Infinite-Fidelity Surrogate Learning via High-order Gaussian Processes

Shibo Li[1], *Li Shi*[2], *Shandian Zhe*[1]
[1]*Kahlert School of Computing, The University of Utah*
[2]*The English Language Institute , The University of Utah*

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 7/8 memory reduction. Meanwhile, uncertainty calibration for IF-HOGP is straightforward.

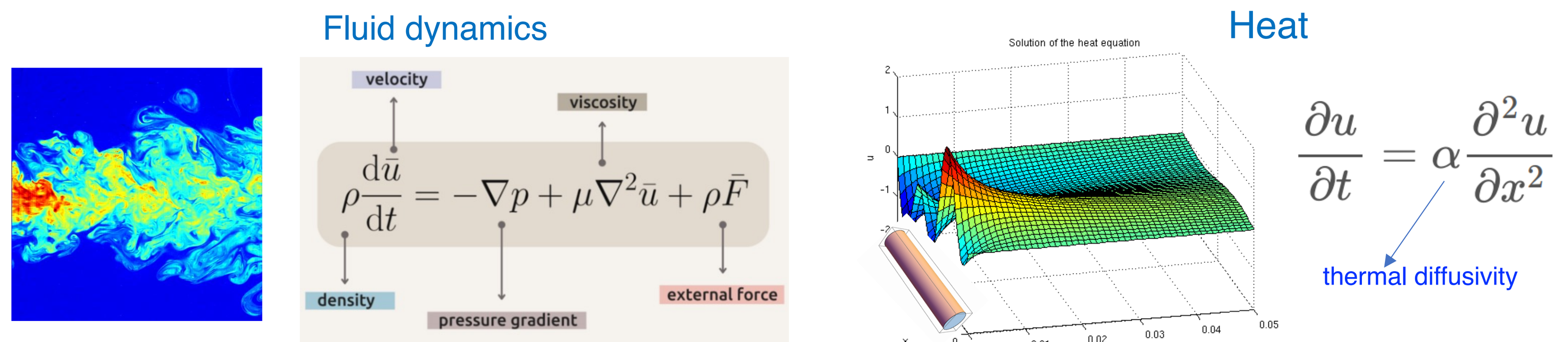
Introduction & Motivation

Physical Simulations by Solving PDEs:

- Daily phenomena are dominated by fundamental *physical laws*
- Fundamental physical laws are written as *Partial differential equations (PDEs) systems*

Key challenges of scientific computing:

- *Identify* the governing model for complex systems
- Efficiently *solving* large-scale non-linear systems of equations



Numerical Solvers:

- (Exact) accurate yet solutions
- Slow
- Do not generalize over the same domain

Data-drive Solvers(Surrogate Learning, Operator Learning):

- Fast inference with new PDE
- Generalize over domain problems
- Prepare large amount of data from numerical solvers

