Infinite-Fidelity Surrogate Learning via High-Order Gaussian Processes

Shibo Li [1] **Li Shi** [2] **Shandian Zhe** [1]

Abstract

Multi-fidelity learning is popular in computational physics. While the fidelity is often up to the choice of mesh spacing and hence is continuous in nature, most methods only model finite, discrete fidelities. The recent work (Li et al., 2022a) proposes the first continuous-fidelity surrogate model, named infinite-fidelity coregionalization (IFC), which uses a neural Ordinary Differential Equation (ODE) to capture the rich information within the infinite, continuous fidelity space. While showing state-of-the-art predictive performance, IFC is computationally expensive in training and is difficult for uncertainty quantification. To overcome these limitations, we propose Infinite-Fidelity High-Order Gaussian Process (IF-HOGP), based on the recent GP high-dimensional output regression model HOGP. By tensorizing the output and using a product kernel at each mode, HOGP can highly efficiently estimate the mapping from the PDE parameters to the high-dimensional solution output, without the need for any low-rank approximation. We made a simple extension by injecting the continuous fidelity variable into the input, and applying a neural network transformation before feeding the input into the kernel. On three benchmark PDEs, IF-HOGP achieves prediction accuracy better than or close to IFC, yet gains 380x speed-up and 87.5% memory reduction. Meanwhile, uncertainty calibration for IF-HOGP is straightforward.

1 Method

Physical simulation is the cornerstone of modern engineering and scientific applications. The central task in physical simulation is to solve a variety of partial differential equations (PDEs). Given the PDE parameters and initial/boundary conditions, conventional numerical methods discretize the PDEs (on a mesh) and convert solving them into iteratively solving a linear system of equations. While successful, the traditional methods are known to be computationally expensive. To reduce the cost, it is natural to consider training a surrogate model with examples of PDE solutions generated by the numerical methods. Given new PDE parameters, one can use the surrogate model to predict the solution, *e.g.*, via a forward pass of a deep neural network, which is therefore much more efficient.

However, collecting the training data still demand we run the numerical solvers, which is a costly bottleneck. To further reduce the cost, one can consider generating multi-fidelity solution examples. High-fidelity examples are computed with dense meshes, and are accurate but expensive. Low-fidelity examples are computed from coarse meshes, and therefore are cheap yet inaccurate. Despite the discrepancy in accuracy, low-fidelity and high-fidelity examples can be strongly correlated since they are based on the same PDE(s). Many multi-fidelity surrogate learning methods have therefore been proposed to synergnize examples at different fidelities to improve the prediction accuracy while reducing the data cost, such as (Parussini et al., 2017; Xing et al., 2021b; Wang et al., 2021).

Despite the success, existing approaches build surrogate models on a pre-specified set of finite, discrete fidelities. However, the fidelity is often determined by the mesh spacing (or finite element size) used in the numerical solvers, and hence is continuous in nature. In other words, one can choose infinitely many fidelities to generate solution data. To extract and leverage the rich information within the infinite, continuous fidelity space, Li et al. (2022a) proposed IFC, the first infinite fidelity surrogate learning approach for high-dimensional physical simulation outcomes. Note that continuous-fidelity modeling has been studied in Bayesian optimization literature(Kandasamy et al., 2017; Wu & Frazier, 2018), but that line of work focus on single-output (objective) functions, and cannot scale to massive function outputs.

Given the continuous fidelity m and PDE parameters $\mathbf{x}$, IFC first models a low-dimensional latent output $\mathbf{h}(m, \mathbf{x})$

[1]Kahlert School of Computing, The University of Utah, Salt Lake City, UT, United States, 84112 [2]The English Language Institute, The University of Utah. Correspondence to: Shandian Zhe <zhe@cs.utah.edu>.

