.2018.00078

Xia, Y., Ek, M. B., Wu, Y., Ford, T., & Quiring, S. M. (2015, oct). Comparison of NLDAS-2 Simulated and NASMD Observed Daily Soil Moisture. Part I: Comparison and Analysis. *Journal of Hydrometeorology*, *16*(5), 1962–1980. Retrieved from http://journals.ametsoc.org/doi/10.1175/JHM-D-14-0096.1 doi: 10.1175/JHM-D-14-0096.1

Zhang, D., Lindholm, G., & Ratnaweera, H. (2018, jan). Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring. *Journal of Hydrology*, *556*, 409–418. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0022169417307722 doi: 10.1016/j.jhydrol.2017.11.018

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018, jun). Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, *561*(April), 918–929. Retrieved from https://linkinghub.elsevier.com/retrieve/pii/S0022169418303184 doi: 10.1016/j.jhydrol.2018.04.065

A The MCD theory and its potential issues

The derivations from GG16 (Gal & Ghahramani, 2016) are quite lengthy, so here we only highlight a few main steps. The prototype network analyzed is a two-layer network written as $\mathbf{f} = \sigma(\mathbf{x}\mathbf{W^{(1)}} + \mathbf{b})\mathbf{W^{(2)}}$, where σ is a nonlinear activation function such as TanH or ReLU and $\mathbf{W^{(1)}}$ and $\mathbf{W^{(2)}}$ are the weights for the first and second layers, respectively. Adding in dropout operators, we obtain $\mathbf{f} = (\sigma(\mathbf{x}(\mathbf{z^{(1)}}\mathbf{W^{(1)}}) + \mathbf{b})(z^{(2)}\mathbf{W^{(2)}})$, where $z^{(1)} \sim \text{Bernoulli}(\beta^{(1)})$ and $z^{(2)} \sim \text{Bernoulli}(\beta^{(2)})$ are dropout masks of the same sizes as $\mathbf{W^{(1)}}$ and $\mathbf{W^{(2)}}$, respectively. $\beta^{(k)}$ is the the probability that a connection on the k-th layer is retained during dropout, or one minus the "dropout rate" in many DL packages. Hence we refer to it as the dropout retention rate.

In a standard Bayesian inference framework, we (i) start with a prior distribution of model parameters, e.g. $p(\mathbf{W}) = \mathcal{N}(0, I)$; (ii) confront the model with the data (evaluating the likelihood function) and calculate the posterior distribution of the parameter sets using Bayes law (i.e. given the training dataset $(\mathbf{X}, \mathbf{Y})$, $p(\mathbf{W}|\mathbf{X}, \mathbf{Y})$ =

$p(\mathbf{Y}|\mathbf{W},\mathbf{X})p(\mathbf{W})/p(\mathbf{Y}|\mathbf{X})$); and (iii) use the posterior distribution to make predictions as well as estimate predictive uncertainty for new test instances X^*:

$$p(\mathbf{Y}^*|\mathbf{X}^*) = \int \mathbf{p}(\mathbf{Y}^*|\mathbf{X}^*,\mathbf{W})\mathbf{p}(\mathbf{W}|\mathbf{X},\mathbf{Y})dW \tag{A.1}$$

The posterior distribution $p(\mathbf{W}|\mathbf{X},\mathbf{Y})$ is the distribution that most likely generated the observed data. However, this distribution cannot be easily estimated as the marginal distribution $p(\mathbf{Y}|\mathbf{X})$ cannot be evaluated analytically, and is intractable for very high-dimensional deep networks. A viable approach is to replace this distribution with a *variational* distribution $q(W)$, whose structure is easier to work with in the integral. Variational inference turns the inference problem into an optimization problem, where we minimize the Kullback-Leibler divergence between the variational distribution and the posterior distribution, $\mathbf{KL}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{X},\mathbf{Y})))$, which measures the dissimilarity between distributions. Typically, this task is further turned into the problem of maximizing the *log evidence lower bound* (LELB)

$$\mathcal{L} = \int q(W)\log p(\mathbf{Y}|\mathbf{X},\mathbf{W})d\omega - \mathbf{KL}(q(\mathbf{W})||p(\mathbf{W}|\mathbf{X},\mathbf{Y})) \tag{A.2}$$

This procedures optimizes both the weights of the neural network and the variational parameters. As a result, after we solve this minimization problem we will have obtained both a functional neural network and a variational distribution that can be easily sampled from. In the case of GG16, the authors would like to prove that dropout training corresponds to *some* form of variational distribution. They defined their variational distributions for the weights of layer 1, $\mathbf{W^{(1)}}$, as a Gaussian mixture which can be factorized over each row vector:

$$q(\mathbf{W^{(1)}}) = \prod_{q=1}^{Q} q(\mathbf{w_q}) \tag{A.3}$$

$$q(\mathbf{w_q}) = \beta^{(1)}\mathcal{N}(\mathbf{m_q},\sigma^{\mathbf{2}}\mathbf{I_K}) + (1-\beta^{(1)})\mathcal{N}(0,\sigma^{\mathbf{2}}\mathbf{I_K}) \tag{A.4}$$

where $\mathbf{W^{(1)}}$ is of the size $Q\times K$ and $\mathbf{w_q}$ is a row vector in $\mathbf{W^{(1)}}$. Similar distributions were put on $\mathbf{W^{(2)}}$. This variational distribution can further be re-parameterized as the following

$$\mathbf{W^{(1)}} = z^{(1)}(M^{(1)} + \sigma\epsilon^{(1)}) + (1 - z^{(1)})\sigma\epsilon^{(1)} \tag{A.5}$$

$$\mathbf{W^{(2)}} = z^{(2)}(M^{(2)} + \sigma\epsilon^{(2)}) + (1 - z^{(2)})\sigma\epsilon^{(2)} \tag{A.6}$$

$$\mathbf{b} = \mathbf{m} + \sigma\epsilon \tag{A.7}$$

The parameterization allows the integral in Eq. A.2 to be estimated using Monte Carlo integration, i.e.,

$$\mathcal{L}_{GP-MC} = \sum_{m=1}^{M} \log p(\mathbf{y_m}|\mathbf{x_m}, \widehat{\mathbf{W}}_{\mathbf{m}}^{(\mathbf{1})}, \widehat{\mathbf{W}}_{\mathbf{m}}^{(\mathbf{2})}, \widehat{\mathbf{b}}_{\mathbf{m}}) - KL(q(\mathbf{W^{(1)}}, \mathbf{W^{(2)}}, \mathbf{b})||p(\mathbf{W^{(1)}}, \mathbf{W^{(2)}}, \mathbf{b})) \tag{A.8}$$

where $\widehat{W}_n^{(1)}, \widehat{W}_n^{(2)}$, and $\widehat{b}_n$ are the weights for the n-th realization. GG16 argued that when σ is small, we simply have $\widehat{\mathbf{W}}^{(\mathbf{1})} \approx \widehat{\mathbf{z}}_{\mathbf{n}}^{(\mathbf{1})}\mathbf{M}^{(\mathbf{1})}$, $\widehat{\mathbf{W}}^{(\mathbf{2})} \approx \widehat{\mathbf{z}}_{\mathbf{n}}^{(\mathbf{1})}\mathbf{M}^{(\mathbf{1})}$, $\widehat{\mathbf{b}}^{\mathbf{n}} \approx \mathbf{m}$. In other words, applying a stochastic dropout mask on the weights is approximately drawing a sample from the variational distribution in Eq. A.7, and the summation term simply amounts to the sum of squared loss for training with dropout and mini-batching. Some other approximations that take advantage of the large size of deep networks were further employed to handle the KL term. Furthermore, by stacking more layers, the same derivation was extended to multi-layer networks.

While it is fortunate that such an interpretation for dropout could exist, there were many approximate steps in this derivation. In particular, we have the following concerns: (i) the Bernoulli distribution and the Gaussian mixture that it approximates might not be competent enough as a variational distribution. The Gaussian mixture itself, as shown in the derivation, must have small variances, and it is uncertain if such strong limitations are valid for Bayesian inference; (ii) the Gaussian prior over the parameters $W \sim \mathcal{N}(1, I)$ is coincidental but not necessarily optimal; (iii) with many approximations stacked up in the derivation, it is dubious if the conclusion still converges to the declared final outcome; and (iv) the derivation was only demonstrated for simple multi-layer neural networks. This derivation has yet to be shown to work for complex recurrent networks like LSTM. It is not certain if LSTM with dropout training is a deep GP. While these concerns are difficult to address analytically at the moment, we can experimentally verify the effectiveness of MCD and answer the research questions presented at the end of the Introduction section.

B Calibration of dropout rate

Here we examine the role that dropout retention rate (β) plays in the uncertainty estimate terms and the predictive error. In the MCD theory, the variational distribution for the parameters are Gaussian mixtures with very small variances, and the weights before them are from a Bernoulli distribution (Appendix A). The dropout rate ($dr = 1-\beta$) should be carefully calibrated. We trained the model from 2015/04 to 2016/03 using $\beta \in \{0.1, 0.2, ..., 0.9\}$. The best β was chosen based on both the error and quality of the uncertainty estimate in the validation set (2016/04 - 2017/03). As figure B.1 shows, both $ubRMSE$ and σ_{comb} are affected by the dropout rate. We chose the model trained with $dr = 0.6$, or $\beta = 0.4$, as it simultaneously gave the smallest $ubRMSE$ and the best uncertainty quality, as measured by d, the Kolmogorov-Smirnov statistic (maximum distance) between the CDF of the error exceedance likelihoods and the one-to-one line.

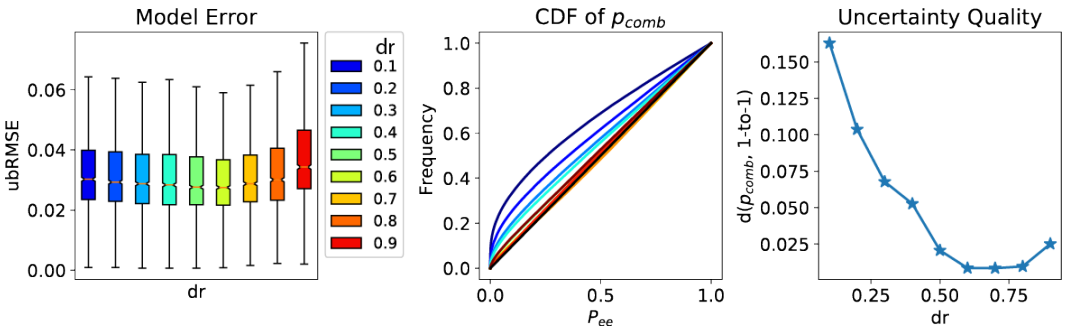


Figure B.1. Performance of uncertainty models with different dropout rates ($dr = 1-\beta$). (a) ubRMSE as a function of dr. (b) The CDF curves of the error exceedance likelihoods. (c) The Kolmogorov-Smirnov statistic as a function of dr. We found that $dr = 0.6$ offers a balance of small d as well as small ubRMSE.

C Test on hydrologic basins instead of ecoregions.

In practice, hydrologic models are commonly developed based on basins instead of ecoregions. Hence, to provide more insights, we trained models on each of the 18 2-digits hydrologic cataloging unit (HUC02) basins dividing CONUS. Similar to the ecoregion experiments, the models were trained over year 2015, validated over 2016 and tested over 2017. We reproduced the figure 4 and 5 as C.1 and C.2 correspondingly, and they revealed similar pattern as we discussed in section 3.3.

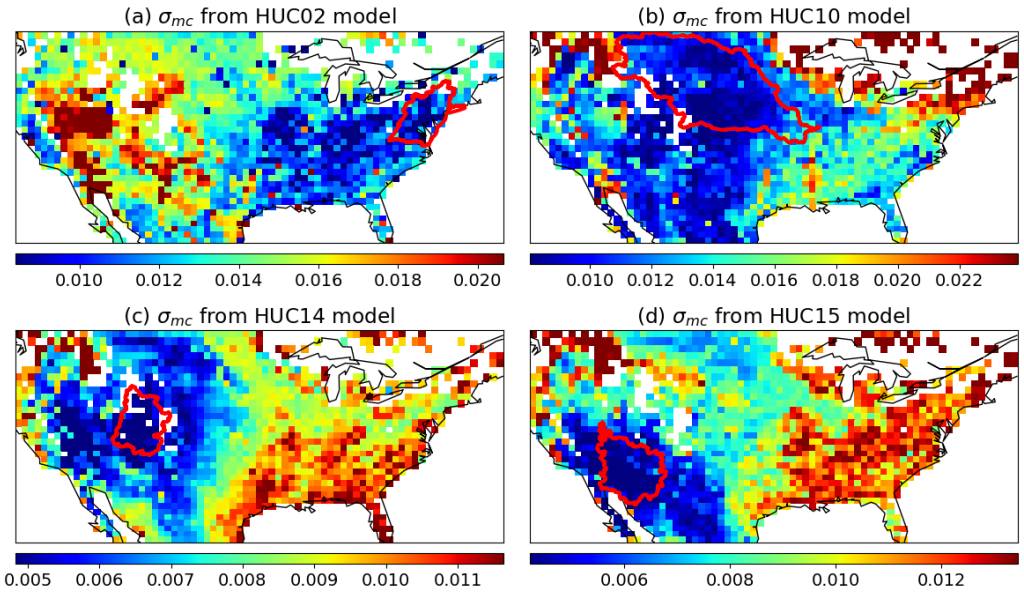


Figure C.1. Maps of σ_{mc} when the LSTM model is trained in one of the HUC2 basins. The training region is highlighted by the red polygon.

