Multi-Resolution Active Learning of Fourier Neural Operators

Shibo Li [1], Xin Yu [1], Wei Xing [4], Robert M. Kirby [1,2], Akil Narayan [2,3], Shandian Zhe [1]

[1] *Kahlert School of Computing, University of Utah*
[2] *Scientific Computing and Imaging (SCI) Institute, University of Utah*
[3] *Department of Mathematics, University of Utah*
[4] *School of Mathematics and Statistics, University of Sheffield*

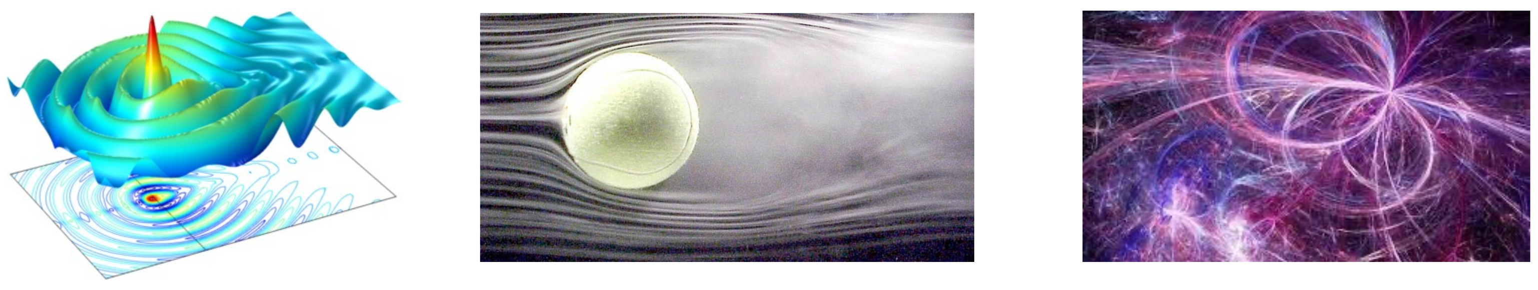
THE UNIVERSITY OF UTAH®
www.sci.utah.edu

Abstract: Fourier Neural Operator (FNO) is a popular operator learning framework. It not only achieves the state-of-the-art performance in many tasks, but also is efficient in training and prediction. However, collecting training data for the FNO can be a costly bottleneck in practice, because it often demands expensive physical simulations. To overcome this problem, we propose ***Multi-Resolution Active learning of FNO (MRA-FNO)***, which can dynamically select the input functions and resolutions to lower the data cost as much as possible while optimizing the learning efficiency. Specifically, we propose a probabilistic multi-resolution FNO and use *ensemble Monte-Carlo* to develop an effective posterior inference algorithm. To conduct active learning, we maximize a utility-cost ratio as the acquisition function to acquire new examples and resolutions at each step. We use moment matching and the matrix determinant lemma to enable tractable, efficient utility computation. Furthermore, we develop a *cost annealing framework to avoid over-penalizing high-resolution queries* at the early stage. The over-penalization is severe when the cost difference is significant between the resolutions, which renders active learning often stuck at low-resolution queries and inferior performance. Our method overcomes this problem and applies to general multi-fidelity active learning and optimization problems. We have shown the advantage of our method in several benchmark operator learning tasks.

Introduction & Motivation

Physics *vs.* Machine Learning

- *Physics:* Accurate, Principled, Extrapolate Well
- *Machine Learning:* Flexible, Efficient



