To Reviewer # 8:

We appreciate the positive feedback from the reviewer. Thanks for the careful revision. The reviewing comments are highly constructive and we have addressed them point-by-point.

To Comment # 1:

Concern: ***Please explicitly point out the novelty of your framework with details. In the abstract you refer to 3 model structures. Please list them specifically. Deep NN, MC, etc.? This will clarify your contribution.***

Response: Yes, thank you for the suggestion. We have made suggested revision to contribution and clarified statements in all parts of the paper.

To Comment # 2:

Concern: ***Consider using your method on existing experimental data. It is always illustrative and convincing when a framework is effective on such data.***

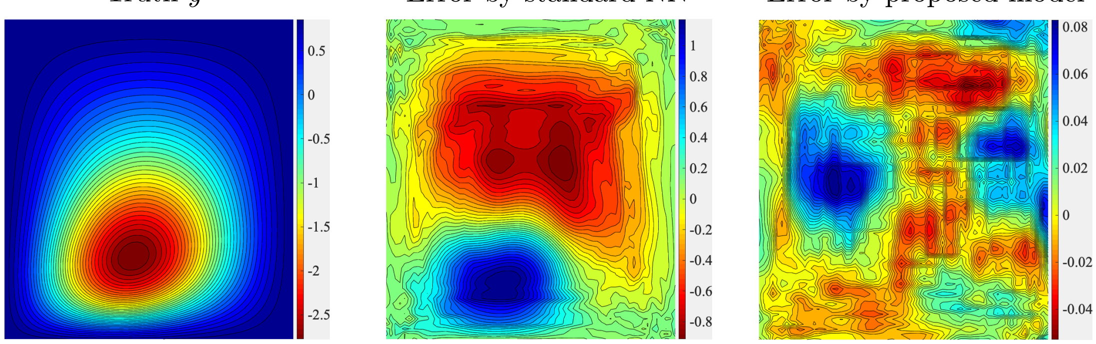
Response: Thank you for this comment. We have demonstrated the proposed method on a variety of examples including the Poisson's equation, Darcy's law, and nonlinear geometry analysis. Enclosed please find the results file (file name: summary.pdf).

To Comment # 3:

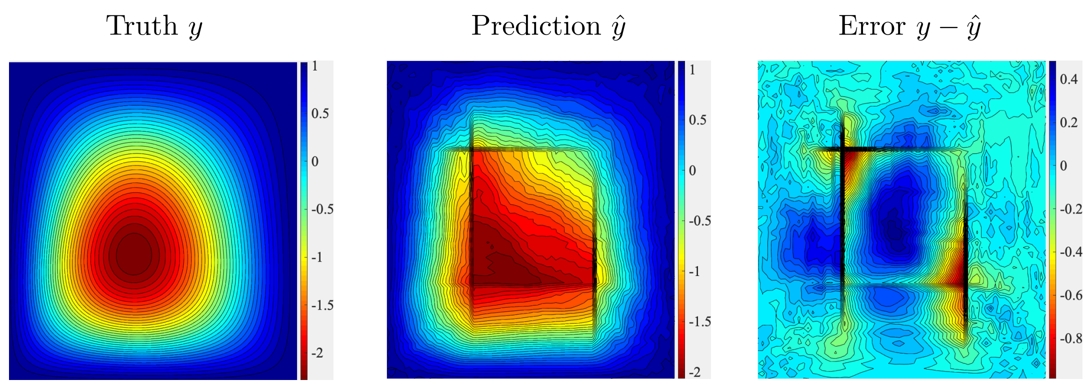
Concern: ***Can you compare these benchmark results to other standard NN without your improvements?***

Response: Yes, we have tested many models during the architecture design and hyperparameters identification. The main improvements of the proposed model include (1) composite function; (2) skip connection; and (3) bicubic resizing. Improvements (1) and (2) center on enhancing the prediction accuracy and training efficiency.

Truth y | Error by standard NN | Error by proposed model



On the other hand, improvement (3) focuses on solving checkerboard effects.



checkerboard effects

To Comment # 4:

Concern: ***Grammatical errors in the abstract.***

Response: Yes, we have corrected spelling and grammatical errors.

To Comment # 5:

Concern: ***"nonlinear structures" is confusing. Do you mean nonlinear computational architectures or nonlinear (e.g. material, geometric) properties of structural systems?***

Response: Thank you so much for noting this phrase. The "*nonlinear structure*" dentoes the nonlinear computational architecture. Because Karhunen-Loève expansion (KLE) seeks the closest linear subspace of the high-dimensional input space, it is suitable for multi-Gaussian random fields. But for samples contain nonlinear structures where higher-order statistics are critical in random field simulation, KLE tends to over-estimate the intrinsic dimensionality of the underlying random field.

To Comment # 6:

Concern: ***Is this statement a finding of this paper? or does it need a citation from a previous paper?***

Response: Thank you for the question. In fact, the topic of using surrogate models for uncertainty quantification has been extensively studied. However, few attempts have been made in the context of modeling high-dimensional input-output relationship. Encouraged by the great success of deep learning in the field of computer vision, image processing, and pattern recognition, we proposed this image-to-image model where engineering property fields represented via random fields are treated as images. We have added some references as the reviewer suggested.

To Comment # 7:

Concern: ***This notation is unclear. Please provide variable names and operator designation.***

Response: Yes, thank you for this careful observation. $\circ$ denote the composite operations, that is, $(f \circ g)(x) = f(g(x))$. We have added this explanation into the revised manuscript.

To Comment # 8:

Concern: ***Formatting issue.***

Response: Yes, we have corrected the Format.

To Comment # 9:

Concern: ***Remove colon.***

Response: Yes, we have corrected it.

To Comment # 10:

Concern: ***Please provide details on the finite element simulation. Such as the software used, solver name, etc. Do the FE parameters approach the equations 22?***

Response: Yes, thank you for bringing this point to our attention. All finite element analyses are carried out on 12 core Intel(R) Haswell processors with 256 GB RAM using the finite element codes programmed in Matlab. The computer codes are inspired by examples covered in [1,2].

Reference

1. Kwon, Y.W. and Bang, H., 2018. The finite element method using MATLAB. CRC press.
2. Ferreira, A.J., 2008. MATLAB codes for finite element analysis: solids and structures (Vol. 157). Springer Science & Business Media.

To Comment # 11:

Concern: ***These points are excellent. Did you compare the method in this paper to a standard BPNN or deep BPNN? Performance, computational time, etc. would be interesting results.***

Response: Thank you for the positive feedback and this suggestion. It would have been interesting to explore this aspect. In the case of pixel-wise prediction, the parameter space of a deep BPNN model can be extremely high. Take case 1 for example, the input/output dimension is 4096. If the BPNN model has 3 hideen layers where each layer has 500 neurons, the dimension of connection weights goes up to 909600, escalating the challenges of model training. Hence, it may be more appropriate not using conventional BPNN in this case.