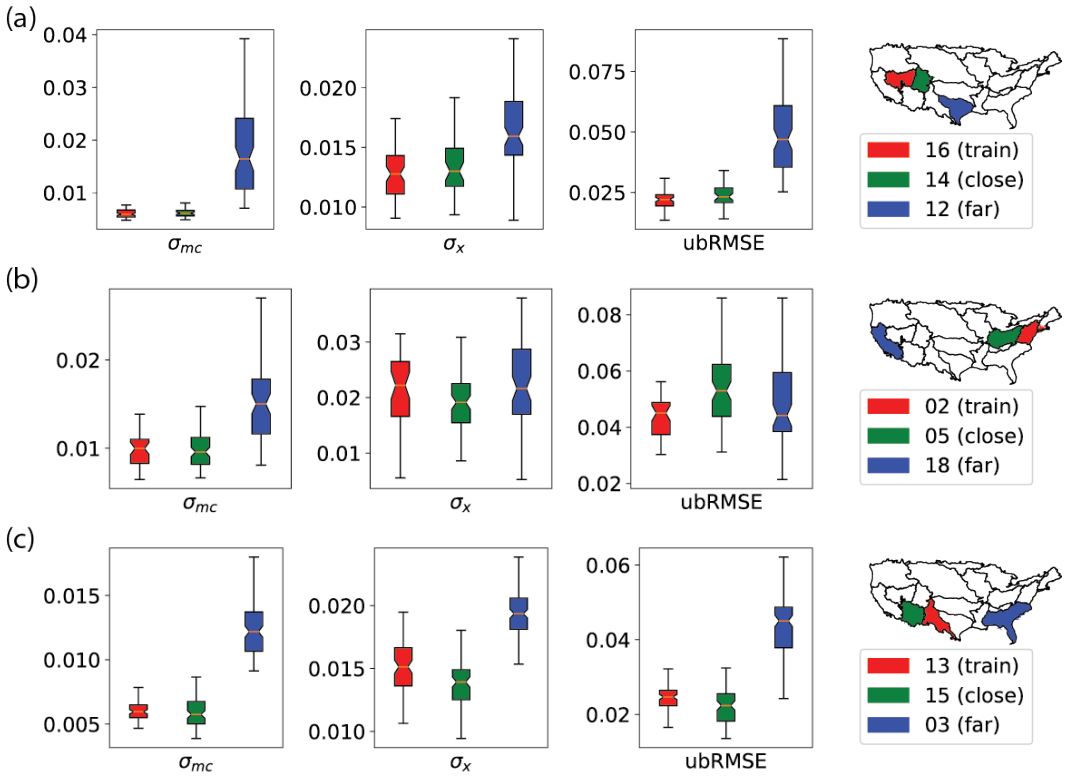
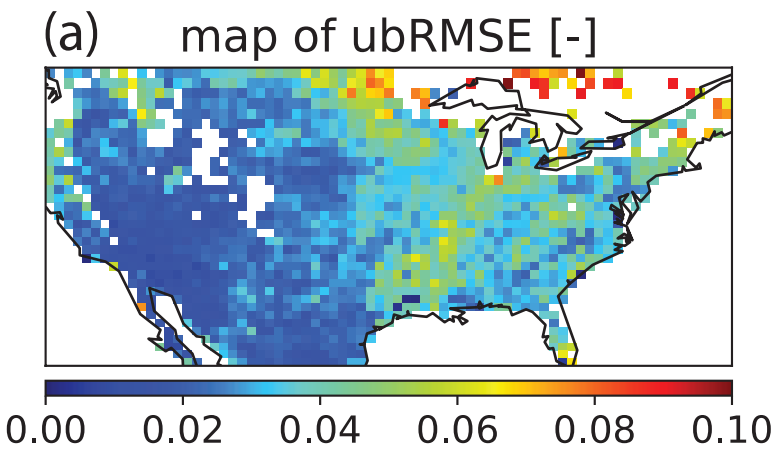
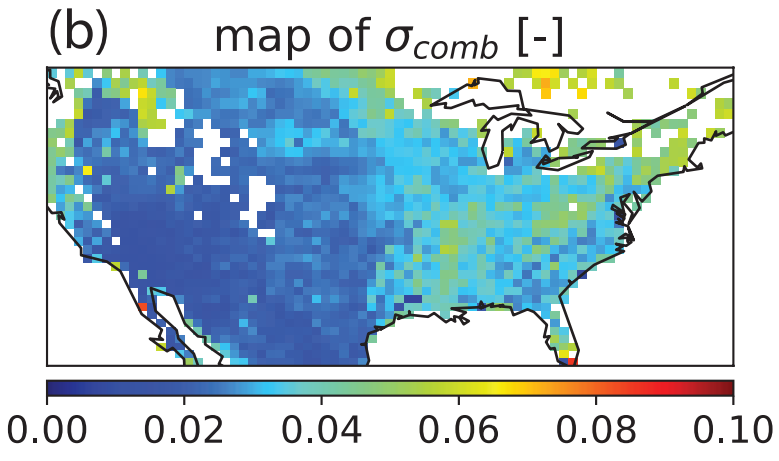


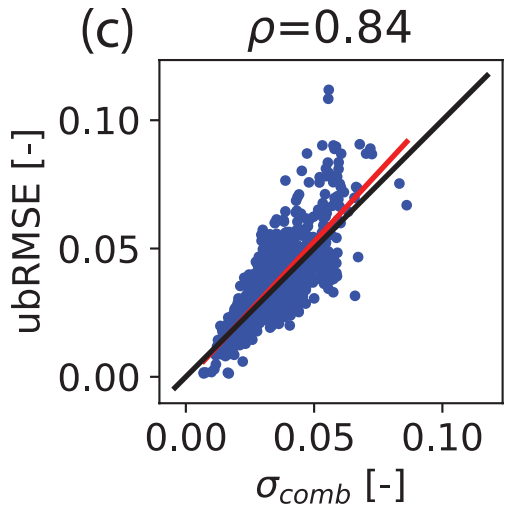
Figure C.2. Metrics of performance when we trained the model in a HUC2, and tested in two other HUC2s: one similar to the training region, one farther away, in a different physiographic region.

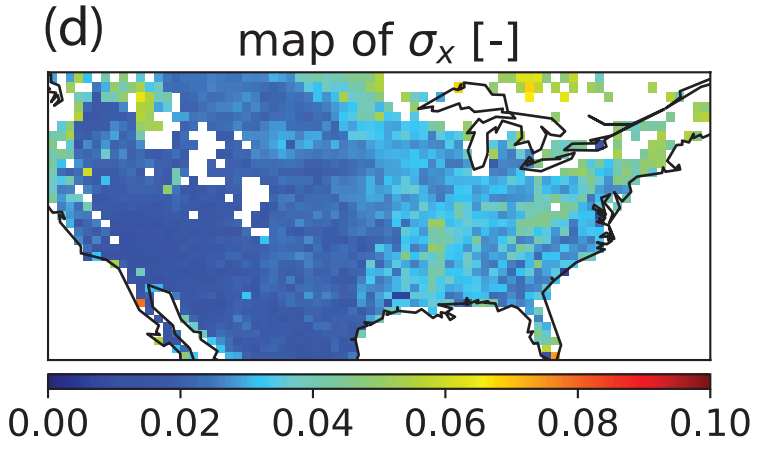
Figure 1.