Wave — $\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$

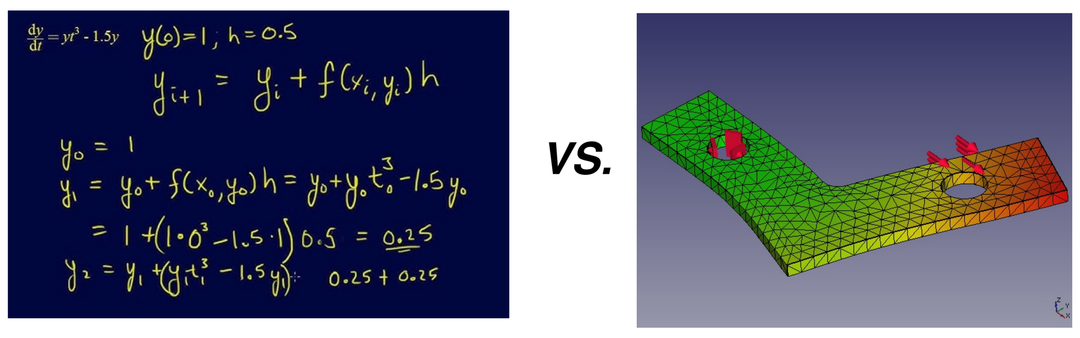
Fluid Dynamics — $\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v}\cdot\nabla)\mathbf{v} = \nu\nabla^2\mathbf{v} - \nabla P$

Electromagnetism — $\nabla\cdot\mathbf{D} = \rho$, $\nabla\cdot\mathbf{B} = 0$, $\nabla\times\mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}$, $\nabla\times\mathbf{H} = \mathbf{J} + \frac{\partial \mathbf{D}}{\partial t}$

Computational Physics: Finding the solutions of PDEs

- *Analytical solutions:* closed forms but not always feasible
- *Numerical solutions:* accurate, reliable but slow and no generalization

Analytical **VS.** Numerical (*e.g.*, FEM)



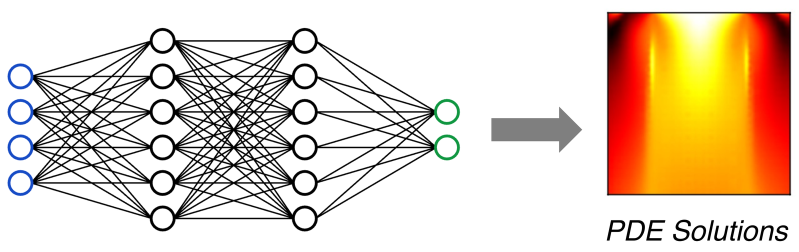
Data-Driven Computational Physics

- Learning PDE solutions directly from data
- Efficient prediction and generalization

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}$$

Heat Equation

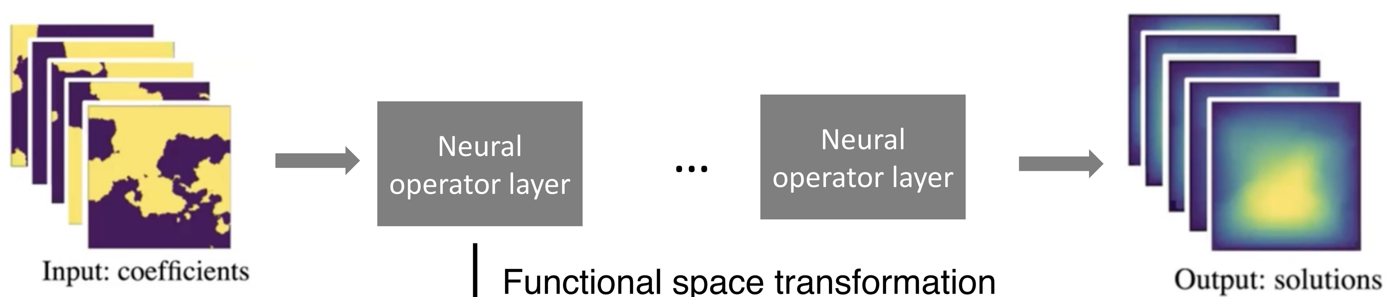
- Initial condition
- Boundary condition
- PDE coefficients



PDE Solutions

Neural Operators and *Fourier Neural Operators*

- Solutions of PDEs are applying operators to the source/parameter/initial functions
- Learning the parametric mapping $\mathcal{G}_\theta : \mathcal{A} \to \mathcal{U}$ between functions to functions