Accepted after peer-review at the 1st workshop on Synergy of Scientific and Machine Learning Modeling, SynS & ML ICML, Honolulu, Hawaii, USA. July, 2023.

with a neural Ordinary Differential Equation (ODE) (Chen et al., 2018), and then predicts the high-dimensional solution output via a linear transformation,

$$\frac{\partial \mathbf{h}}{\partial m} = \text{NN}(\mathbf{h}, m, \mathbf{x}), \quad \mathbf{y}(m, \mathbf{x}) = \mathbf{B}(m) \cdot \mathbf{h}, \qquad (1)$$

where NN denotes a neural network, $\mathbf{y}(m, \mathbf{x})$ is the solution prediction, and $\mathbf{B}$ is the basis matrix that varies along with m. To estimate $\mathbf{B}(m)$, Li et al. (2022a) proposed two methods. One is to assign a Gaussian process (GP) prior over each element of $\mathbf{B}(m)$, denoted by IFC-GPODE, and the other is to use a neural ODE to model each $[\mathbf{B}(m)]_{i,j}$, denoted by IFC-ODE^2.

While IFC has shown the state-of-the-art predictive performance on a series of benchmark tasks, it falls short in two aspects. First, IFC is computationally expensive in training. The challenge is rooted in the ODE formulation (1). To calculate the model likelihood (training objective), one has to run a numerical integrator, like RK45, to solve the ODE, which typically needs many evaluations of the ODE dynamics over a small step-size. To back-propagate the gradient w.r.t the model parameters, one can either use computational graphs to perform automatic differentiation or the adjoint state method to solve a backward companion ODE. For the former, the computational graph grows fast and can easily exhaust the memory. The computation is costly since it has to track a large number of steps in the forward solving. Though the latter is more memory efficient, the computation is even slower and is often unstable (solving the adjoint ODE in backward/reverse time is easy to diverge) (Gholaminejad et al., 2019). Second, IFC is difficult to quantify the uncertainty, especially for the prediction. Due to the ODE modeling, estimating a posterior distribution of the model parameters is challenging. We followed (Dandekar et al., 2020) to try a variety of popular Bayesian inference methods, including variational inference, Hamiltonian Monte-Carlo (HMC), stochastic HMC, stochastic gradient Langevin dynamics (SGLD), which however, deteriorate the predictive performance yet not guaranteeing a reasonable uncertainty estimate.

To address these problems, we propose a new infinite-fidelity surrogate learning approach based on the recent GP high-dimensional output regression model, HoGP (Zhe et al., 2019). The advantage of HoGP is twofold. First, HoGP does not involve any numerical integration, and is highly efficient in both the computation speed and memory usage. Second, as a GP model, HoGP can conveniently compute the posterior distribution of the prediction. We henceforth named our approach as IFHoGP. Specifically, we combine the continuous fidelity m and PDE parameters $\mathbf{x}$ as the input vector. To improve the representation power, we apply an NN to obtain a transformed input $\widetilde{\mathbf{x}} = \text{NN}(\mathbf{x}, m)$. We then feed $\widetilde{\mathbf{x}}$ into the HoGP model. To handle the high-dimensional output, HoGP tensorizes the output as an s-mode tensor, $D = d_1 \times d_2 \times \cdots \times d_s$, where D is the output dimension, and $\{d_1, \ldots d_s\}$ are the dimensions of each tensor mode. We then introduce d_j feature vectors to represent each coordinate $1, \ldots, d_j$ in mode j. We denote the feature matrix by $\mathbf{V}^j = [\mathbf{v}_1^j, \ldots, \mathbf{v}_{d_j}^j]^\top$. Accordingly, we construct a product kernel to model the covariance between arbitrary two elements in the output tensors,

$$\begin{aligned} &\kappa(y_{\mathbf{i}}(m, \mathbf{x}), y_{\mathbf{j}}(m', \mathbf{x}')) \\ &= \kappa\left(\text{NN}(m, \mathbf{x}), \text{NN}(m', \mathbf{x}')\right) \cdot \prod_{k=1}^{s} \kappa(\mathbf{v}_{i_k}^k, \mathbf{v}_{j_k}^k) \end{aligned} \qquad (2)$$