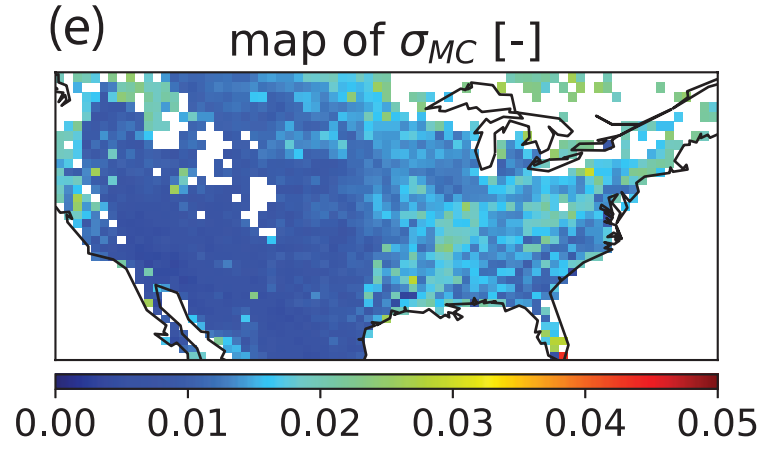
Temporal Test

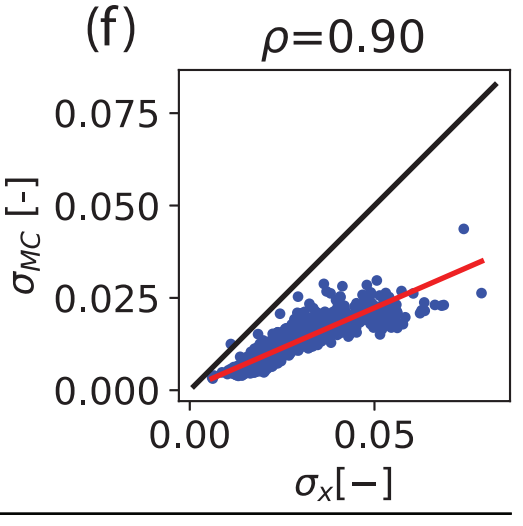




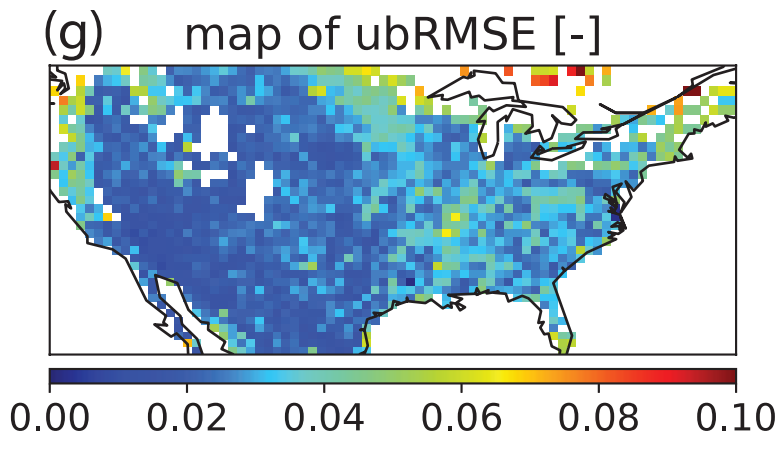


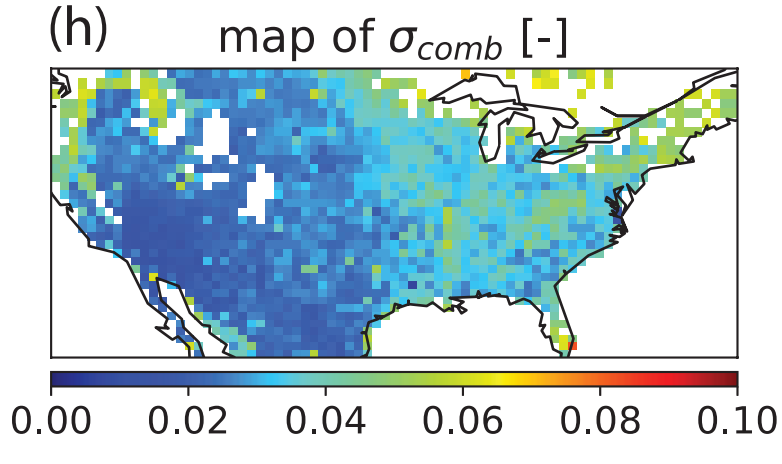


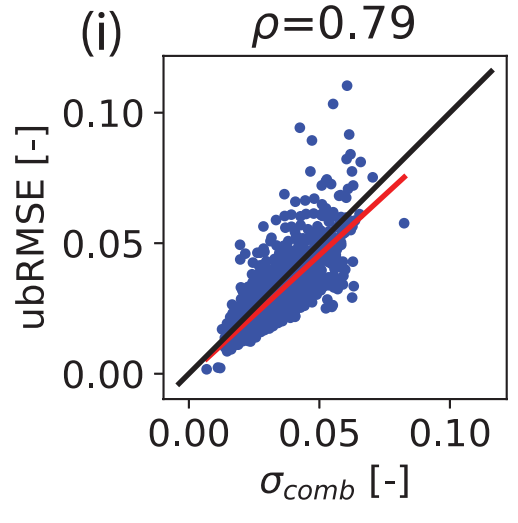


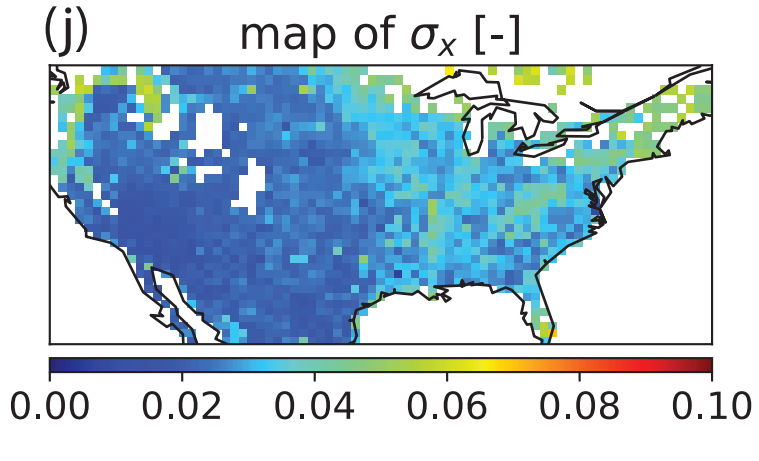


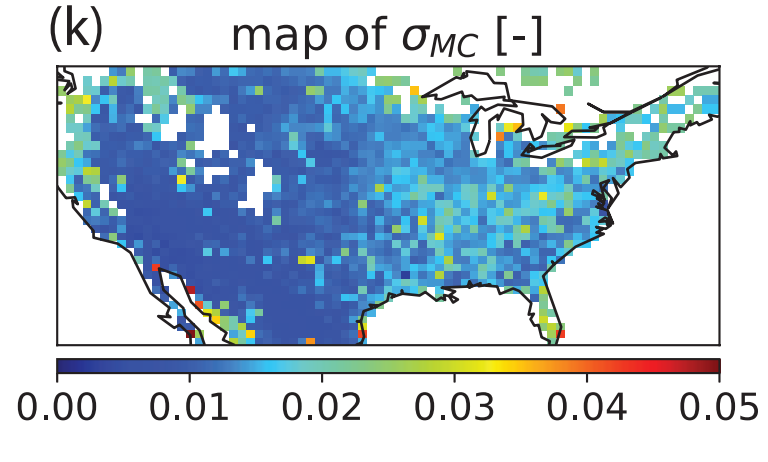
Spatial Test











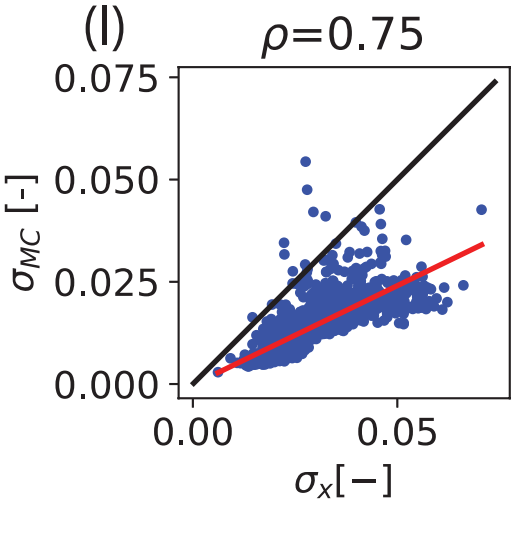
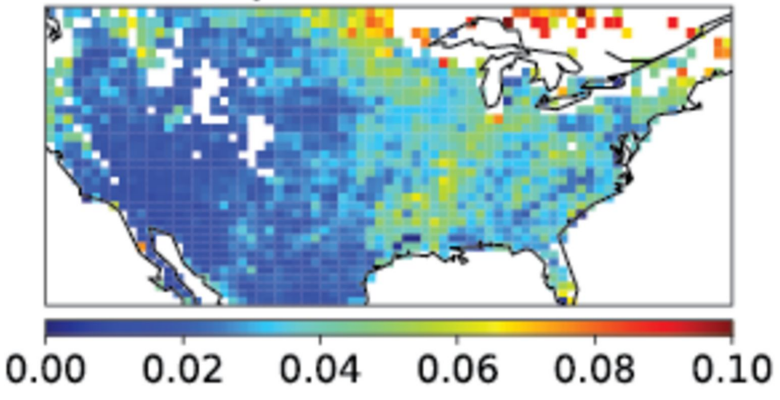


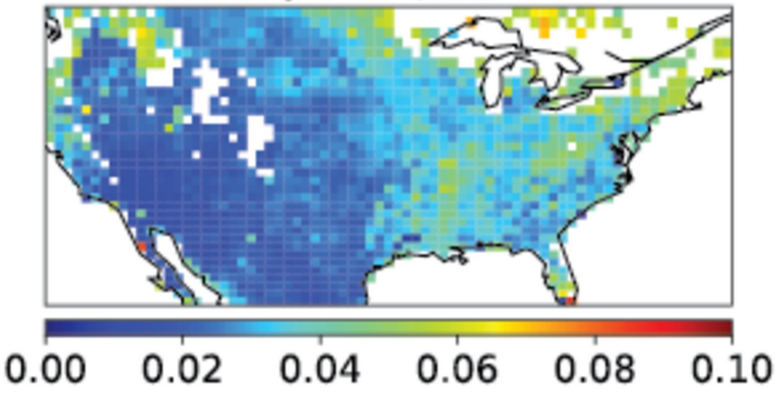
Figure 1 png ver.

Temporal Test

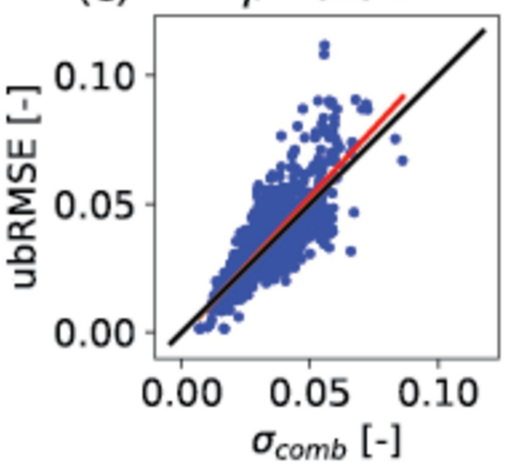
(a) map of ubRMSE [-]



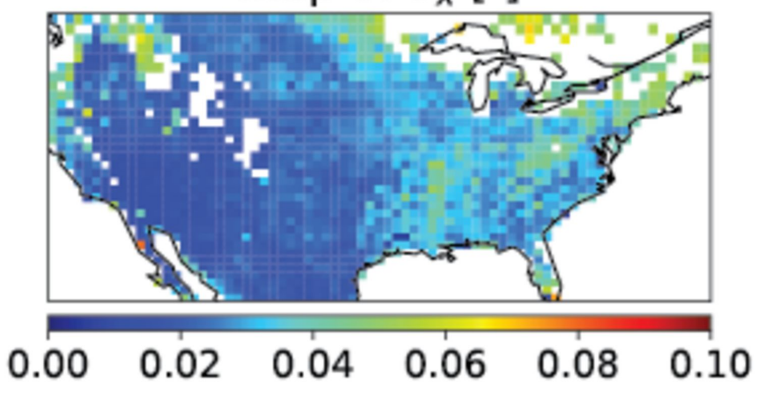
(b) map of σ_{comb} [-]



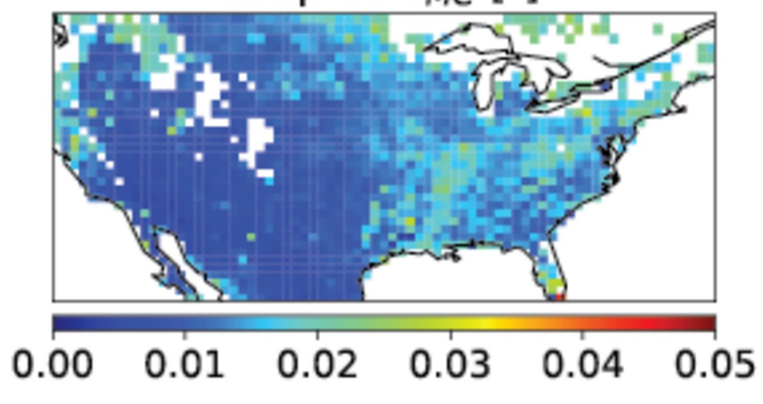
(c) ρ=0.84



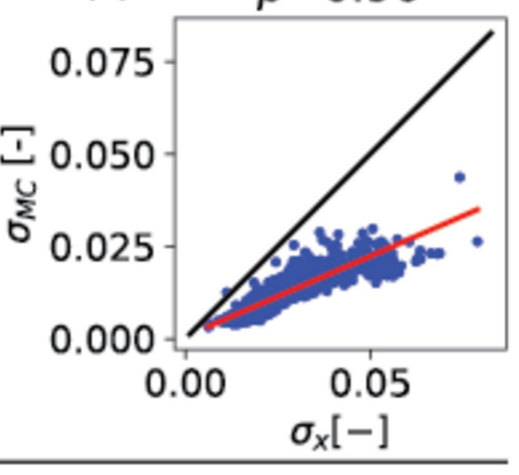
(d) map of σ_x [-]



(e) map of σ_{MC} [-]

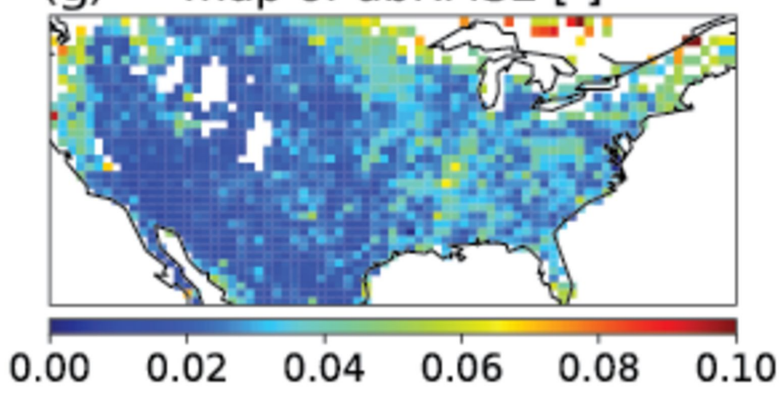


(f) ρ=0.90

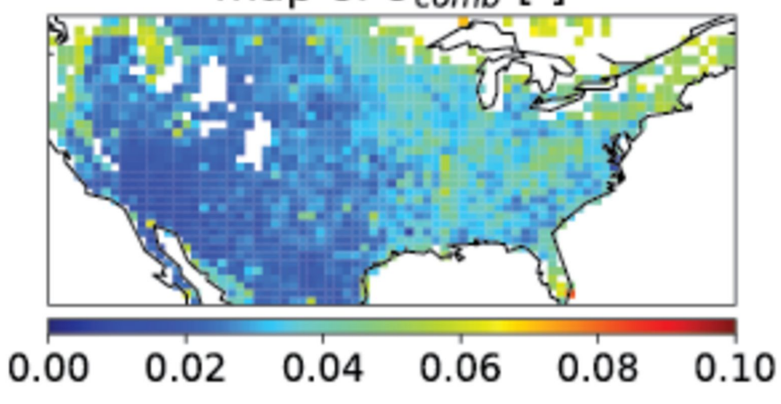


Spatial Test

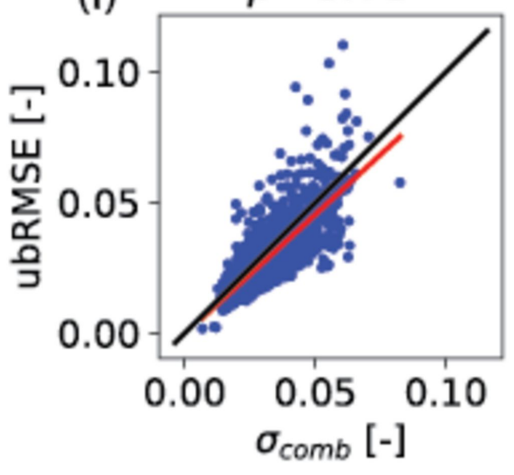
(g) map of ubRMSE [-]



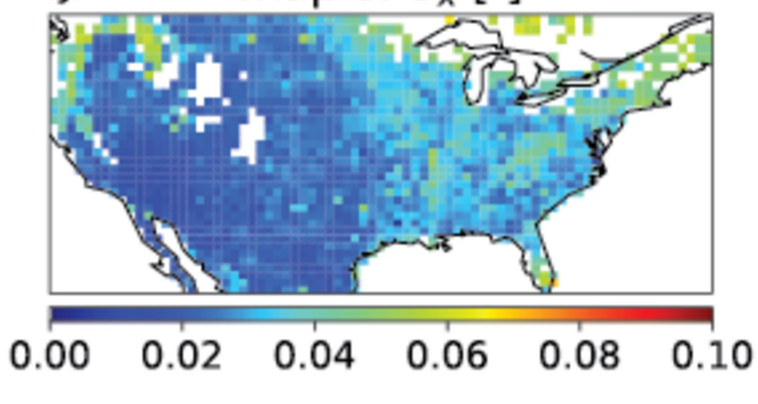
(h) map of σ_{comb} [-]



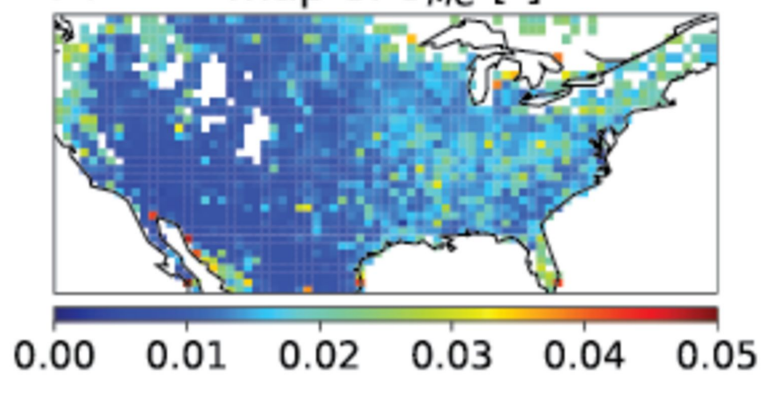
(i) ρ=0.79



(j) map of σ_x [-]



(k) map of σ_{MC} [-]



(l) ρ=0.75

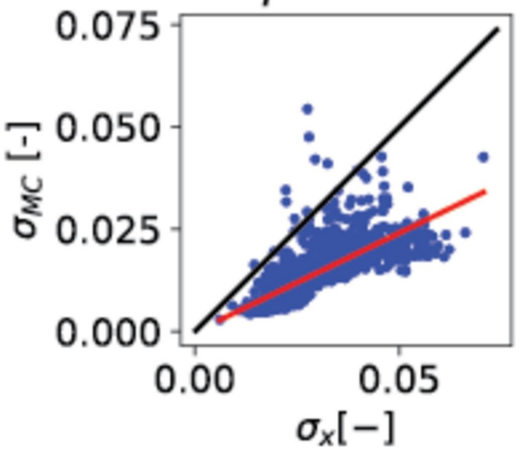


Figure 2.

