Batch Multi-Fidelity Active Learning with Budget Constraints

Shibo Li*, Jeff M. Phillips*, Xin Yu, Robert Mike Kirby, Shandian Zhe
School of Computing, Scientific Computing and Imaging Institute; University of Utah
{shibo, jeffp, xiny, kirby, zhe}@cs.utah.edu

THE UNIVERSITY OF UTAH

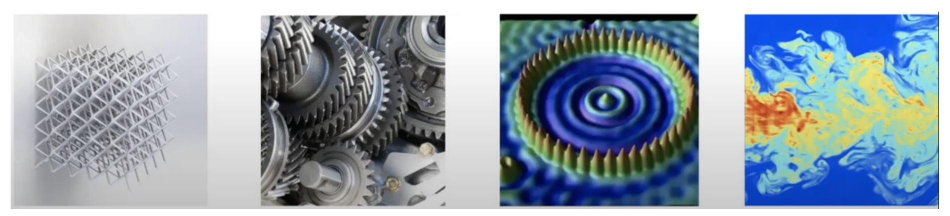
www.sci.utah.edu

Abstract

Learning functions with high-dimensional outputs is critical in many applications, such as physical simulation and engineering design. However, collecting training examples for these applications is often costly, *e.g.*, by running numerical solvers. The recent work (Li et al., 2022) proposes the first multi-fidelity active learning approach for high-dimensional outputs, which can acquire examples at different fidelities to reduce the cost while improving the learning performance. However, this method only queries at one pair of fidelity and input at a time, and hence has a risk to bring in strongly correlated examples to reduce the learning efficiency. In this paper, we propose Batch Multi-Fidelity Active Learning with Budget Constraints (BMFAL-BC), which can promote the diversity of training examples to improve the benefit-cost ratio, while respecting a given budget constraint for batch queries. Hence, our method can be more practically useful. Specifically, we propose a novel batch acquisition function that measures the mutual information between a batch of multi-fidelity queries and the target function, so as to penalize highly correlated queries and encourages diversity. The optimization of the batch acquisition function is challenging in that it involves a combinatorial search over many fidelities while subject to the budget constraint. To address this challenge, we develop a weighted greedy algorithm that can sequentially identify each (fidelity, input) pair, while achieving a near (1 - 1/e)-approximation of the optimum. We show the advantage of our method in several computational physics and engineering applications.

Motivation

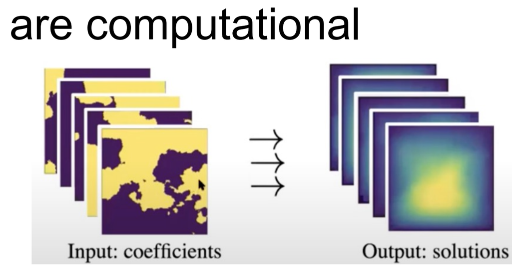
PDEs: "Differential equations... represent the most powerful tool humanity has ever created for making sense of the material world." (Strogatz 2009).



Zongyi Li et al. 2020

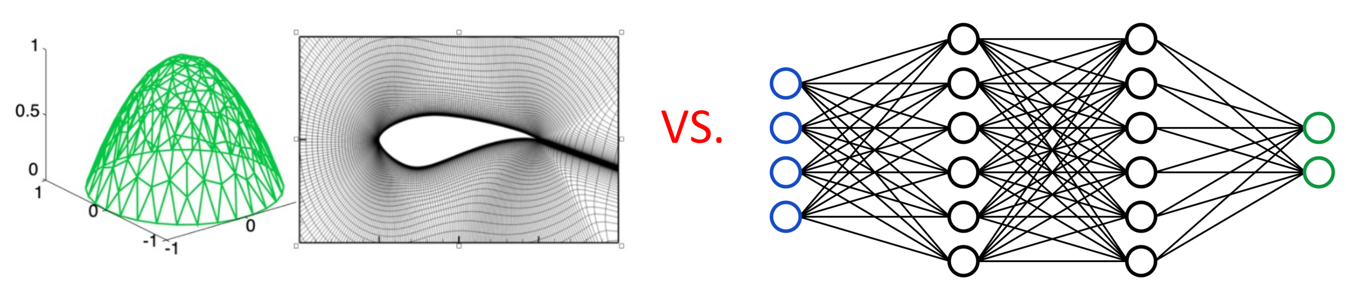
System Identification:
- Requires extensive prior knowledge in the corresponding field
- Ex: modeling the deformation and failure of solid structure requires detailed knowledge of the relationship between stress and strain in the constituent material