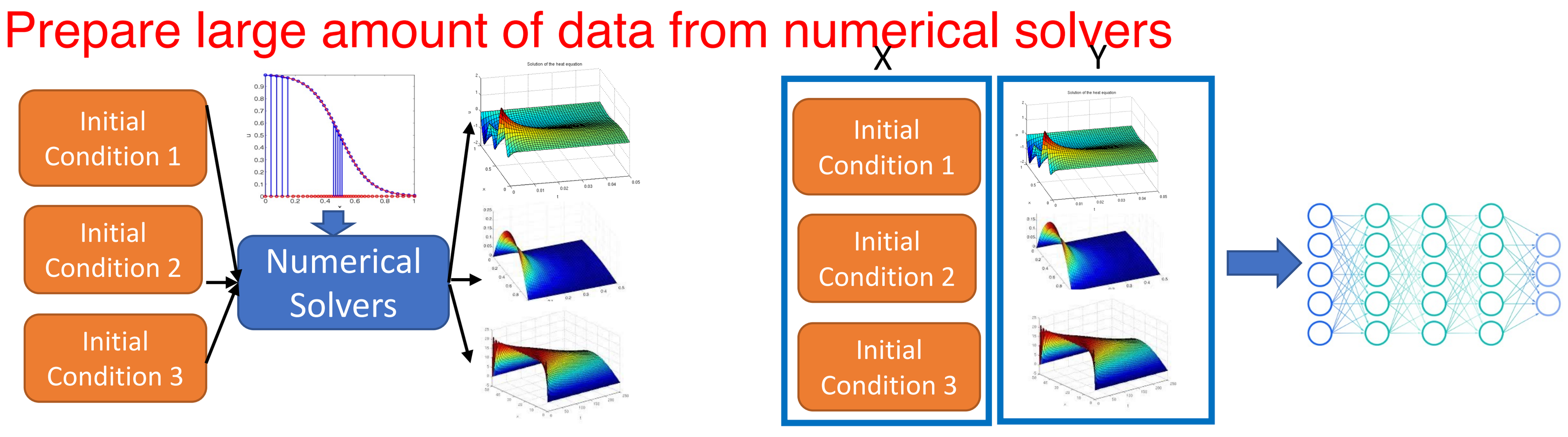
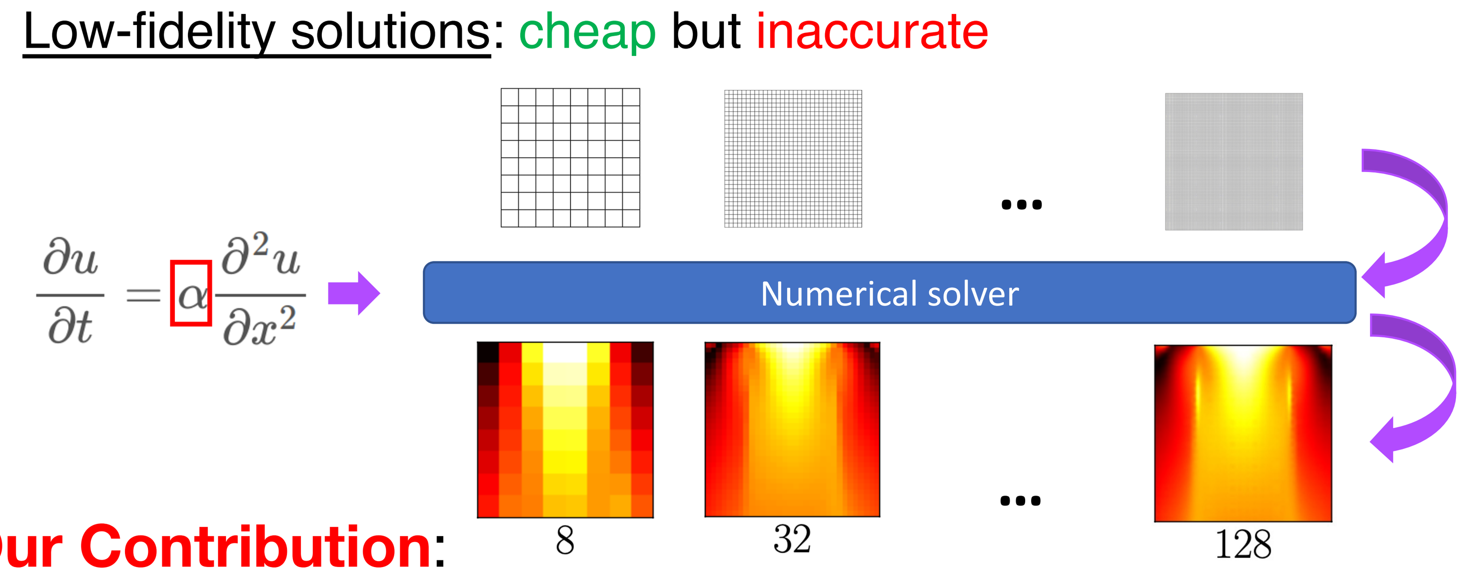


Figure 1. Conventional PDEs solvers

Figure 2. Data-driven PDEs solvers

Multi-fidelity Modeling:

- Conventional solvers usually have multi-fidelity evaluations natively
- High-fidelity solutions: expensive but accurate
- Low-fidelity solutions: cheap but inaccurate