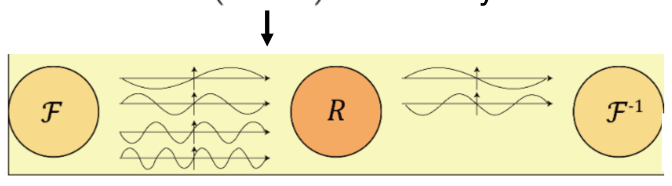


Functional space transformation

$$\sigma\left(W v_t(x) + \int_{B(x,r)} \kappa_\phi(x, y, a(x), a(y)) v_t(y)\, dy\right)$$ Function convolution

Fourier Neural Operators (FNO)

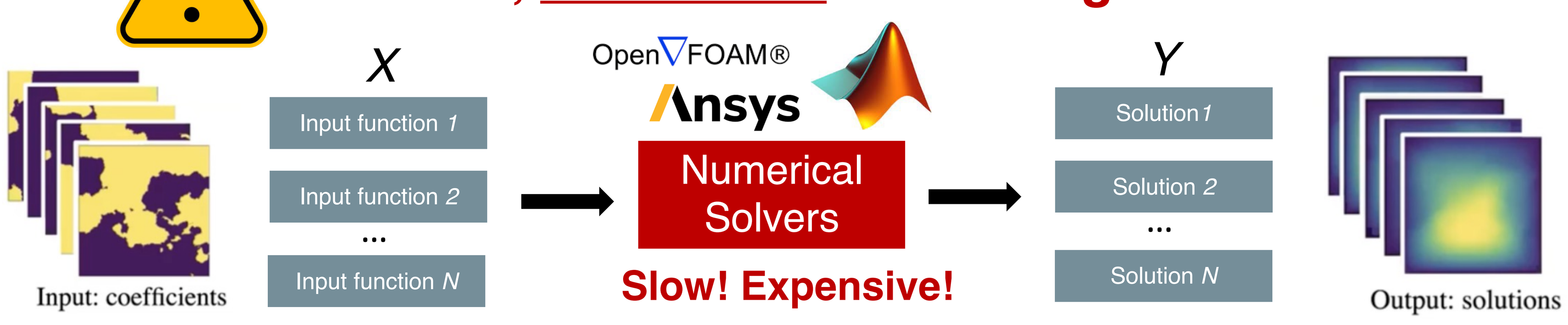
$\kappa(\mathbf{x}-\mathbf{x}')$ Stationary kernel functions



Convolutional Theorem

$$\int \kappa(\mathbf{x}-\mathbf{x}')v(\mathbf{x}')d\mathbf{x}' = \mathcal{F}^{-1}\left[\mathcal{F}[\kappa]\cdot\mathcal{F}[v]\right]$$

However, *Data Costs* for Training FNO!!!



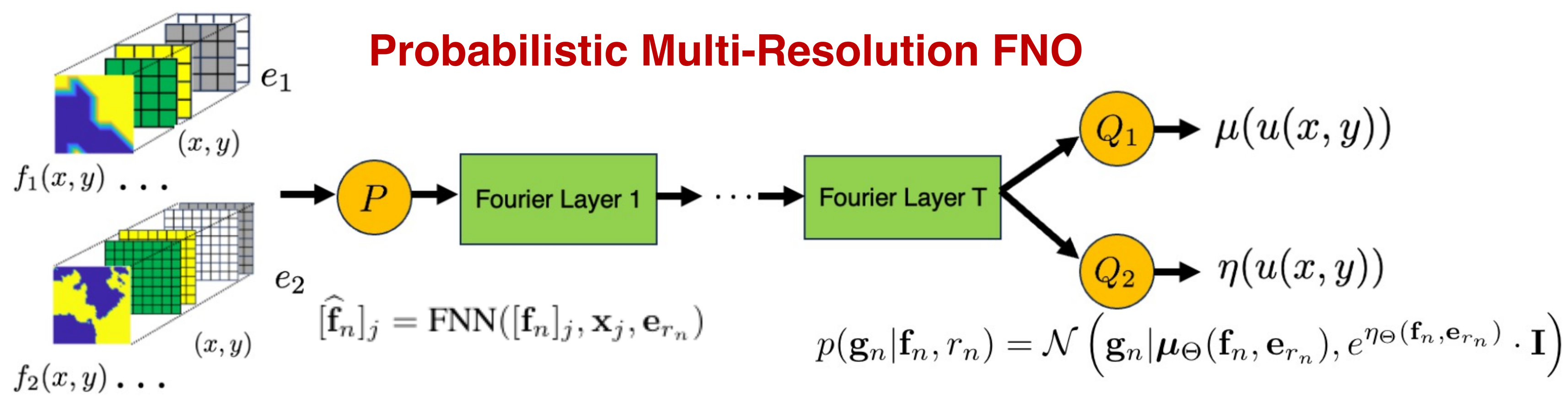
Slow! Expensive!