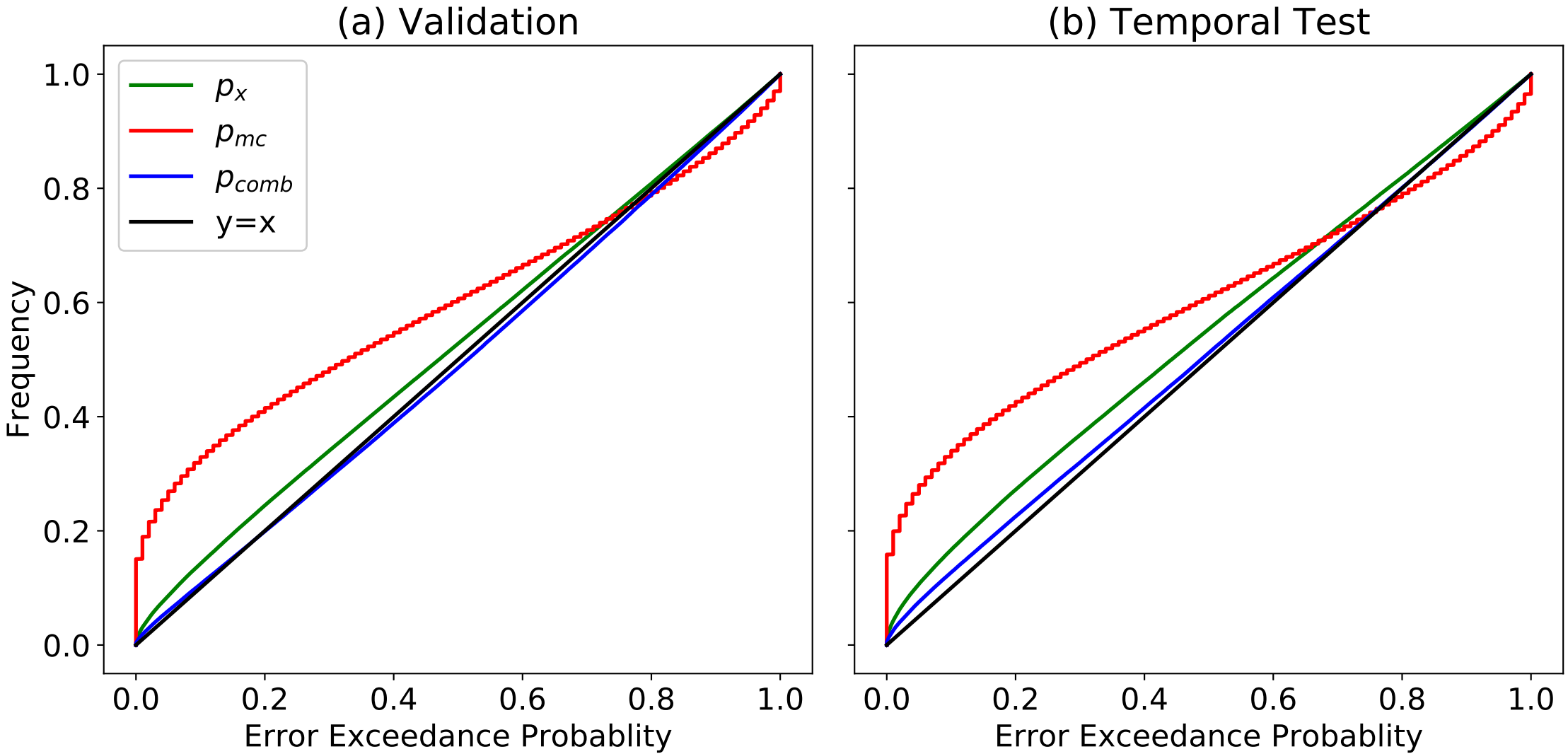

(a) Validation
(b) Temporal Test
p_x
p_{mc}
p_{comb}
y=x
Frequency
Error Exceedance Probablity
Error Exceedance Probablity
0.0
0.2
0.4
0.6
0.8
1.0

Figure 2 png ver.

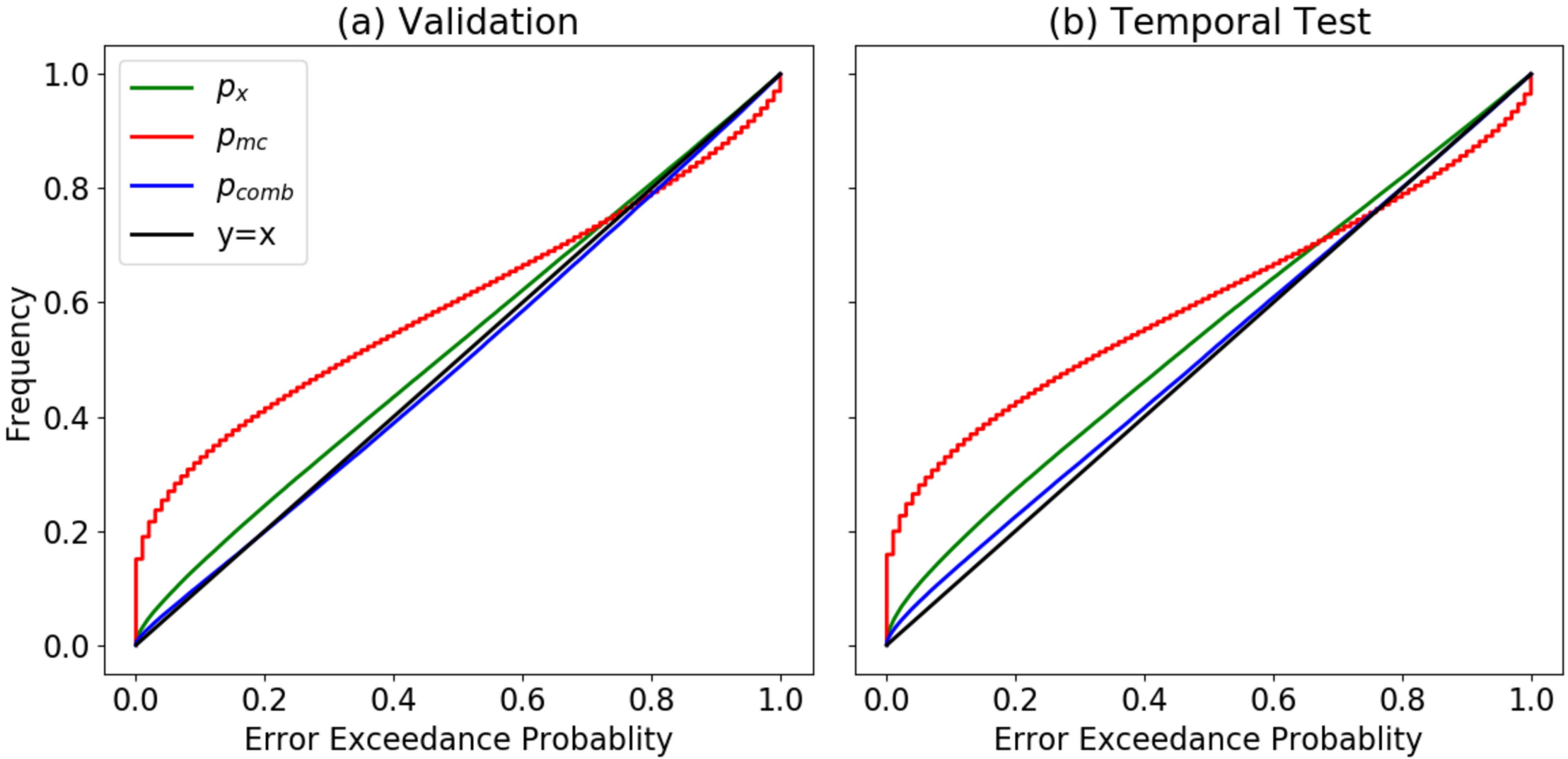
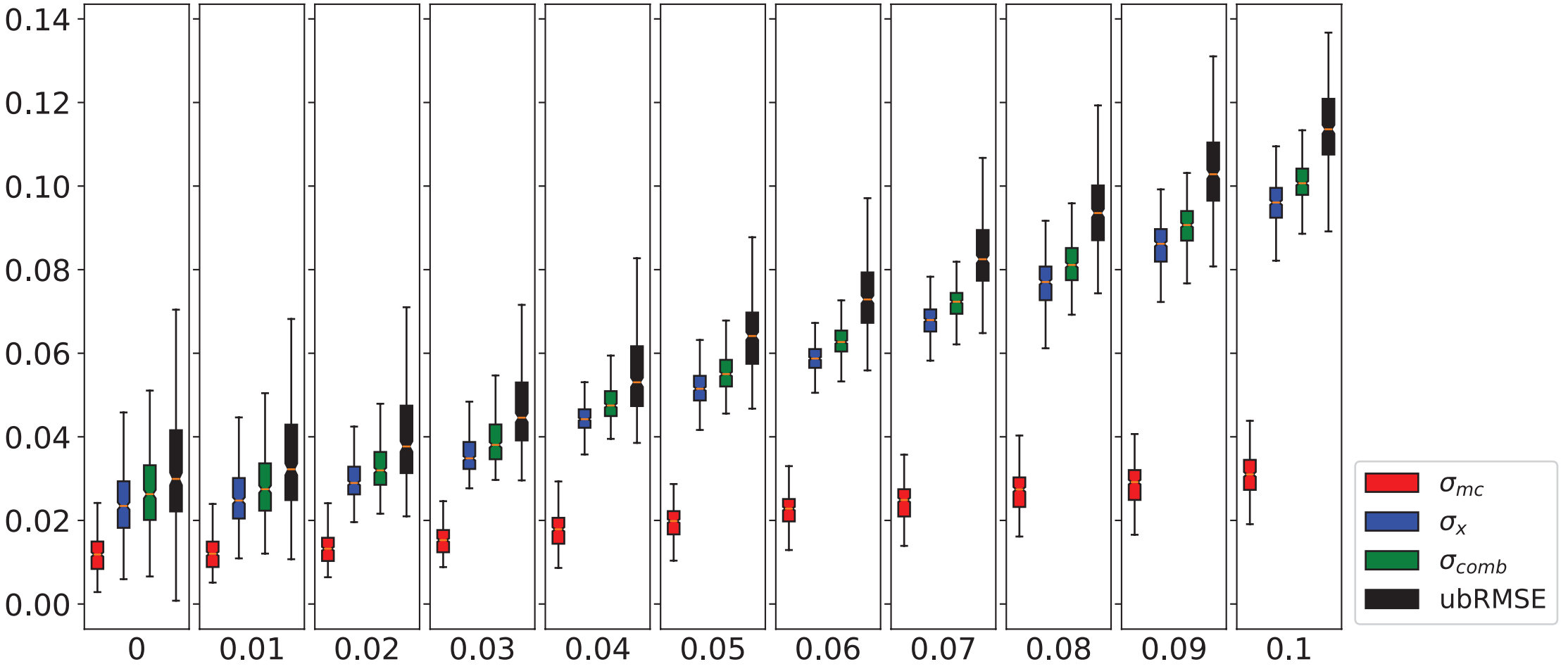
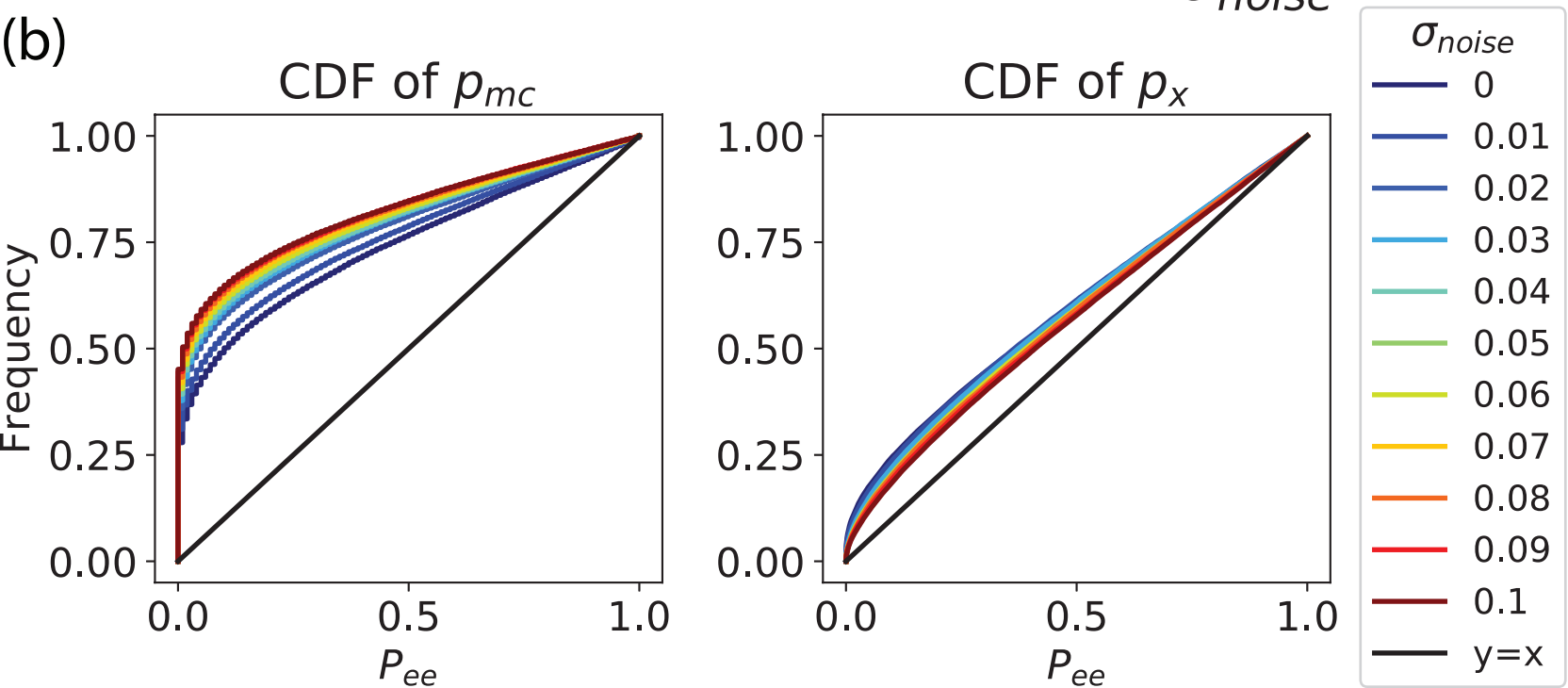

(a) Validation
(b) Temporal Test
p_x
p_{mc}
p_{comb}
y=x
Frequency
Error Exceedance Probablity
Error Exceedance Probablity
0.0
0.2
0.4
0.6
0.8
1.0

Figure 3.