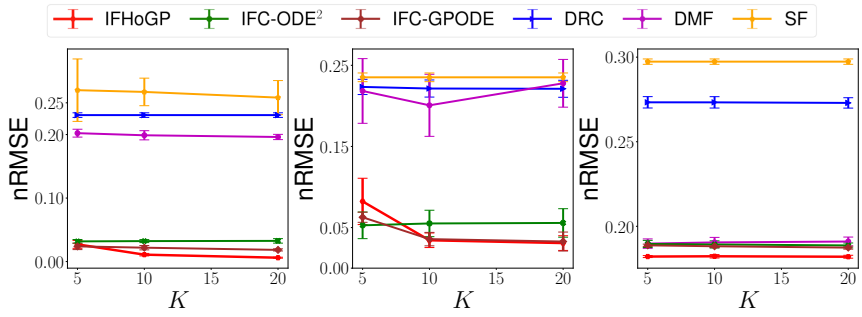
where $\mathbf{i} = (i_1, \ldots, i_s)$ and $\mathbf{j} = (j_1, \ldots, j_s)$ are indices of two tensor elements, $y_{\mathbf{i}}(m, \mathbf{x})$ and $y_{\mathbf{j}}(m', \mathbf{x}')$ represent the output element of $\mathbf{y}(m, \mathbf{x})$ indexed by $\mathbf{i}$, and of $\mathbf{y}(m', \mathbf{x}')$ indexed by $\mathbf{j}$, respectively. Accordingly, given the training dataset $\mathcal{D} = \{(m_1, \mathbf{x}_1, \mathbf{y}_1), \ldots, (m_N, \mathbf{x}_N, \mathbf{y}_N)\}$, the marginal likelihood of IFHoGP is given by

$$p(\mathcal{Y}) = \mathcal{N}(\text{vec}(\mathcal{Y}) | \mathbf{0}, \mathbf{K}^1 \otimes \cdots \otimes \mathbf{K}^s \otimes \mathbf{K} + \tau^{-1}\mathbf{I}) \qquad (3)$$

where $\mathcal{Y}$ is a $d_1 \times \ldots \times \mathrm{d}_s \times N$ tensor which stacks all the N tensorized training outputs, $\mathbf{K}^j = \kappa(\mathbf{V}^j, \mathbf{V}^j; \boldsymbol{\theta}_j)$ is the covariance matrix at mode j, $\mathbf{K} = \kappa(\mathbf{X}, \mathbf{X}; \boldsymbol{\theta})$ is the covariance matrix for the transformed training inputs $\mathbf{X} = [\text{NN}(m_1, \mathbf{x}_1), \ldots, \text{NN}(m_N, \mathbf{x}_N)]^\top$, $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_j$ are the kernel parameters, and τ is the inverse noise variance. We maximize (3) to estimate all the model parameters. Due to the structure in the covariance, we can use the Kronecker product properties and tensor algebra to avoid computing the full covariance matrix and its inverse/determinant during training and avoid computing the full cross covariance during prediction. Instead, we can restrict the kernel matrix computation to be at each mode, and use the tensor-matrix product to combine the results across different modes. In this way, the training and prediction is highly efficient yet we do not need any low rank approximation. The details are provided in (Zhe et al., 2019).

2 Experiment

We evaluated IFHoGP in predicting the solution of several benchmark PDEs, including Poisson's, Heat, and Burgers' equations (Olsen-Kettle, 2011). To collect the training data, we ran the numerical solvers with four meshes — 8×8, 16×16, 32×32 and 64×64 — to generate examples of four fidelities for each PDE. The number of training examples for each fidelity is 200, 100, 50 and 20, respectively. For testing, we generated 128 examples with the highest fidelity. Both the training and testing inputs were uniformly sampled from the domain. The input includes the parameters in the equation, the boundary and/or the initial conditions. We followed (Li et al., 2022a) to map the lowest fidelity to



(a) *Poisson* (b) *Heat* (c) *Burgers*

Figure 1: Normalized Root-Mean-Square Error (nRMSE) in predicting the solution fields of Poisson's, and Heat and Burgers' equations. K is the dimension of the latent output for IFC, DRC and DMF, and the dimension of the coordinate feature vector for IFHoGP.