Solving Complicated PDEs:
- Ex: those arising from turbulence and plasticity are computational demanding and intractable
- Numerical solvers vs. data driven solvers

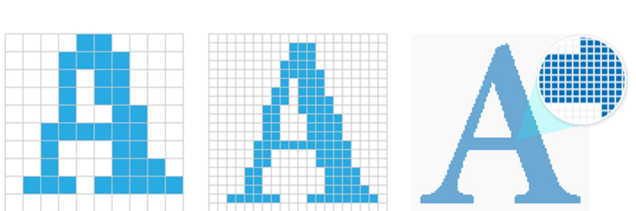


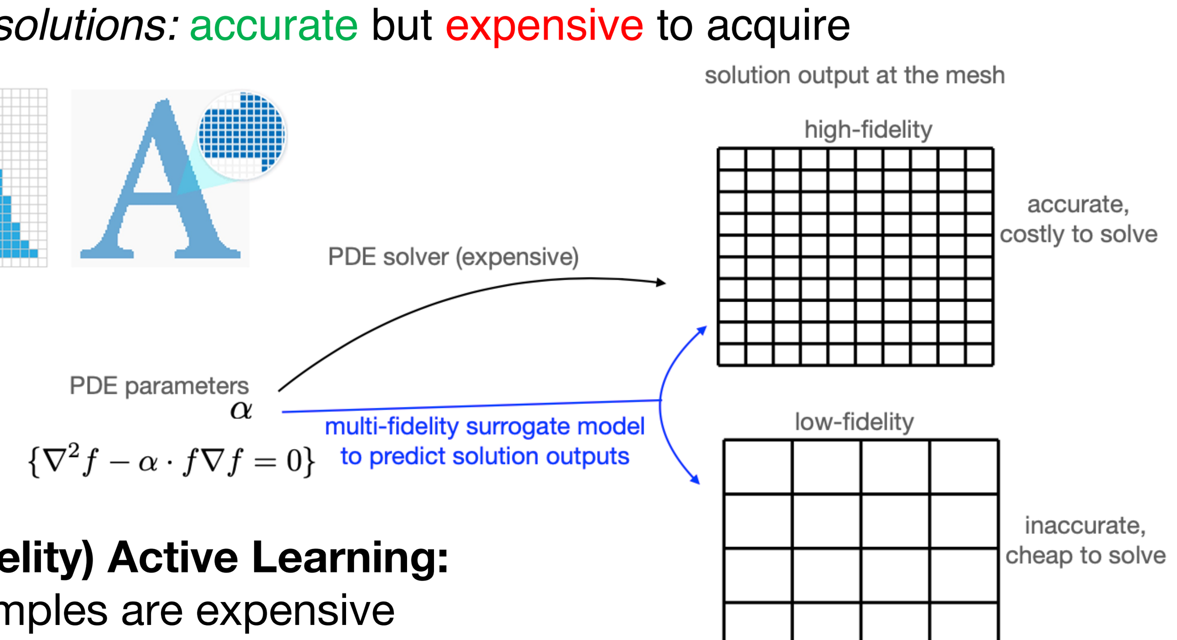
Zongyi Li et al. 2020

Solve vs. Learn:
- Solving PDEs are slow and one instance only
- Learn a family of solutions slow to train but fast to evaluate



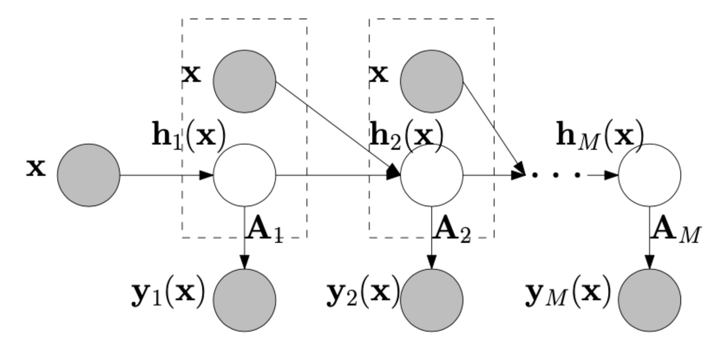
Intuition of Multi-Fidelity Learning :
- Numerical Solvers are fast on coarse grid and slow on fine grid which implies
- *Low-fidelity solutions:* cheap to acquire but inaccurate
- *High-fidelity solutions:* accurate but expensive to acquire