(a) Error and uncertainty estimates in temporal test



(b)



(c)

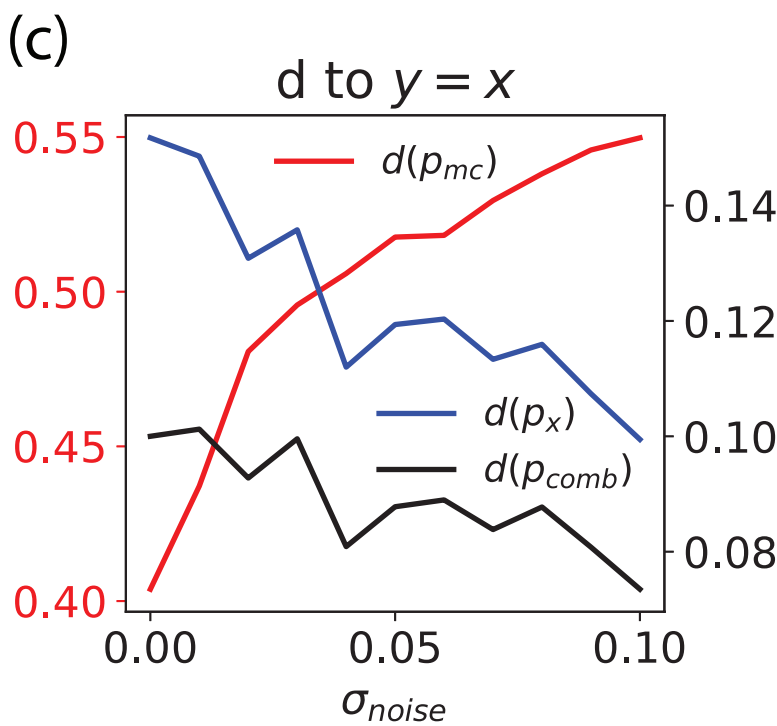
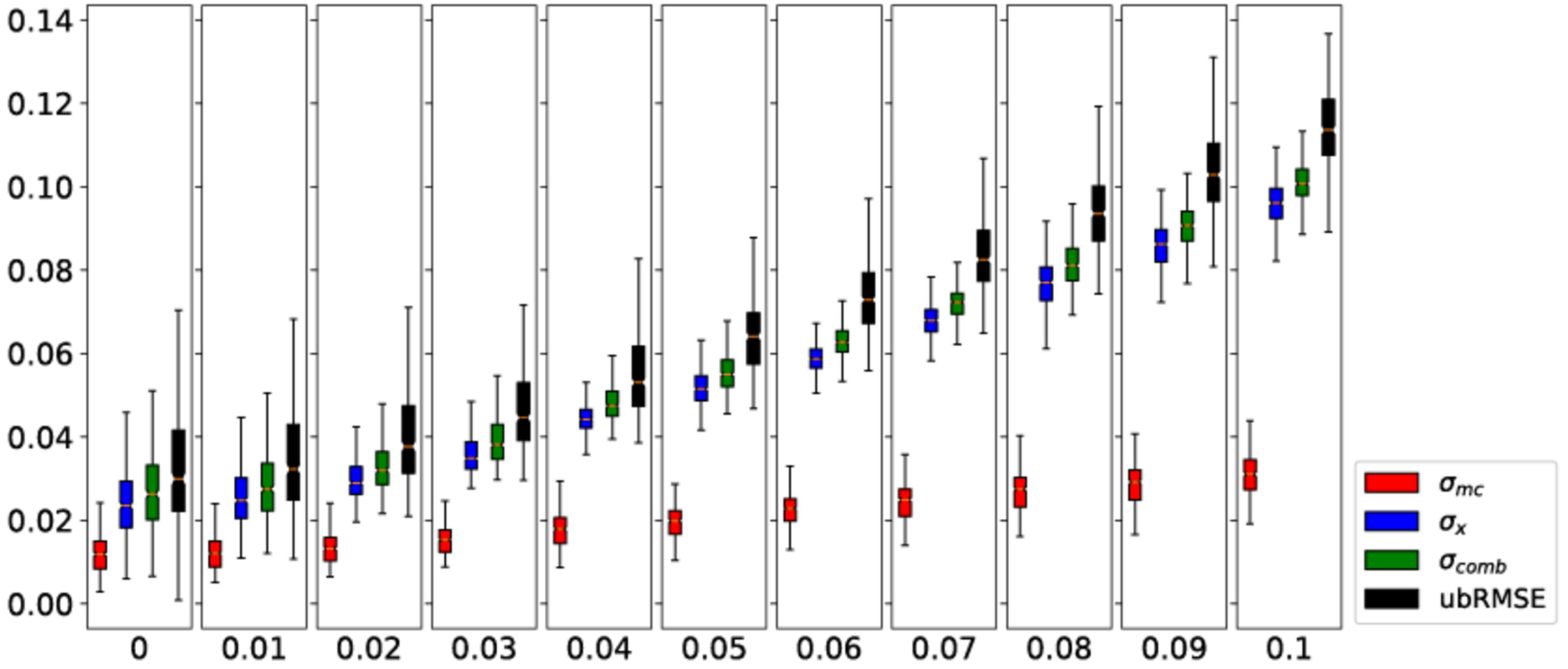
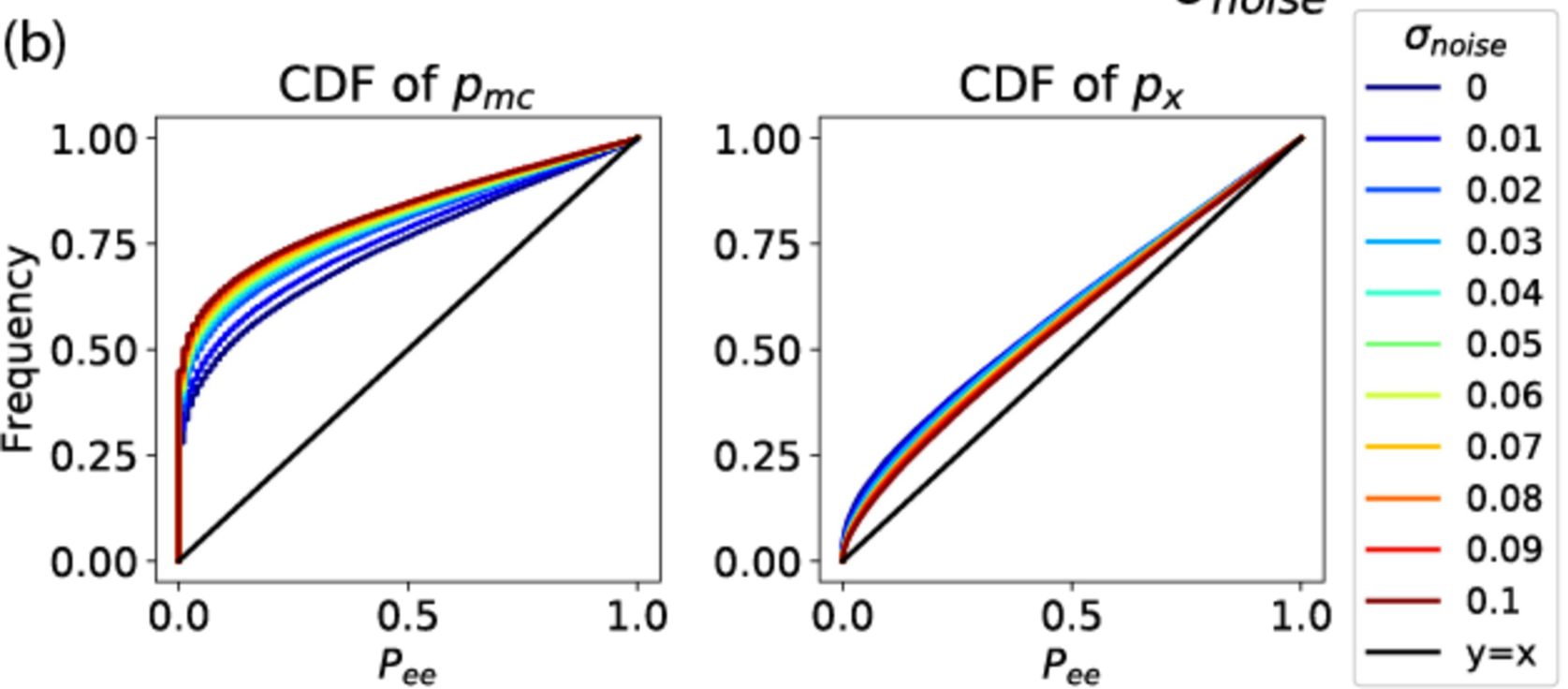


Figure 3 png ver.

(a) Error and uncertainty estimates in temporal test



(b)



(c)

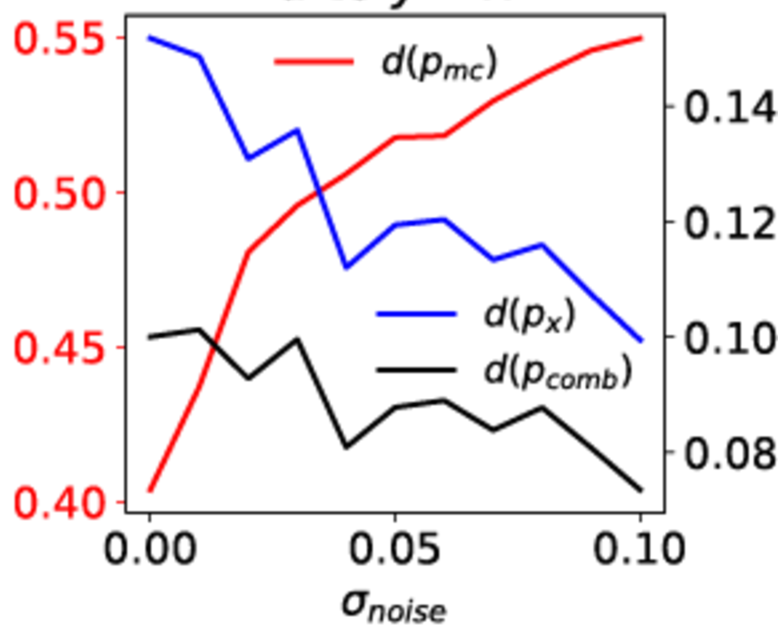


Figure 4.

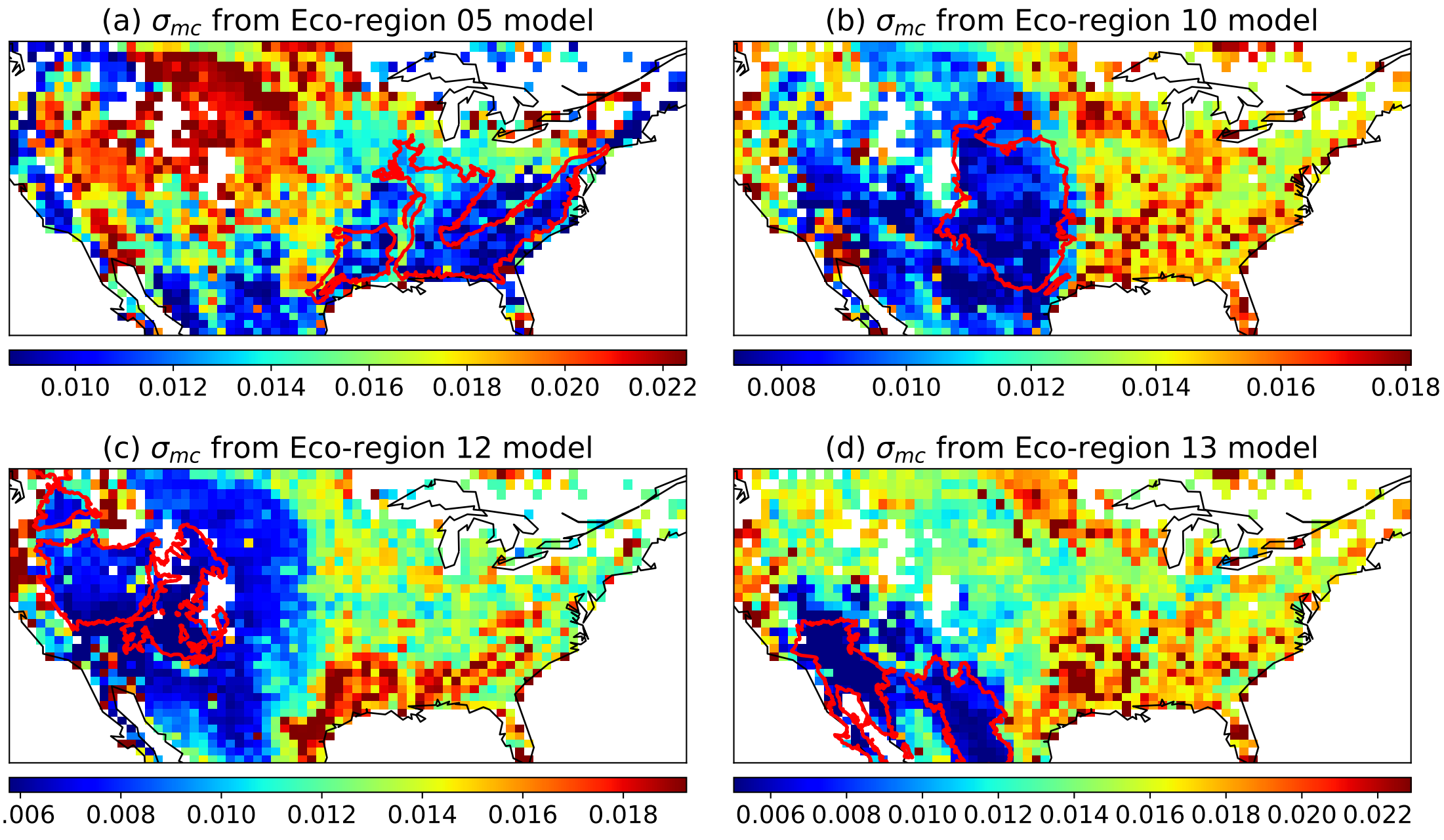

(a) σ_{mc} from Eco-region 05 model
0.010 0.012 0.014 0.016 0.018 0.020 0.022
(b) σ_{mc} from Eco-region 10 model
0.008 0.010 0.012 0.014 0.016 0.018
(c) σ_{mc} from Eco-region 12 model
0.006 0.008 0.010 0.012 0.014 0.016 0.018
(d) σ_{mc} from Eco-region 13 model
0.006 0.008 0.010 0.012 0.014 0.016 0.018 0.020 0.022

Figure 4 png ver.