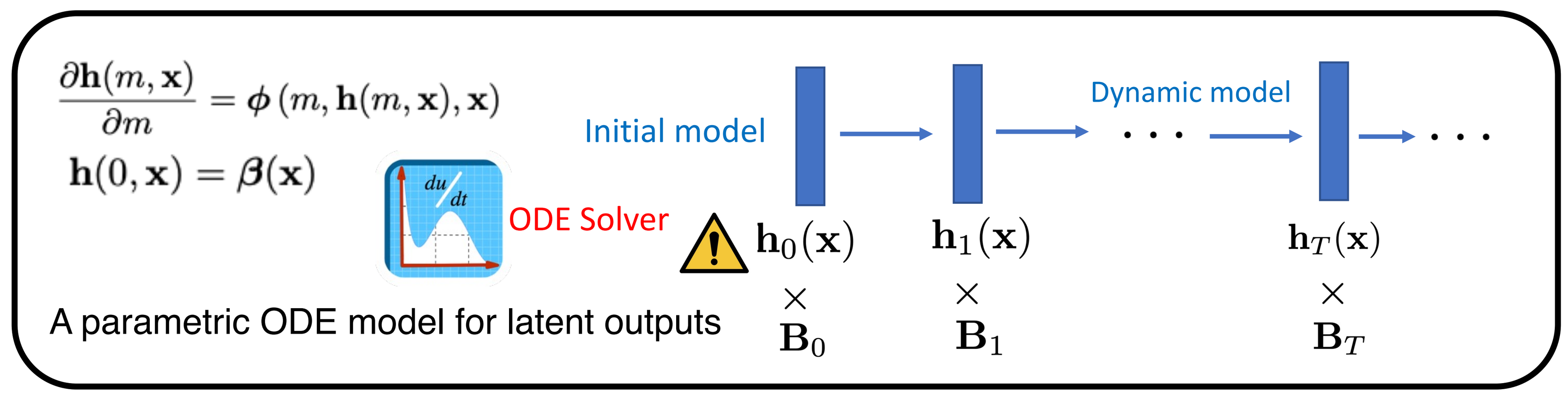
Our Contribution:

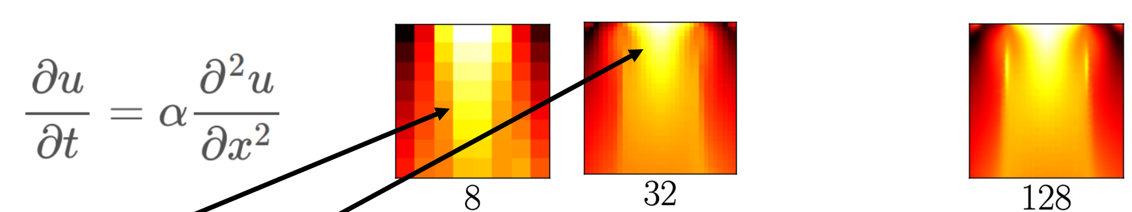
- ***IF-HoGP***: A ***scalable*** multi-fidelity modeling for ***very high-dimensional*** outputs
- Flexibly handles **infinity/continuous** fidelities while captures all fidelities' nonlinear, non-stationary correlations.
- Predicts on **unseen** fidelities.
- ***Uncertainty quantification*** on PDE solutions.

	DRC (Xing et al 2021)	MFHoGP (Wang et al. 2021)	DMF (Li et al. 2022)	IFC (Li et al. 2022)	IF-HoGP (Ours)
Multi-Fidelity Modeling ?	✓	✓	✓	✓	✓
Capture Complex Fidelities Correlations?	✓	✓	✓	✓	✓
Models **all** fidelities?	✗	✗	✗	✓	✓
Predict on **unseen** fidelities?	✗	✗	✗	✓	✓
Extrapolate to **higher than target** fidelities?	✗	✗	✗	✓	✓
UQ on Solutions?	✗	✓	✓	✗	✓
Efficiency and Scalability?	☺	☺	☺	☹	☺

Methods

Infinite-Fidelity Coregionalization (IFC)