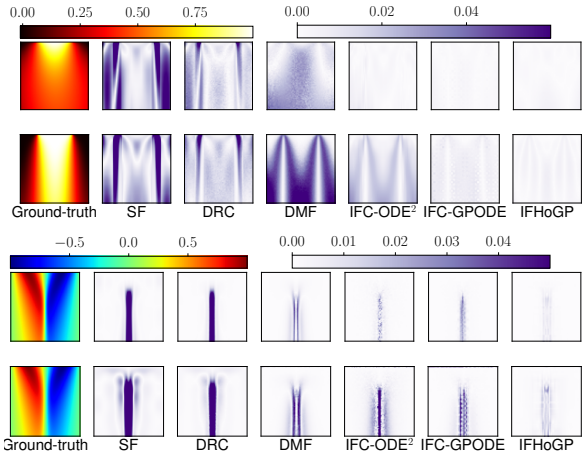
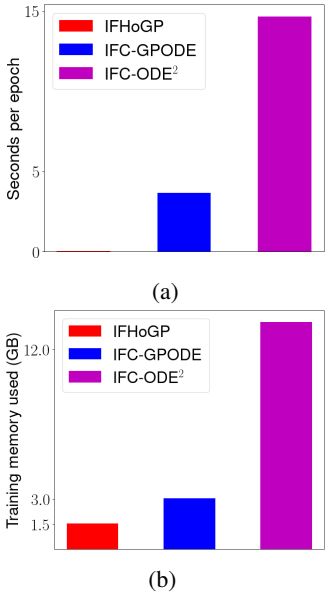


Figure 2: Element-wise solution error. The left most column is the ground-truth and the other columns are the errors of each method. The lighter the color, the smaller the error.



(a)

(b)

Figure 3: Per-epoch running time (in seconds) and GPU memory usage on *Heat* equation with $K = 20$. The evaluation was performed on a single NVIDIA A100 GPU instance.

$m = 0$ and the highest to $m = 1$. The data generation details are the same as (Li et al., 2022a).

We compared with the following state-of-the-art multi-fidelity surrogate learning methods. (1) DRC (Xing et al., 2021a) which estimates Linear Models of Coreginalization (LMC) on the residuals between successive fidelities. The prediction is made by adding the prediction at the lowest fidelity and all the residual predictions from the lowest to the highest fidelities. (2) DMF (Li et al., 2022b), an auto-regressive NN for multi-fidelity learning, where each NN models one fidelity. To capture the nonlinear relationship between successive fidelities, the latent output of each NN is fed into the NN for the next fidelity. (3) IFC-ODE^2 and IFC-GPODE (Li et al., 2022a), which use an neural ODE for continuous-fidelity learning and estimates the basis matrix via an element-wise neural ODE and a GP prior, respectively. (4) SF, the single fidelity degeneration of IFC.

All the methods were implemented by PyTorch (Paszke et al., 2019) except that DRC was implemented by MATLAB. We used the hyper-parameters recommended in the original papers for all the baselines. For our method, we used the Square-Exponential Automatic Relevance Determination (SE-ARD) kernel. We used two layers in the NN transformation in (2), with 40 neurons per layer. We ran ADAM (Kingma & Ba, 2014) to train all the models, except for DRC, it used L-BFGS and the maximum number of iterations was set to 1000. We set the number of ADAM epochs to 500 to ensure convergence for each method. We varied the number of bases — the latent output dimension (or rank) K for IFC, DRC, and DMF, from $\{5, 10, 15, 20\}$. For a fair comparison, we also varied the dimension of the coordinate feature vectors in the same range. For each setting, we repeated the experiment for five times. We report

the average normalized root-mean-square-error (nRMSE) and its standard deviation in Figure 1.

As we can see, on *Poisson's* and *Burgers*' equations, IFHoGP consistently outperforms all the competing methods ($p < 0.05$) except when the $K = 5$, the prediction of IFHoGP is almost identical to IFC-GPODE. On *Heat* equation, the prediction accuracy of IFHoGP overlaps with IFC-GPODE when $K = 10, 15, 20$, and is slightly worse than IFC-ODE^2 yet better than IFC-GPODE when $K = 5$. Overall, IFHoGP almost always shows the best performance. There is only one case that IFHoGP is second to IFC-ODE^2. The gap between IFHoGP and IFC is much smaller as compared with the gap between IFHoGP and the other competing methods. In other words, both IFHoGP and IFC outperform the remaining discrete-fidelity methods by a large margin, which confirms the advantage of continuous-fidelity learning. We then randomly selected two test examples for Heat and Burgers' equation to visualize the point-wise solution error. As shown in Fig. 2, the local prediction error of IFHoGP is much smaller and uniform, reflecting that IFHoGP also better recovers the individual solution outputs.