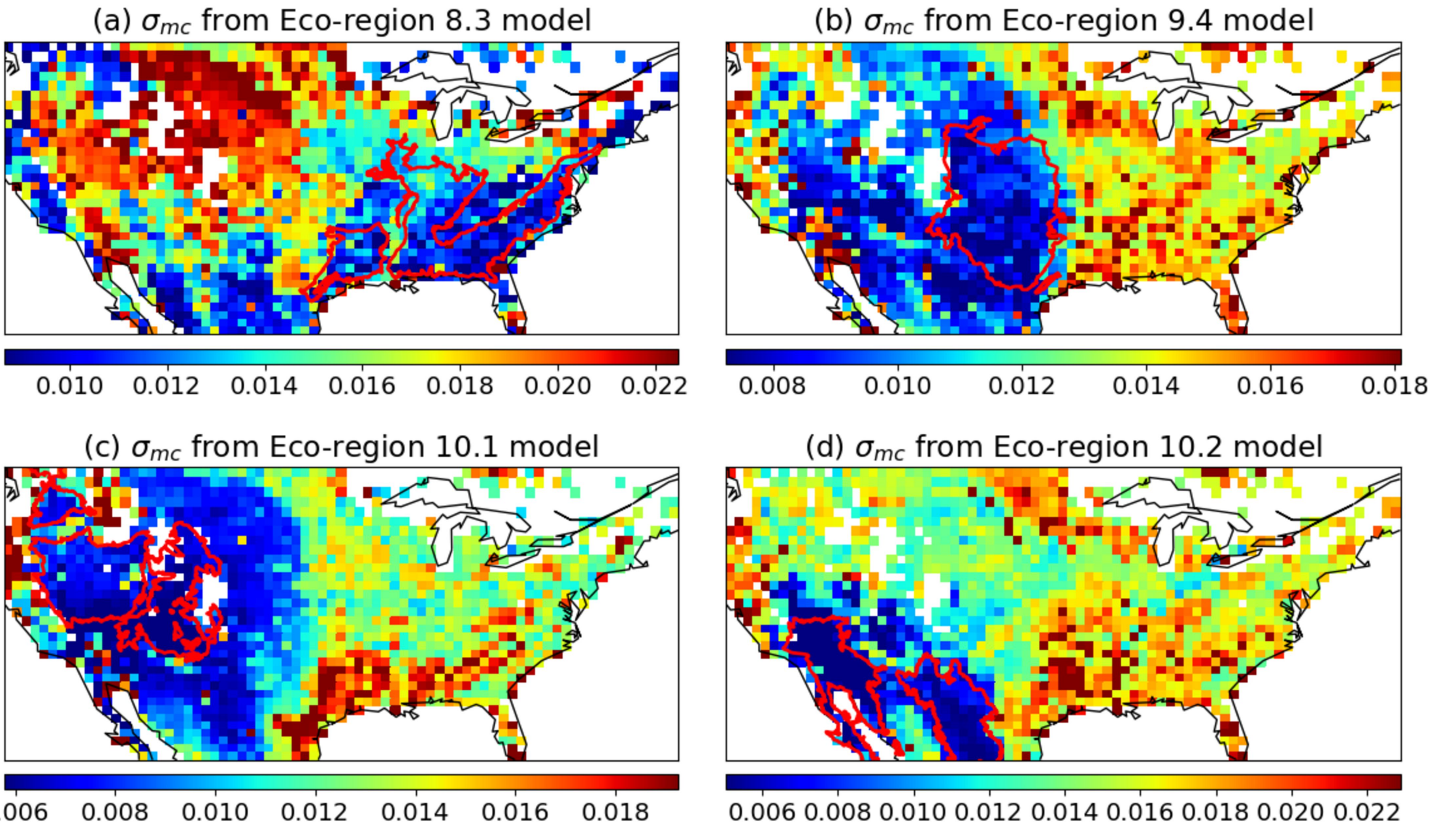

(a) σ_{mc} from Eco-region 8.3 model
0.010 0.012 0.014 0.016 0.018 0.020 0.022
(b) σ_{mc} from Eco-region 9.4 model
0.008 0.010 0.012 0.014 0.016 0.018
(c) σ_{mc} from Eco-region 10.1 model
0.006 0.008 0.010 0.012 0.014 0.016 0.018
(d) σ_{mc} from Eco-region 10.2 model
0.006 0.008 0.010 0.012 0.014 0.016 0.018 0.020 0.022

Figure 5.

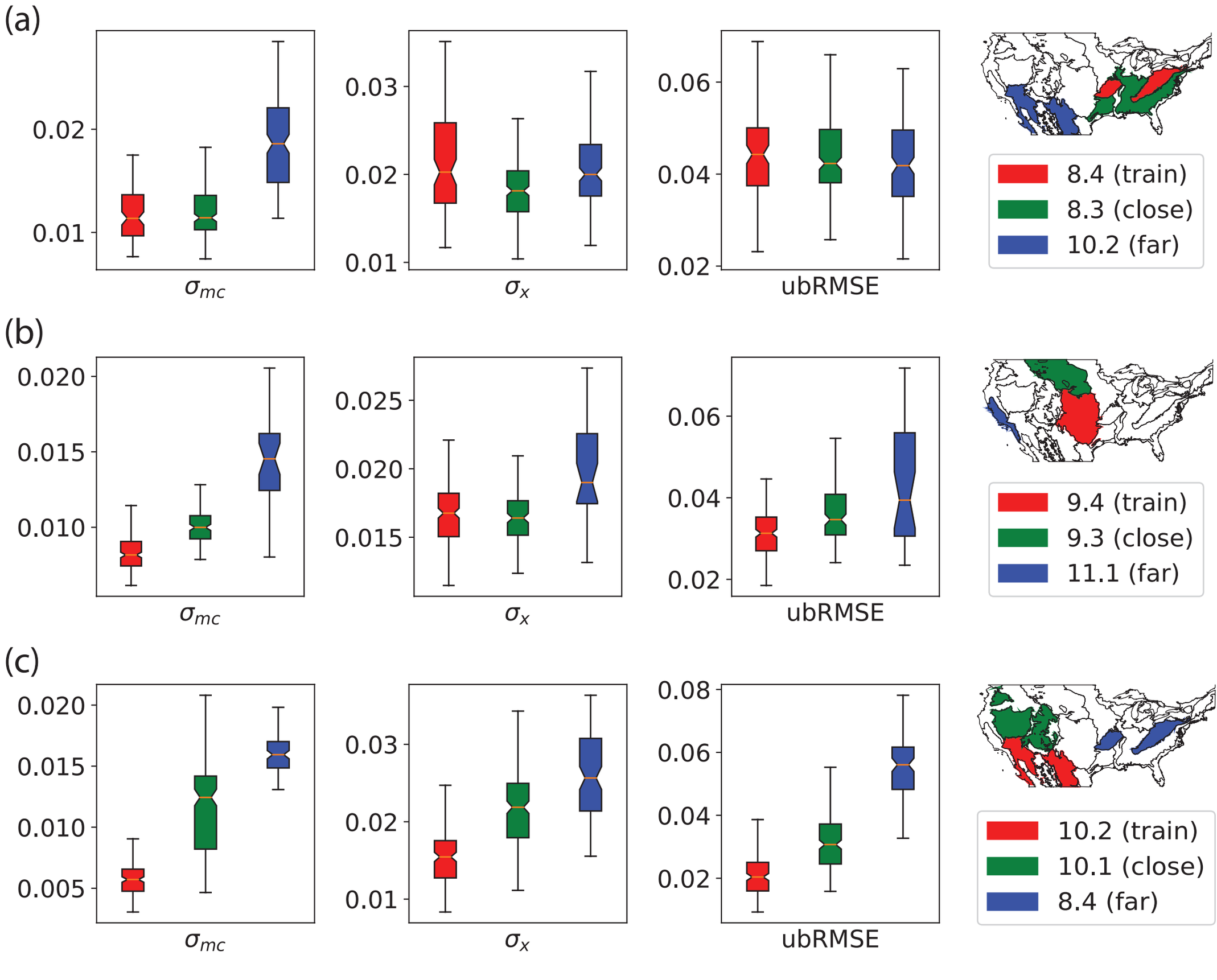

(a)
0.02
0.01
σ_{mc}
0.03
0.02
0.01
σ_x
0.06
0.04
0.02
ubRMSE
8.4 (train)
8.3 (close)
10.2 (far)
(b)
0.020
0.015
0.010
σ_{mc}
0.025
0.020
0.015
σ_x
0.06
0.04
0.02
ubRMSE
9.4 (train)
9.3 (close)
11.1 (far)
(c)
0.020
0.015
0.010
0.005
σ_{mc}
0.03
0.02
0.01
σ_x
0.08
0.06
0.04
0.02
ubRMSE
10.2 (train)
10.1 (close)
8.4 (far)

Figure 5 png ver.

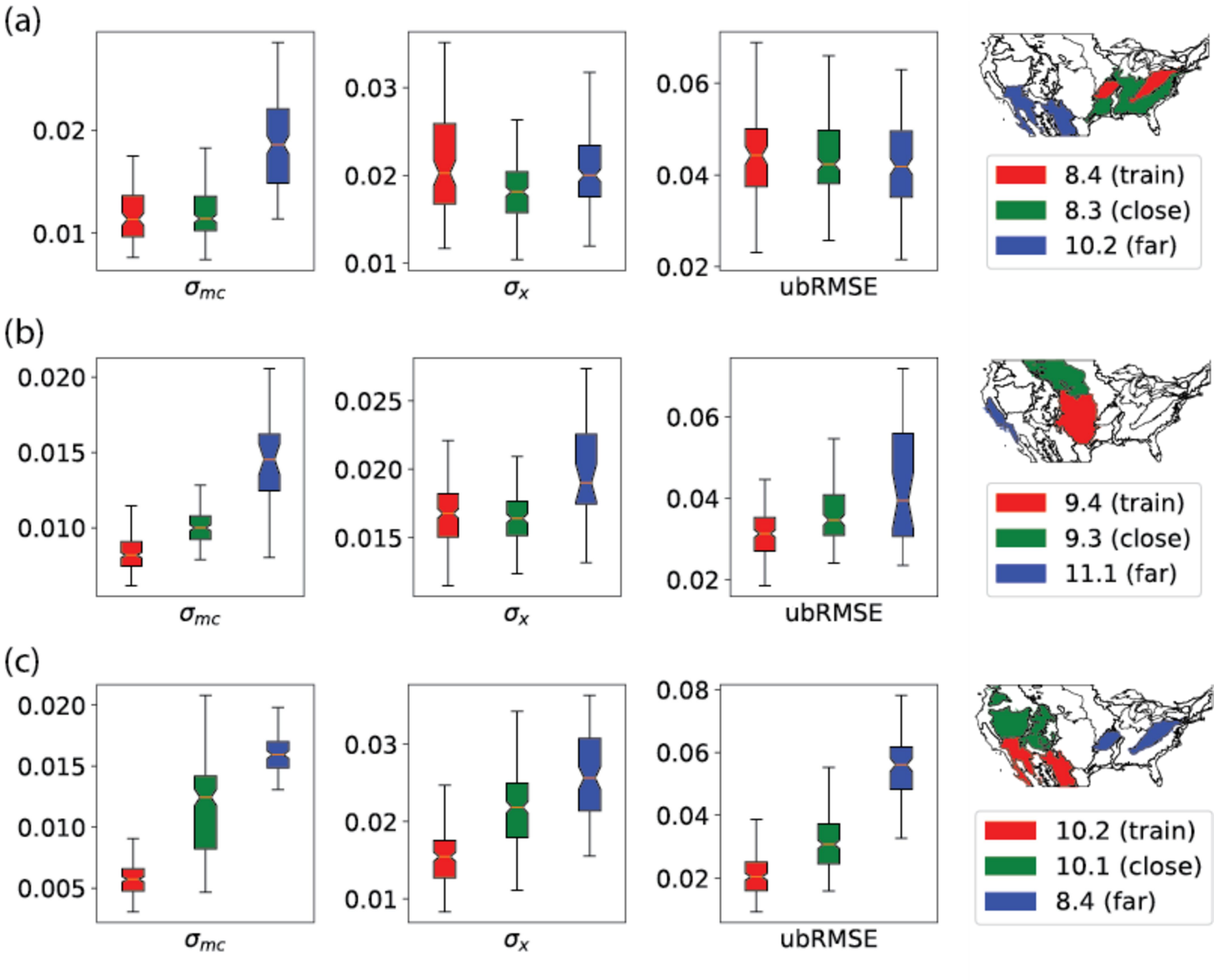

(a)
0.02
0.01
σ_{mc}
0.03
0.02
0.01
σ_x
0.06
0.04
0.02
ubRMSE
8.4 (train)
8.3 (close)
10.2 (far)
(b)
0.020
0.015
0.010
σ_{mc}
0.025
0.020
0.015
σ_x
0.06
0.04
0.02
ubRMSE
9.4 (train)
9.3 (close)
11.1 (far)
(c)
0.020
0.015
0.010
0.005
σ_{mc}
0.03
0.02
0.01
σ_x
0.08
0.06
0.04
0.02
ubRMSE
10.2 (train)
10.1 (close)
8.4 (far)

Figure 6.