Our Contributions

- *Probabilistic* Multi-Resolution FNO
- *Active Learning* for FNO
- Robust performance via *cost-annealing*

Methods

Probabilistic Multi-Resolution FNO

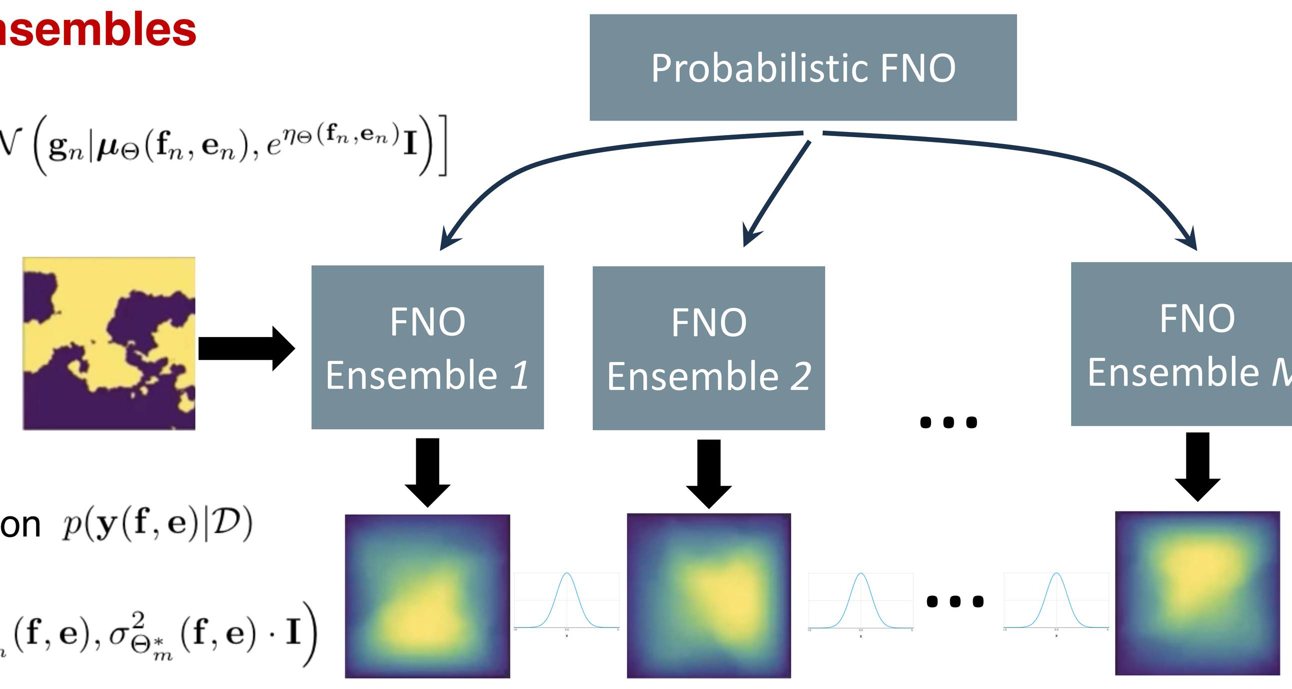


$$[\hat{\mathbf{f}}_n]_j = \mathrm{FNN}([\mathbf{f}_n]_j, \mathbf{x}_j, \mathbf{e}_{r_n})$$

$$p(\mathbf{g}_n|\mathbf{f}_n, r_n) = \mathcal{N}\left(\mathbf{g}_n|\boldsymbol{\mu}_\Theta(\mathbf{f}_n, \mathbf{e}_{r_n}), e^{\eta_\Theta(\mathbf{f}_n, \mathbf{e}_{r_n})}\cdot\mathbf{I}\right)$$

