To Reviewer # 3:

We appreciate the positive feedback from the reviewer. Thanks for the careful assessment. We found the comments extremely helpful and have revised accordingly.

To Comment # 1:

Concern: ***Calculation efficiency is a very important indicator. Can you give the calculation time of the proposed method, including the model training and prediction?***

Response: Yes, we completely agree with the comment that the computational efficiency is an important indicator in the surrogate modeling. The detailed results on the calculation efficiency of the proposed surrogate over conventional numerical analysis are:

- The solution of the high fidelity model is obtained by the finite element method (FEM) solver implemented in Matlab. The scripts are tested on 12 core Intel(R) Haswell processors with 256 GB RAM. The surrogate modeling algorithm is implemented in the TensorFlow. The scripts are tested on a single NVIDIA GeForce GTX 1080 Ti X GPU. More details on the computational resources can be found at `https://crc.nd.edu`.
- Training a surrogate model took 35 minutes of computational time but once trained uncertainty propagation of 1×10^5 was done in 40 minutes with field regressor solver whereas it took 5 seconds with FEM solver for a single sample, that is, 138.89 hours for the same uncertainty propagation task.

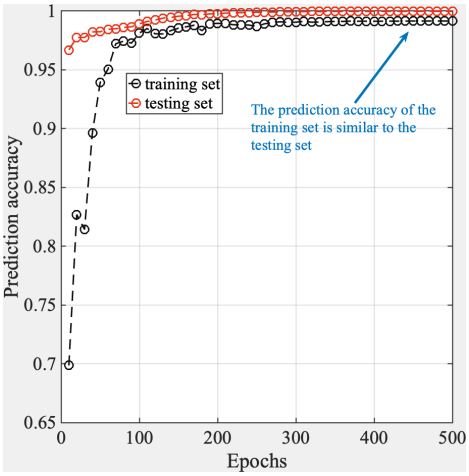
To Comment # 2:

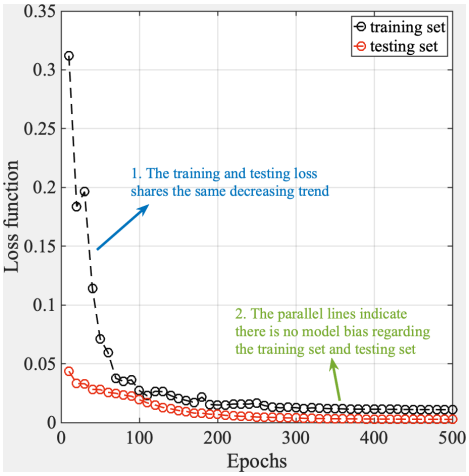
Concern: ***How to ensure that there will be no overfitting, in the ordinary study, how to determine the accuracy of the calculation?***

Response: Yes, thank you for raising this point. It is known that overfitting is a common problem in terms of training a deep learning model. Handling overfitting in deep learning models covers two parts: (1) detection of overfitting and (2) prevention of overfitting.

For (1), we adopted the following techniques:

- *Occam's Razor Test*: Take FR-21 from the first case for example, we started with a model that has 15 layers in total and used it as a benchmark. Then, we gradually increased the size of the network and compared the performance with FR-15. Once we have obtained satisfactory prediction performance, we set it as a reference point. We kept building new models until additional complexity has subtle impacts on the model performance. When we had two models of comparable performance, we picked the simpler one, that is, FR-21.