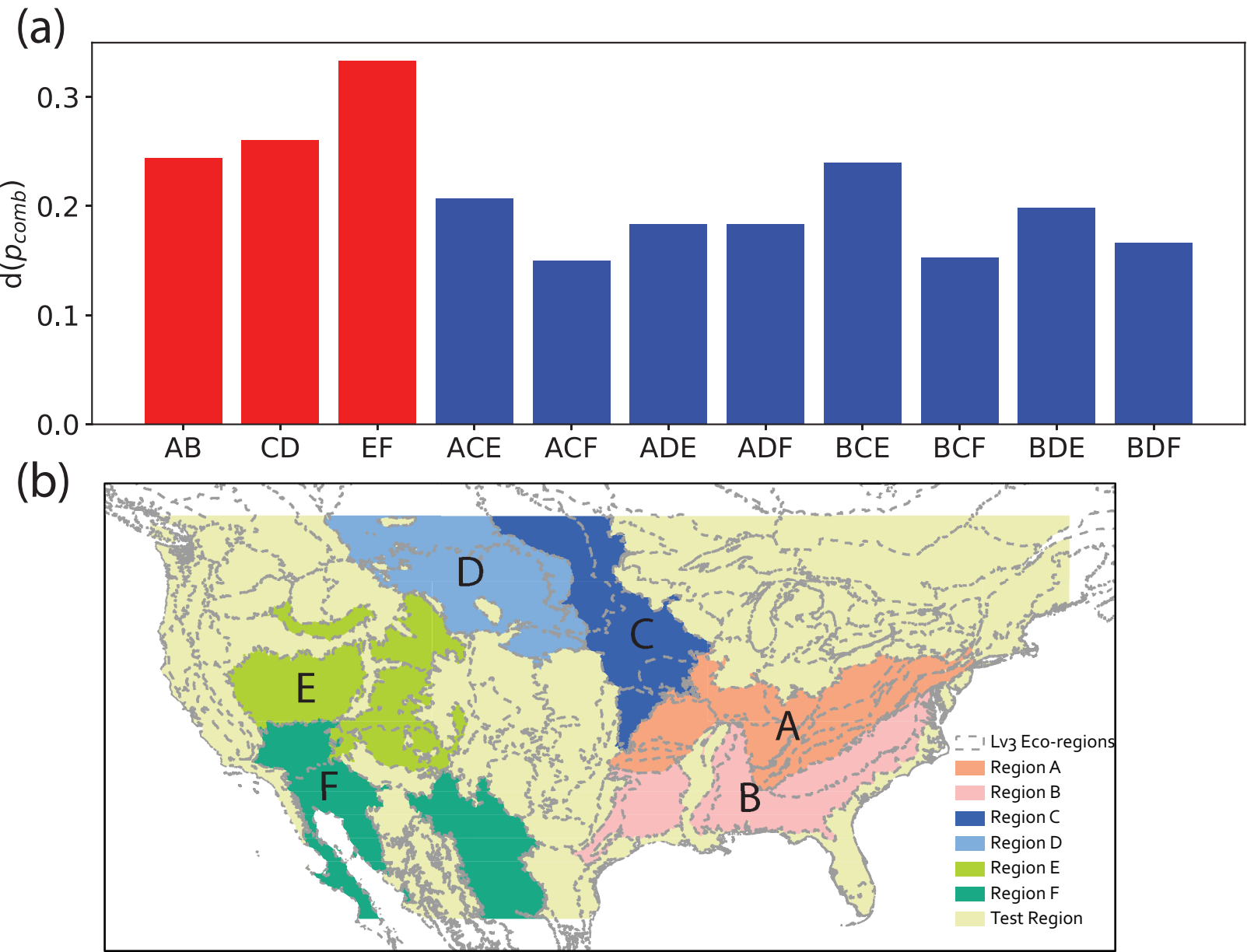

(a)
0.3
0.2
0.1
0.0
d(p_{comb})
AB
CD
EF
ACE
ACF
ADE
ADF
BCE
BCF
BDE
BDF
(b)
D
C
E
A
F
B
Lv3 Eco-regions
Region A
Region B
Region C
Region D
Region E
Region F
Test Region

Figure 6 png ver.

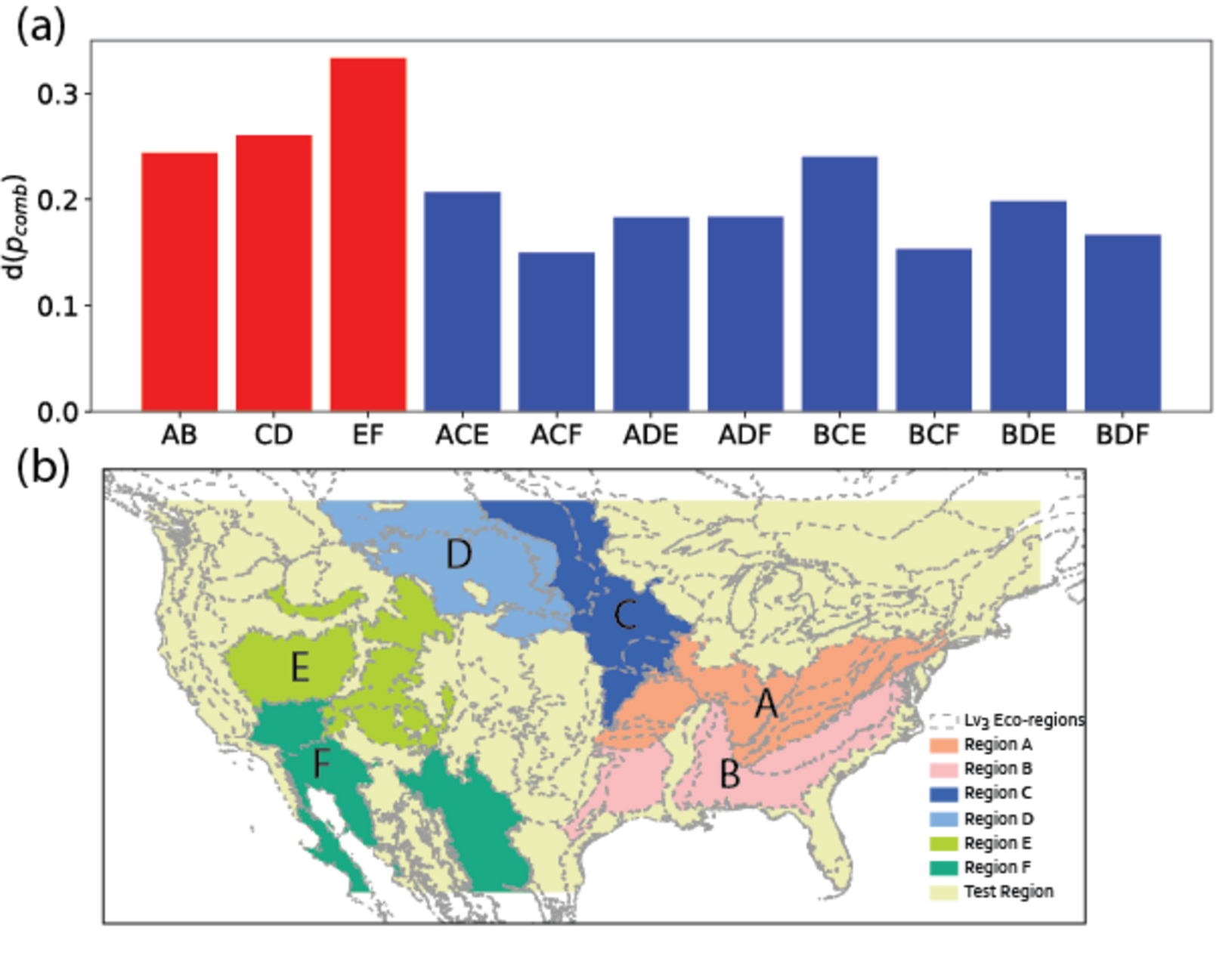

(a)
0.3
0.2
0.1
0.0
d(p_{comb})
AB
CD
EF
ACE
ACF
ADE
ADF
BCE
BCF
BDE
BDF
(b)
D
C
E
A
F
B
Lv3 Eco-regions
Region A
Region B
Region C
Region D
Region E
Region F
Test Region

Figure B1.

Model Error

ubRMSE

0.06
0.04
0.02
0.00

dr

0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

CDF of p_{comb}

Frequency

1.0
0.8
0.6
0.4
0.2
0.0

0.0
0.5
1.0

P_{ee}

Uncertainty Quality

d(p_{comb}, 1-to-1)

0.150
0.125
0.100
0.075
0.050
0.025

0.25
0.50
0.75

dr

Figure B1 png ver.

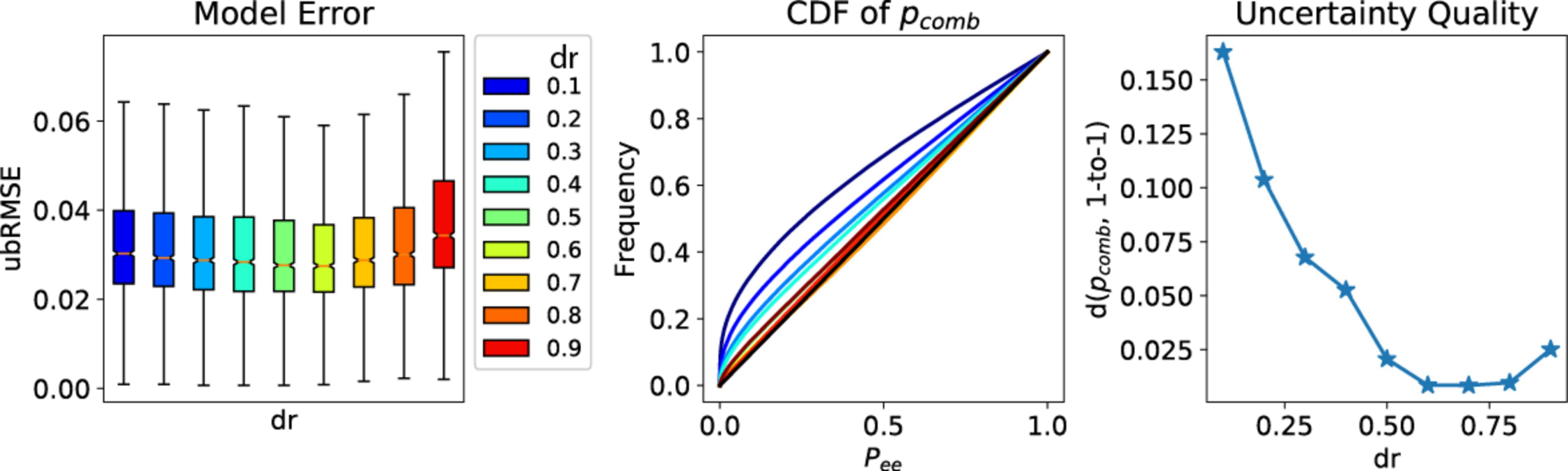

Model Error
ubRMSE
0.00
0.02
0.04
0.06
dr
dr
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
CDF of p_{comb}
Frequency
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
P_{ee}
Uncertainty Quality
d(p_{comb}, 1-to-1)
0.025
0.050
0.075
0.100
0.125
0.150
0.25
0.50
0.75
dr

Figure C1.

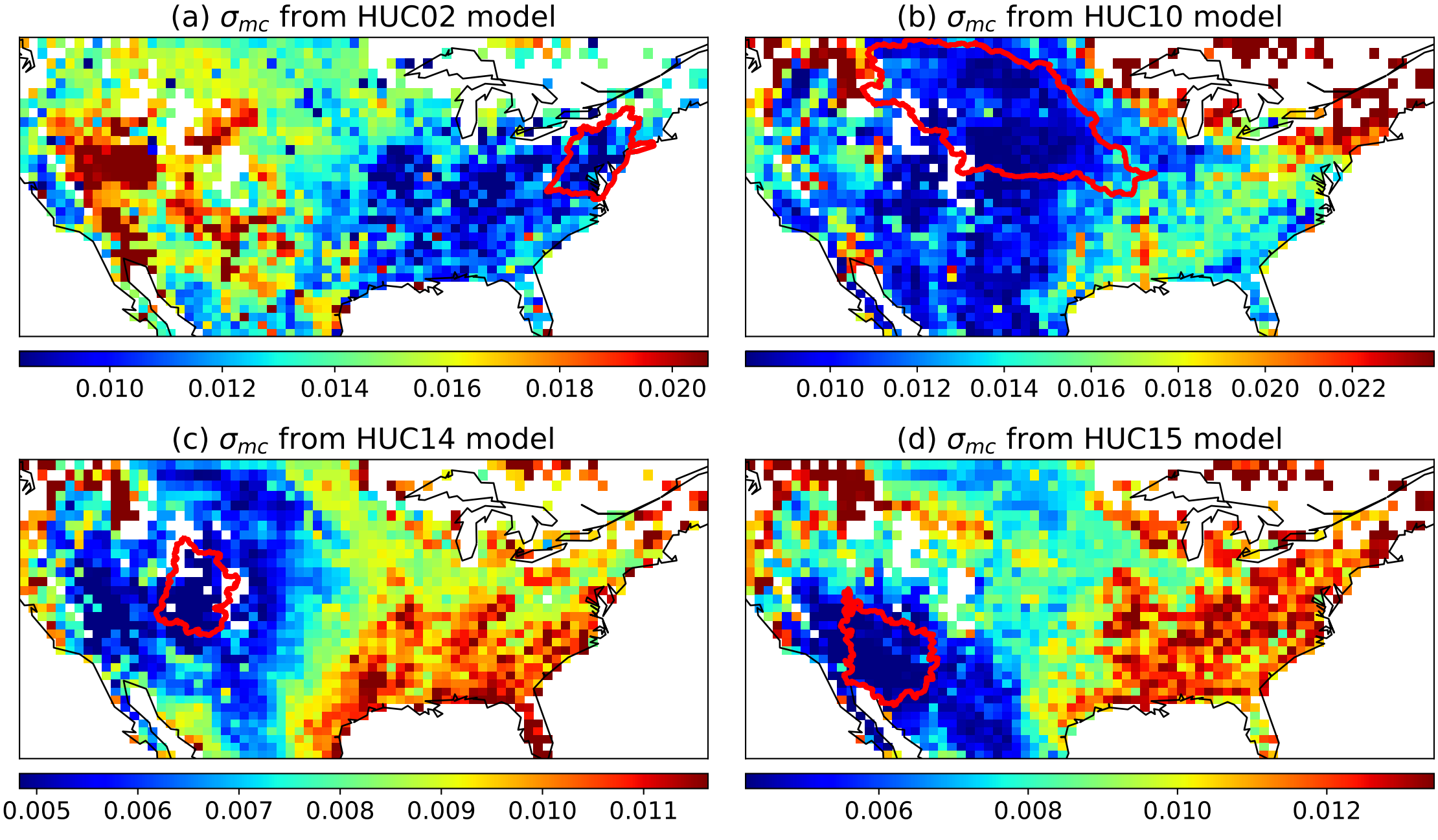

(a) σ_{mc} from HUC02 model
0.010
0.012
0.014
0.016
0.018
0.020
(b) σ_{mc} from HUC10 model
0.010
0.012
0.014
0.016
0.018
0.020
0.022
(c) σ_{mc} from HUC14 model
0.005
0.006
0.007
0.008
0.009
0.010
0.011
(d) σ_{mc} from HUC15 model
0.006
0.008
0.010
0.012

Figure C1 png ver.