Monte-Carlo Ensembles

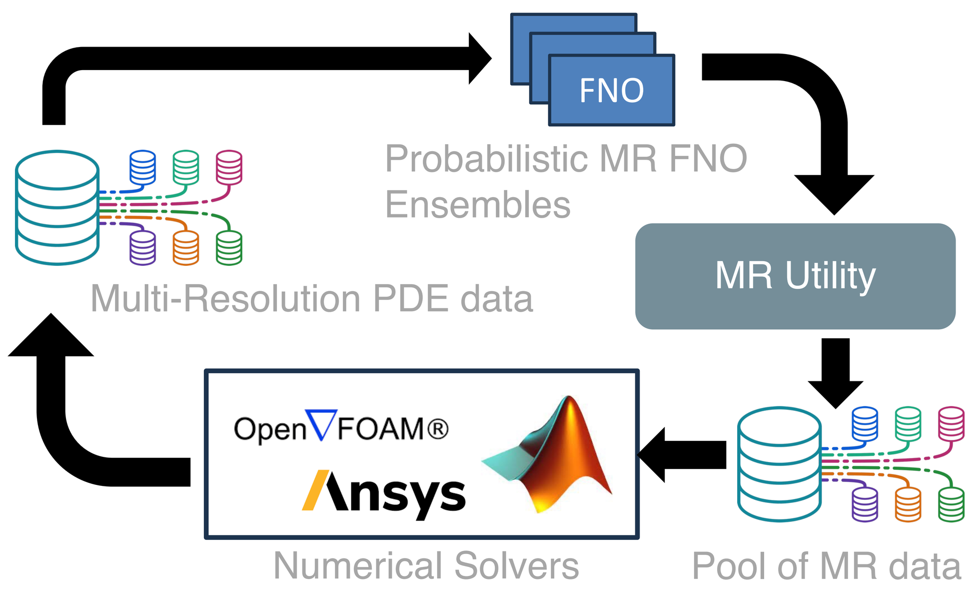
$$\Theta^* = \operatorname*{argmax}_\Theta \sum_{n=1}^N \log\left[\mathcal{N}\left(\mathbf{g}_n|\boldsymbol{\mu}_\Theta(\mathbf{f}_n, \mathbf{e}_n), e^{\eta_\Theta(\mathbf{f}_n,\mathbf{e}_n)}\mathbf{I}\right)\right]$$



Predictive distribution $p(\mathbf{y}(\mathbf{f},\mathbf{e})|\mathcal{D})$

$$= \frac{1}{M}\sum_{m=1}^M \mathcal{N}\left(\mathbf{y}|\boldsymbol{\mu}_{\Theta_m^*}(\mathbf{f},\mathbf{e}), \sigma^2_{\Theta_m^*}(\mathbf{f},\mathbf{e})\cdot\mathbf{I}\right)$$

Multi-Resolution Active Learning



Multi-resolution acquisition function

$$u(h, r) = \mathbb{I}(\mathbf{y}(\mathbf{h}^r, \mathbf{e}_r), \mathbf{y}(\mathbf{h}^R, \mathbf{e}_R)|\mathcal{D})$$

Step1: Monte-Carlo approximation

$$\hat{u}(h, r) = \frac{1}{A}\sum_{l=1}^A \mathbb{I}(\mathbf{y}(\mathbf{h}^r, \mathbf{e}_r), \mathbf{y}(\mathbf{h}_l^R, \mathbf{e}_R)|\mathcal{D})$$

$$\mathbb{I}(\mathbf{y}_1, \mathbf{y}_2|\mathcal{D}) = \mathbb{H}(\mathbf{y}_1|\mathcal{D}) + \mathbb{H}(\mathbf{y}_2|\mathcal{D}) - \mathbb{H}(\mathbf{y}_1, \mathbf{y}_2|\mathcal{D})$$