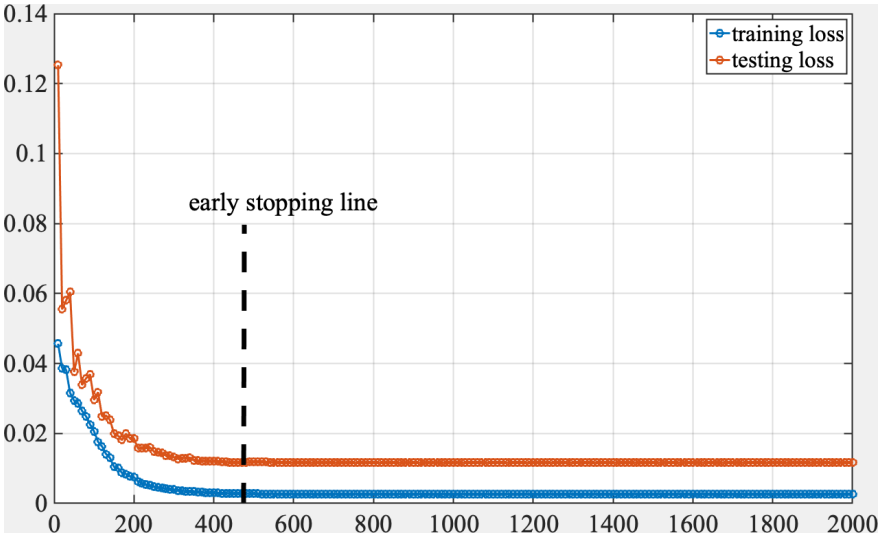




- *Data Splitting & Loss Inspection*: The collected was partitioned into the training set and testing set. First, we checked the prediction performance on these two sets, ensuring our model does not do much better on the training set than on the testing set. Secondly, we plotted the optimization history to check whether the training and testing loss shares the same trend.

For (2), we adopted the following techniques:

- *Early Stopping*: Instead of training for a fixed number of epochs, we stopped as soon as the validation loss rises. This was done by tracking the optimization history of the loss function. The figure below shows the early stopping point we researched via tracking the optimization process.



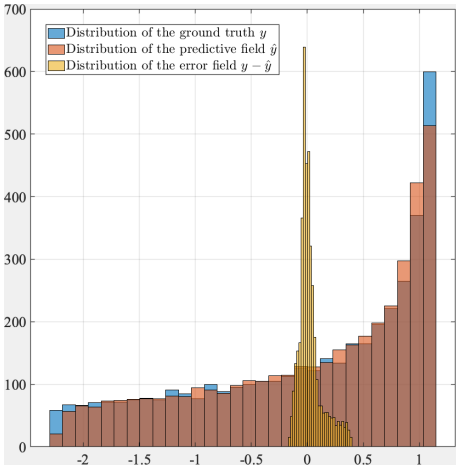
- *Dropout*: We pruned the network parameters by the dropout method. Consequently, some of the connections between adjacent layers are randomly deactivated. Note this technique was only used in searching optimized network architecture. Once the architecture has been decided, we set the dropout rate to 0.

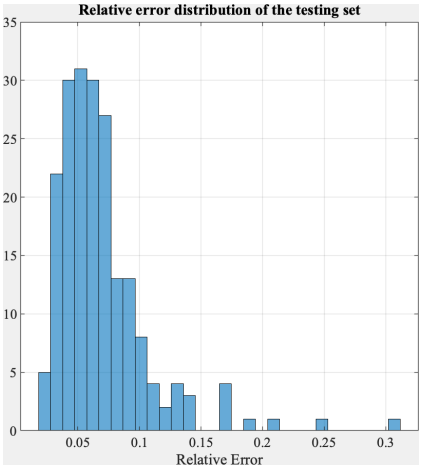
- *Cross Validation*: The standard K-fold cross validation method is adopted. First, we partitioned the data into K subsets. Then, we iteratively trained the model on K-1 folds while using the remaining fold as the test set.

The second part of this comment focuses on how to determine the accuracy of the calculation. In the previous version, we evaluated the trained model on unseen data and computed the difference $y-\hat{y}$ between the prediction $\hat{y}$ and the ground truth y. For illustration purpose, we randomly select a sample from the first case here and plot the distribution of the ground truth, predictions, and errors of the entire field. In the revised version, we have added another accuracy indicator that is defined as:

$$\mathcal{E}\left(\mathbf{u}_{\mathrm{FR}}, \mathbf{u}_{\mathrm{FEM}}\right) = \frac{\left|\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\left|\mathbf{u}_{\mathrm{FEM}}\right| - \sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\left|\mathbf{u}_{\mathrm{FR}}\right|\right|}{\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}\left|\mathbf{u}_{\mathrm{FEM}}\right|}$$

where $\mathcal{E}\left(\mathbf{u}_{\mathrm{FR}}, \mathbf{u}_{\mathrm{FEM}}\right)$ shows the corresponding relative absolute error regarding the predictive field. Details can be found in the optimization results in Section 4.





To Comment # 3:

Concern: ***English writing of the manuscript needs improvement.***

Response: All spelling and grammatical errors pointed out by the reviewers have been corrected. In addition, a systematic examination in terms of typos, omissions, and grammar mistakes has been conducted.