Finally, we examined the computational efficiency and memory usage of IFHoGP and IFC. As show in Fig. 3, IFHoGP is 380x and 95x faster than IFC-GPODE and IFC-ODE^2, respectively. In the meantime, IFHoGP uses only 1/2 and 1/8 GPU memory during the training, as compared with IFC-GPODE and IFC-ODE^2, respectively. Together the results demonstrate that the training of IFHoGP is much more efficient and economic, while still achieving better or the same predictive performance.

3 Conclusion

We have presented IFHoGP, an alternative infinite-fidelity learning approach for physical simulation. IFHoGP is based on GPs, and avoids using the neural ODE to model the complex relationships within the continuous fidelity space. IFHoGP is efficient in both computation and memory usage, while showing better or close predictive performance as compared with IFC. In the future, we will continuous the evaluation of IFHoGP in different problems, not only in prediction accuracy but also in uncertainty quantification.

4 Broader Impact

This work can be used in a variety of engineering design problems that involve intensive computation, *e.g.*, finite elements or differences. Hence, the work has potential positive impacts in the society if it is used to design passenger aircrafts, biomedical devices, automobiles, and all the other devices or machines that can benefit human lives. At the same time, this work may have some negative consequences if it is used to design weapons or weapon parts.

References

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. Neural ordinary differential equations. Advances in neural information processing systems, 31, 2018.

Dandekar, R., Chung, K., Dixit, V., Tarek, M., Garcia-Valadez, A., Vemula, K. V., and Rackauckas, C. Bayesian neural ordinary differential equations. arXiv preprint arXiv:2012.07244, 2020.

Gholaminejad, A., Keutzer, K., and Biros, G. ANODE: Unconditionally accurate memory-efficient gradients for neural ODEs. In International Joint Conferences on Artificial Intelligence, 2019.

Kandasamy, K., Dasarathy, G., Schneider, J., and Póczos, B. Multi-fidelity bayesian optimisation with continuous approximations. In International Conference on Machine Learning, pp. 1799–1808. PMLR, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Li, S., Wang, Z., Kirby, R., and Zhe, S. Infinite-fidelity coregionalization for physical simulation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), Advances in Neural Information Processing Systems, 2022a. URL https://openreview.net/forum?id=dUYLikScE-.

Li, S., Wang, Z., Kirby, R. M., and Zhe, S. Deep multi-fidelity active learning of high-dimensional outputs. Proceedings of the Twenty-Fifth International Conference on Artificial Intelligence and Statistics, 2022b.

Olsen-Kettle, L. Numerical solution of partial differential equations. Lecture notes at University of Queensland, Australia, 2011.

Parussini, L., Venturi, D., Perdikaris, P., and Karniadakis, G. E. Multi-fidelity gaussian process regression for prediction of random fields. Journal of Computational Physics, 336:36–50, 2017.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32, 2019.

Wang, Z., Xing, W., Kirby, R., and Zhe, S. Multi-fidelity high-order Gaussian processes for physical simulation. In International Conference on Artificial Intelligence and Statistics, pp. 847–855. PMLR, 2021.

Wu, J. and Frazier, P. I. Continuous-fidelity bayesian optimization with knowledge gradient. 2018.

Xing, W. W., Kirby, R. M., and Zhe, S. Deep coregionalization for the emulation of simulation-based spatial-temporal fields. Journal of Computational Physics, 428: 109984, 2021a.

Xing, W. W., Shah, A. A., Wang, P., Zhe, S., Fu, Q., and Kirby, R. M. Residual Gaussian process: A tractable nonparametric Bayesian emulator for multi-fidelity simulations. Applied Mathematical Modelling, 97:36–56, 2021b.

Zhe, S., Xing, W., and Kirby, R. M. Scalable high-order gaussian process regression. In The 22nd International Conference on Artificial Intelligence and Statistics, pp. 2611–2620, 2019.