Step 2: Predictive distribution via moment matching

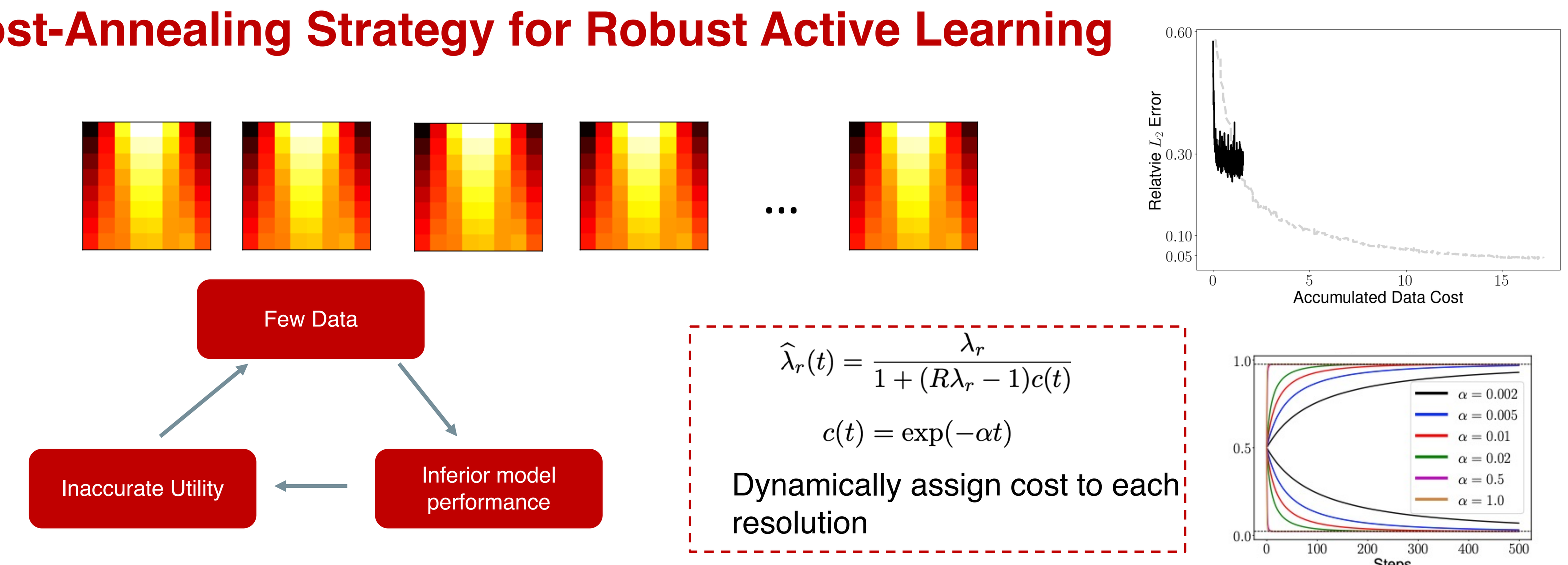
$$\mathbb{E}(\widehat{\mathbf{y}}|\mathcal{D}) = \frac{1}{M}\sum_{m=1}^M \boldsymbol{\rho}_m,$$

$$\mathrm{cov}(\widehat{\mathbf{y}}|\mathcal{D}) = \frac{1}{M}\sum_{m=1}^M (\boldsymbol{\Lambda}_m + \boldsymbol{\rho}_m\boldsymbol{\rho}_m^\top) - \mathbb{E}(\widehat{\mathbf{y}}|\mathcal{D})\mathbb{E}(\widehat{\mathbf{y}}|\mathcal{D})^\top$$