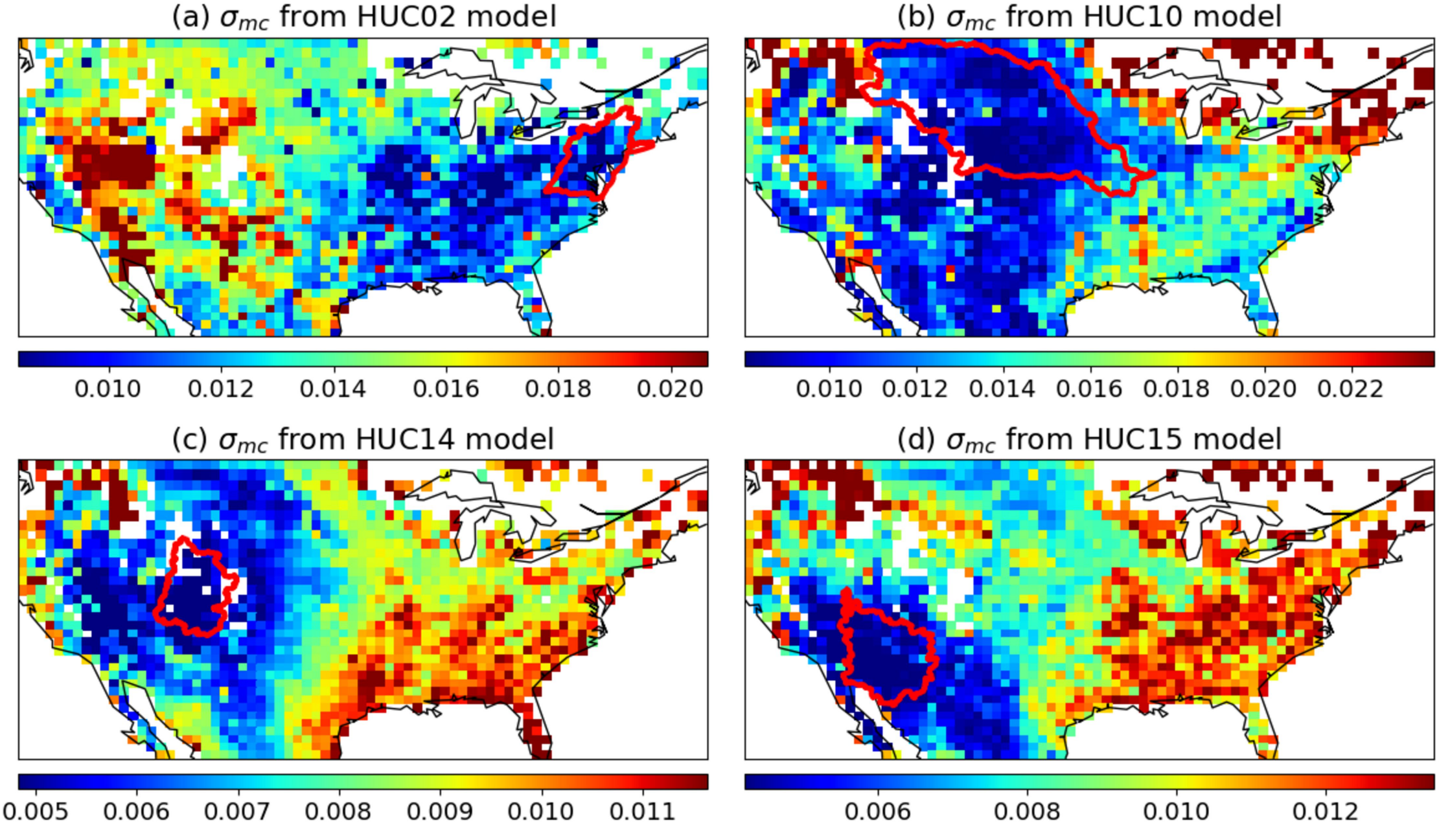

(a) σ_{mc} from HUC02 model
0.010
0.012
0.014
0.016
0.018
0.020
(b) σ_{mc} from HUC10 model
0.010
0.012
0.014
0.016
0.018
0.020
0.022
(c) σ_{mc} from HUC14 model
0.005
0.006
0.007
0.008
0.009
0.010
0.011
(d) σ_{mc} from HUC15 model
0.006
0.008
0.010
0.012

Figure C2.

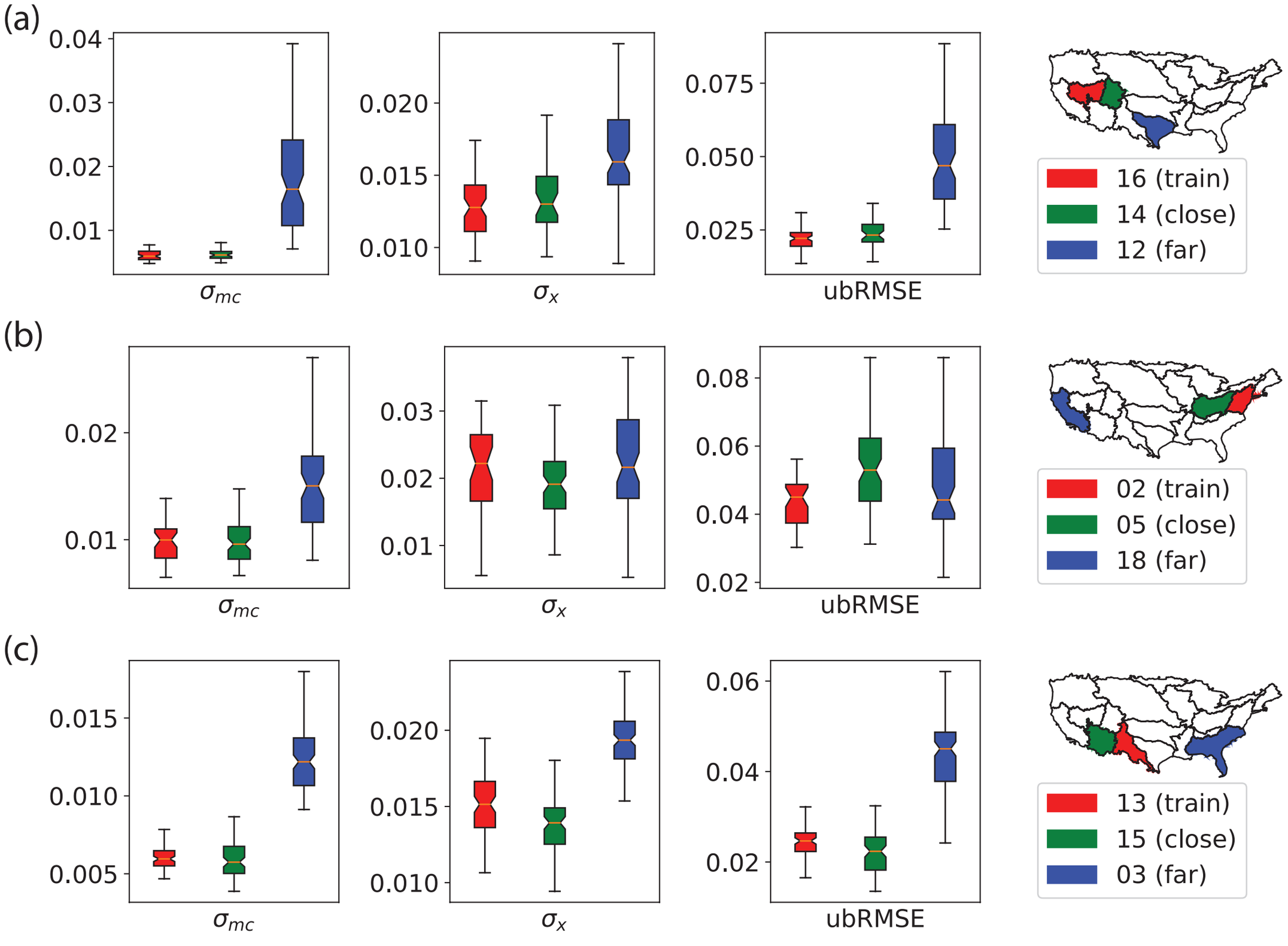
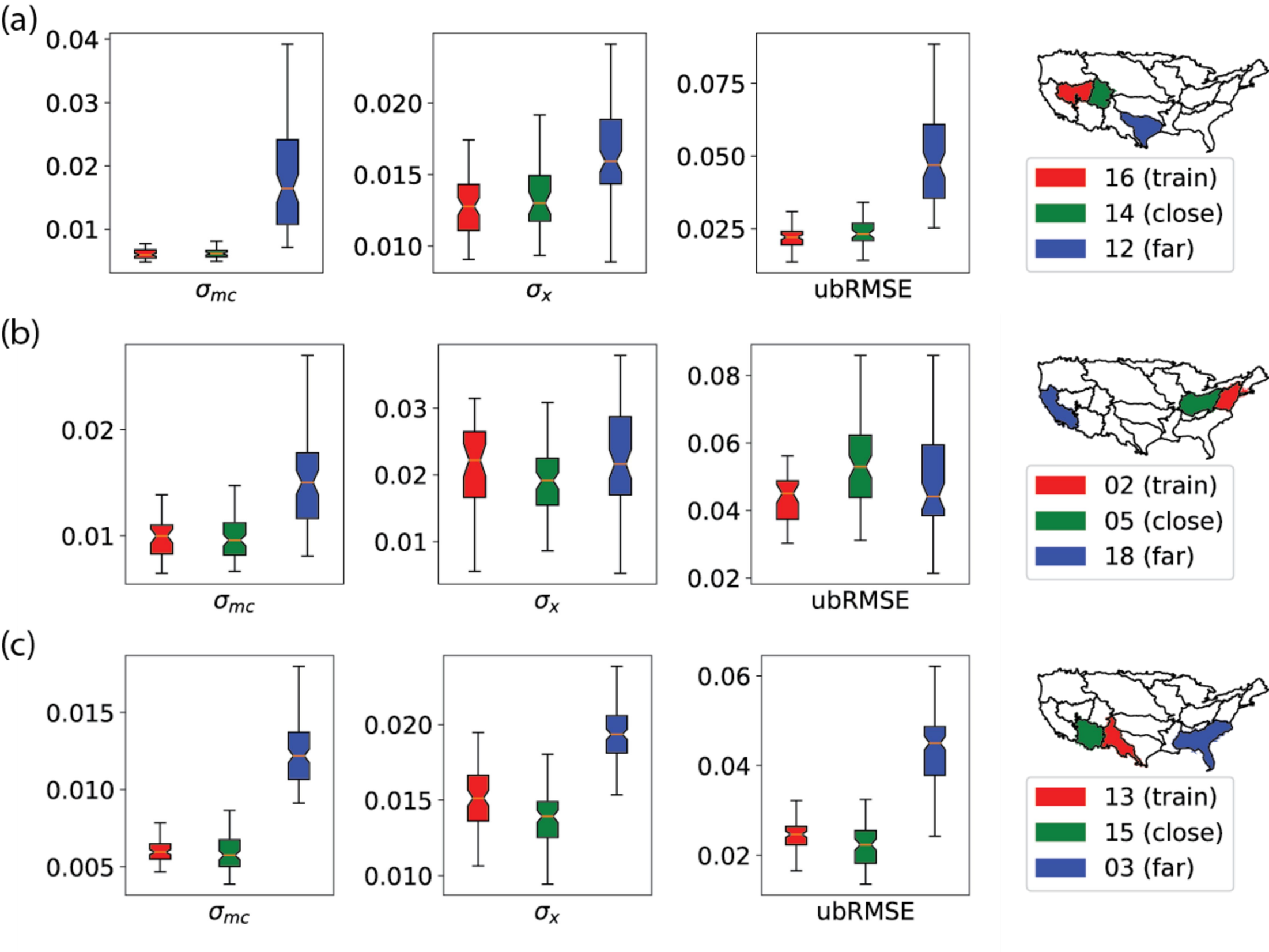

(a)
0.04
0.03
0.02
0.01
σ_{mc}
0.020
0.015
0.010
σ_x
0.075
0.050
0.025
ubRMSE
16 (train)
14 (close)
12 (far)
(b)
0.02
0.01
σ_{mc}
0.03
0.02
0.01
σ_x
0.08
0.06
0.04
0.02
ubRMSE
02 (train)
05 (close)
18 (far)
(c)
0.015
0.010
0.005
σ_{mc}
0.020
0.015
0.010
σ_x
0.06
0.04
0.02
ubRMSE
13 (train)
15 (close)
03 (far)

Figure C2 png ver.


(a)
0.04
0.03
0.02
0.01
σ_{mc}
0.020
0.015
0.010
σ_x
0.075
0.050
0.025
ubRMSE
16 (train)
14 (close)
12 (far)
(b)
0.02
0.01
σ_{mc}
0.03
0.02
0.01
σ_x
0.08
0.06
0.04
0.02
ubRMSE
02 (train)
05 (close)
18 (far)
(c)
0.015
0.010
0.005
σ_{mc}
0.020
0.015
0.010
σ_x
0.06
0.04
0.02
ubRMSE
13 (train)
15 (close)
03 (far)