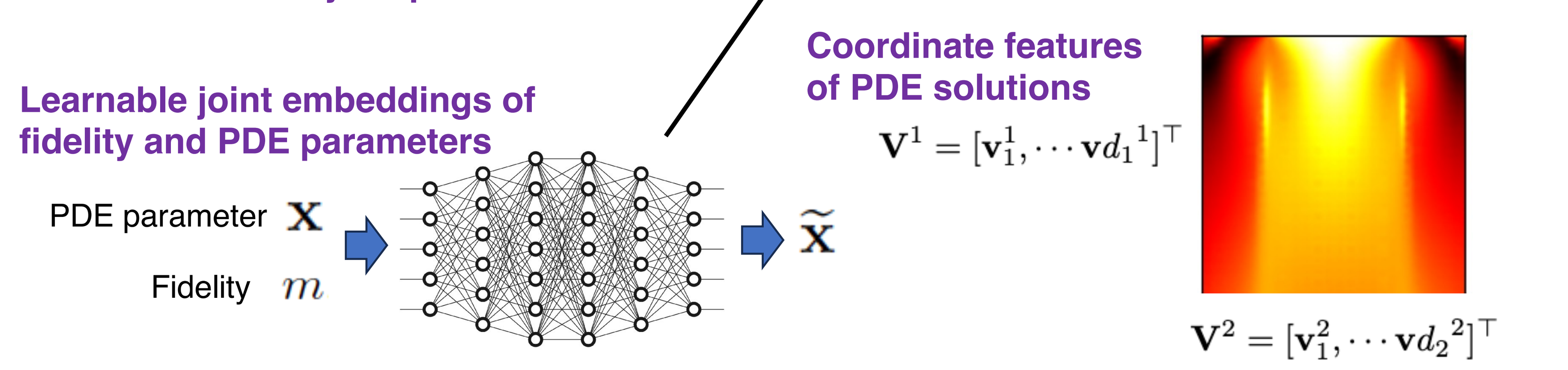




IFHoGP (Ours)

$$\kappa(y_{\mathbf{i}}(m,\mathbf{x}), y_{\mathbf{j}}(m',\mathbf{x}')) = \kappa(\mathrm{NN}(m,\mathbf{x}), \mathrm{NN}(m',\mathbf{x}')) \cdot \prod_{k=1}^{s} \kappa(\mathbf{v}^k_{i_k}, \mathbf{v}^k_{j_k})$$

Product kernel to model the multi-fidelity outputs



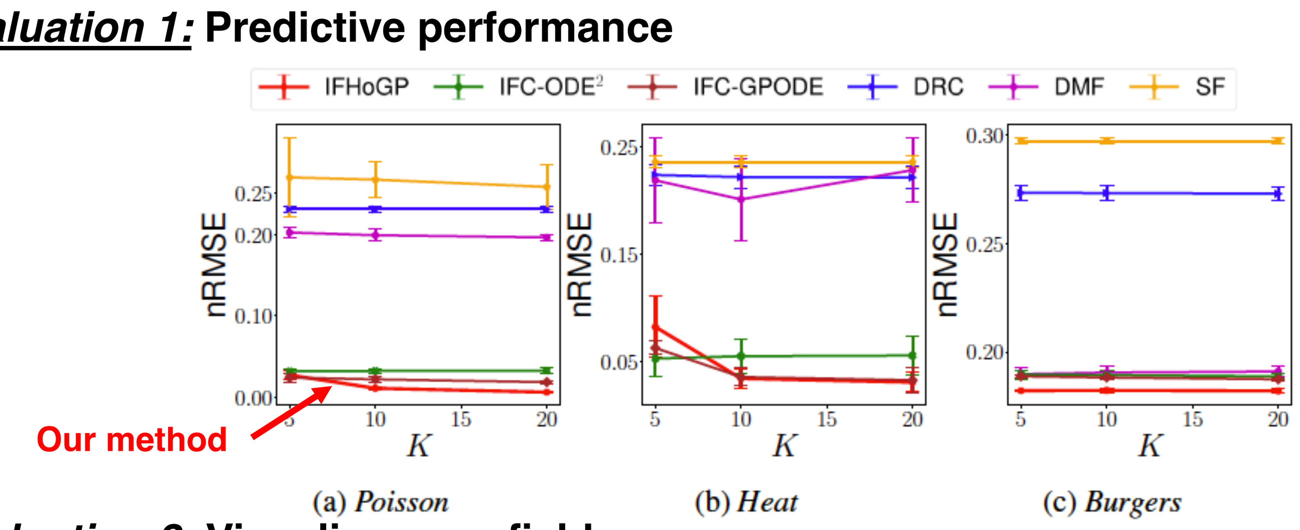
Optimal parameters can be acquired by maximizing marginal likelihood