Step 3: Efficient computation via Weinstein-Aronszajn Identity

$$\log\det[\mathrm{cov}(\widehat{\mathbf{y}}|\mathcal{D})] = \log\det[\boldsymbol{\Lambda} + \mathbf{B}\mathbf{B}^\top]$$
$$= \log\det[\boldsymbol{\Lambda}] + \log\det[\mathbf{I} + \mathbf{B}^\top\boldsymbol{\Lambda}^{-1}\mathbf{B}].$$

Cost-Annealing Strategy for Robust Active Learning



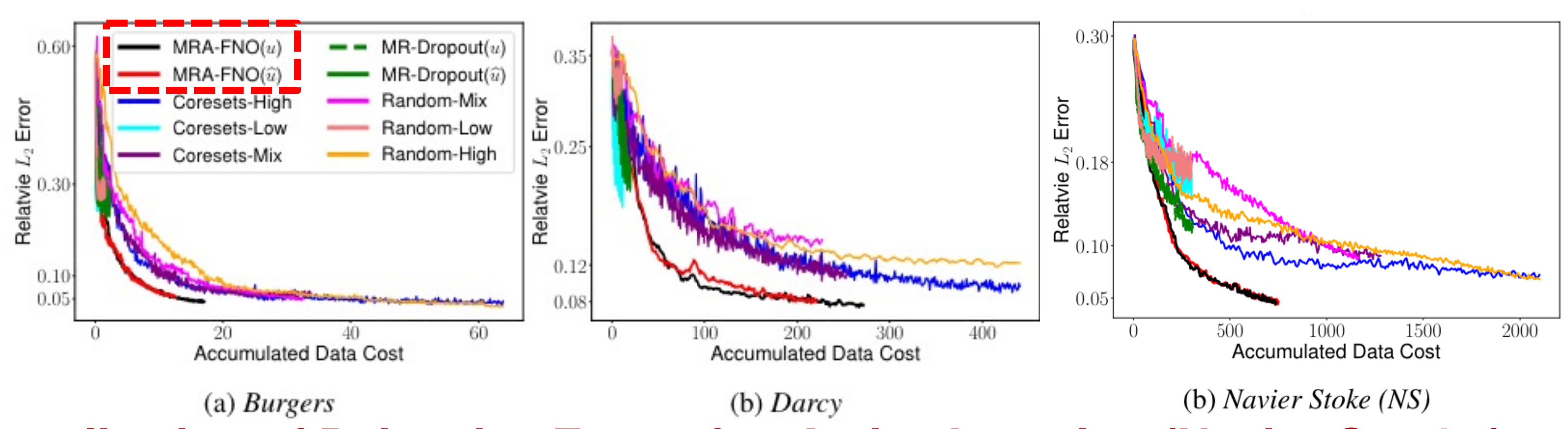
$$\widehat{\lambda}_r(t) = \frac{\lambda_r}{1 + (R\lambda_r - 1)c(t)}$$

$$c(t) = \exp(-\alpha t)$$

Dynamically assign cost to each resolution

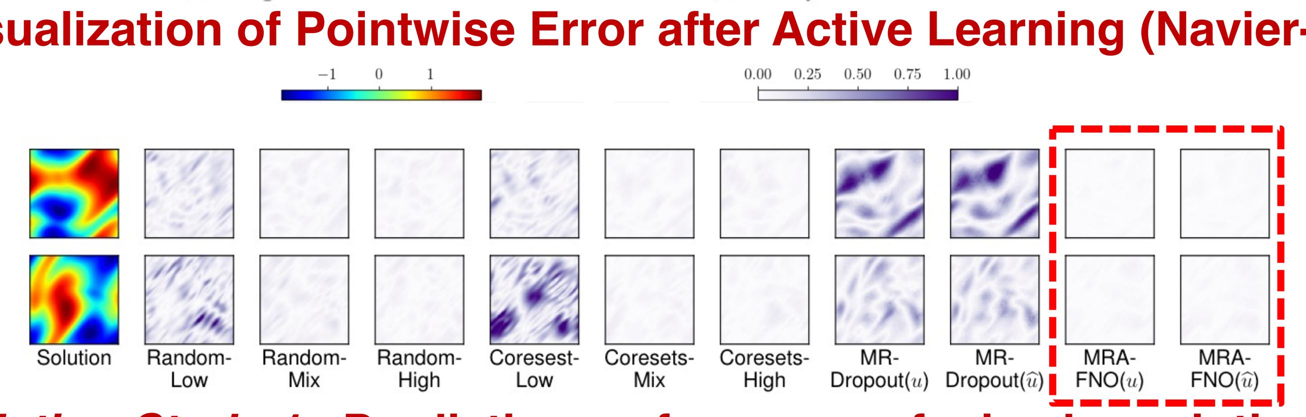
Experiments

Evaluations of Multi-Resolution Active Learning



(a) *Burgers* (b) *Darcy* (b) *Navier Stoke (NS)*

Visualization of Pointwise Error after Active Learning (Navier-Stockes)



Ablation Study 1: Predictive performance of mixed-resolutions examples vs. low-resolutions only examples

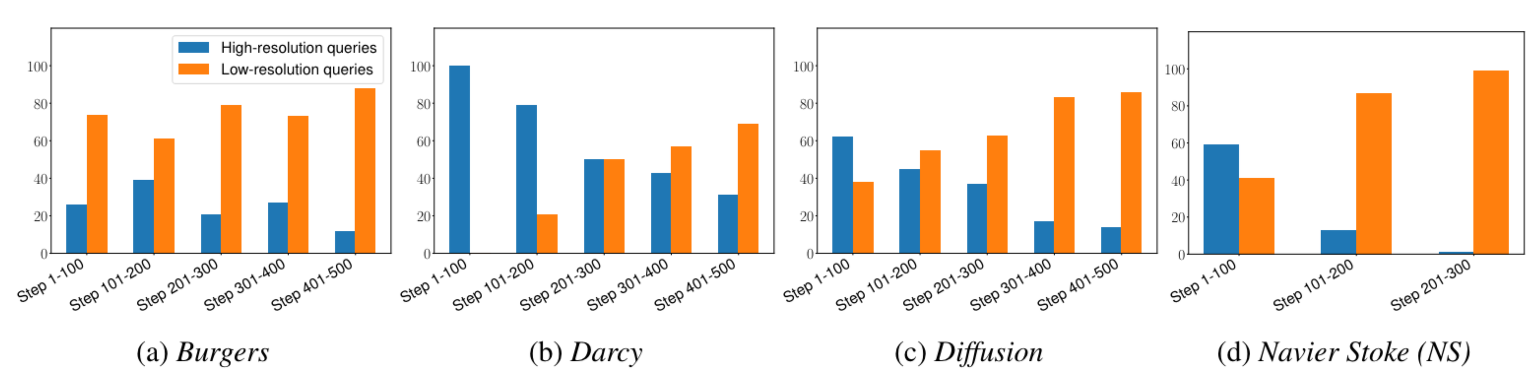
Method	Burgers	Darcy
FNO	**0.0575 ± 0.0031**	0.0891 ± 0.0078
FNO-Dropout	0.0791 ± 0.0035	0.1038 ± 0.0056
FNO-SGLD	0.0804 ± 0.0049	0.0933 ± 0.0074
FNO-SVI	0.1182 ± 0.0056	0.0946 ± 0.0041
MRA-FNO	**0.0586 ± 0.0042**	**0.0876 ± 0.0059**

(a) Relative L_2 error

Method	Burgers	Darcy
FNO	296.31%	218.54%
FNO-Dropout	331.64%	233.04%
FNO-SGLD	200.62%	183.60%
FNO-SVI	111.14%	188.16%
MRA-FNO	286.86%	201.71%

(c) Relative L_2 error increase

Ablation Study 2: Queried Resolutions Statistics via Cost-Annealing



(a) *Burgers* (b) *Darcy* (c) *Diffusion* (d) *Navier Stoke (NS)*