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

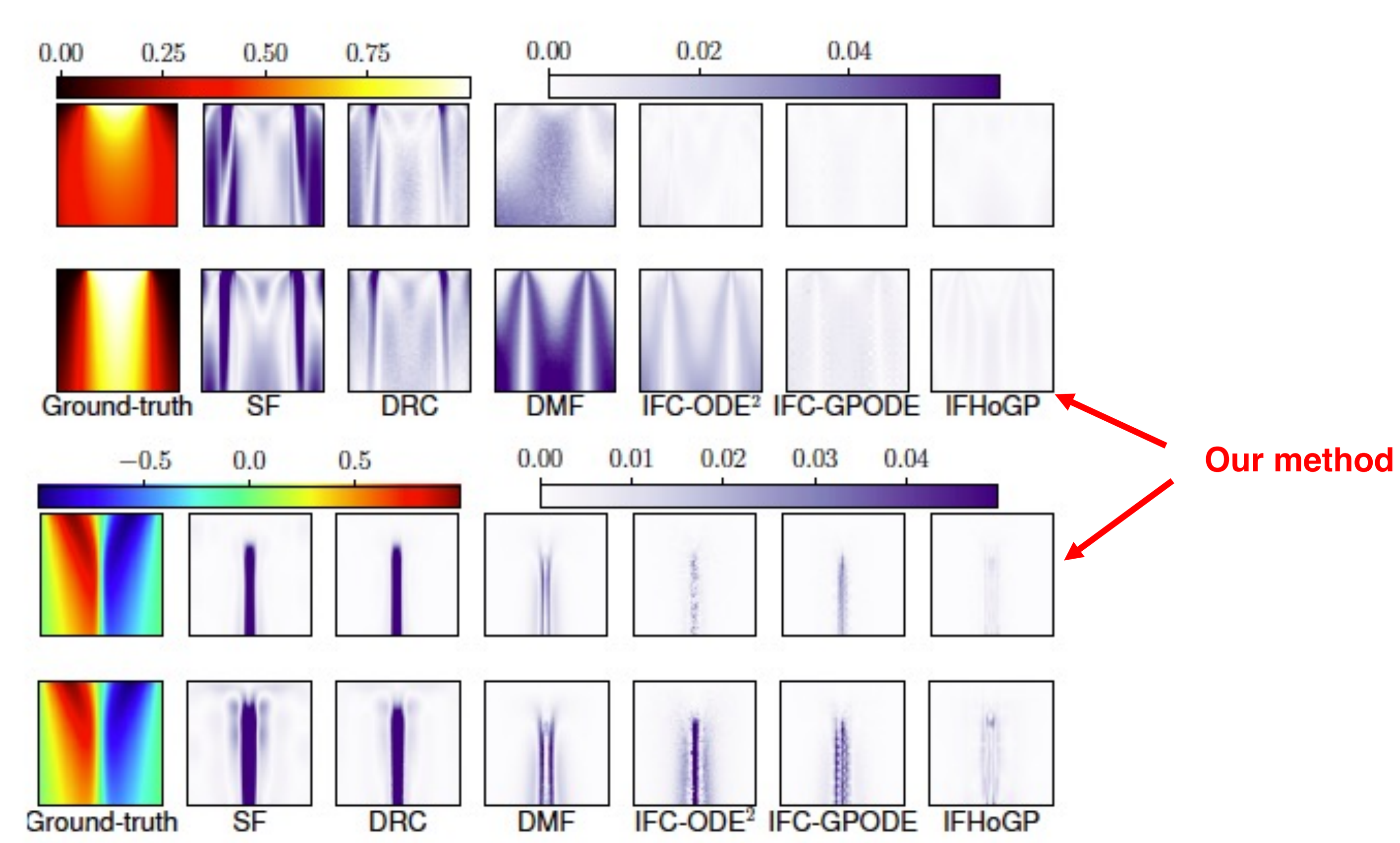
Can be efficiently computed by exploring the Kronecker structure

Experiments

Evaluation 1: **Predictive performance**

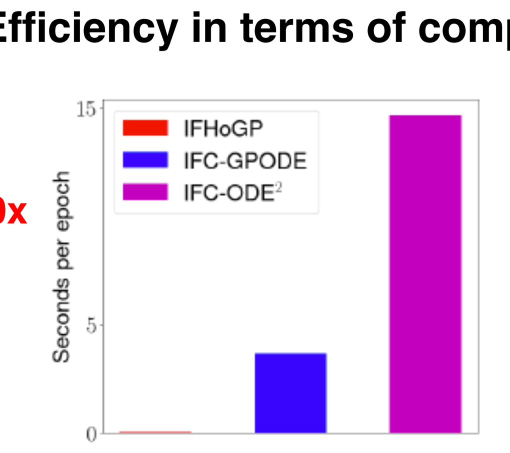


Evaluation 2: **Visualize error fields**



Evaluation 3: **Efficiency in terms of computation reduction compared with IFC**

Training Time: 380x



Memory: 8x