Why (Multi-Fidelity) Active Learning:
- Training examples are expensive
- Leverage the cost-benefit ratio

Deep Multi-Fidelity Active Learning:
- Training examples are expensive
- Leverage the cost-benefit ratio



Concatenate inputs — Low-rank outputs

$$\boldsymbol{\xi}_m = [\mathbf{x}; \mathbf{h}_{m-1}(\mathbf{x})], \quad \mathbf{h}_m(\mathbf{x}) = \mathbf{W}_m \boldsymbol{\rho}_{\boldsymbol{\theta}_m}(\boldsymbol{\xi}_m)$$
$$\mathbf{y}_m(\mathbf{x}) = \mathbf{A}_m \mathbf{h}_m(\mathbf{x}) + \boldsymbol{\epsilon}_m,$$

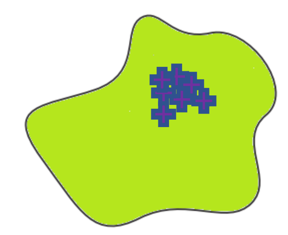
Outputs in solution space

MF Modeling

$$a(\mathbf{x}, m) = \frac{1}{\lambda_m} \mathbb{I}\left(\mathbf{y}_m(\mathbf{x}), \mathbf{y}_M(\mathbf{x}) | \mathcal{D}\right)$$
$$= \frac{1}{\lambda_m}\left(\mathbb{H}(\mathbf{y}_m|\mathcal{D}) + \mathbb{H}(\mathbf{y}_M|\mathcal{D}) - \mathbb{H}(\mathbf{y}_m, \mathbf{y}_M|\mathcal{D})\right)$$

MF Acquisition

Ignoring the correlations between the successive queries

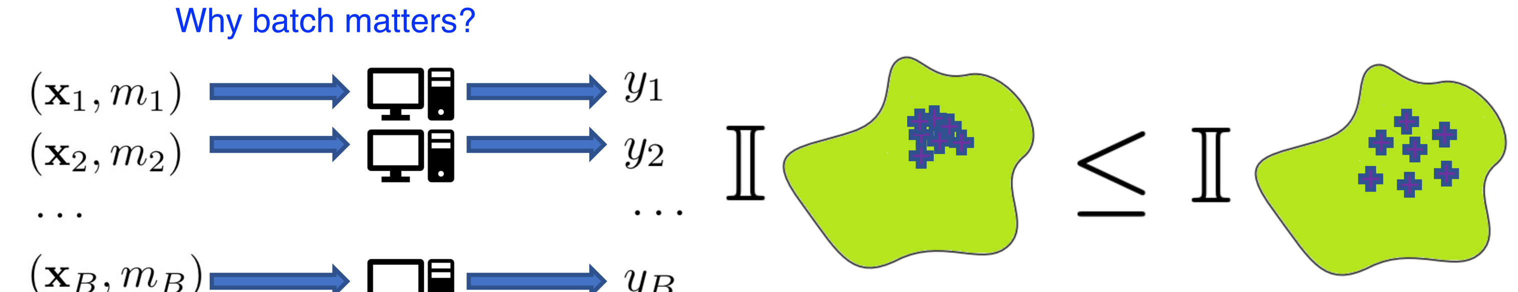


Methods

Our Contribution:

- A batch multi-fidelity active learning approach for high-dimensional outputs
- Consider budget constraints in query samples
- A proved efficient greedy algorithms that nearly 1-1/e optimality

Why batch matters?



Intuition of Novel Batch Acquisition

- Single acquisition that considers the improvements of at all inputs

$$a_s(m, \mathbf{x}) = \mathbb{E}_{p(\mathbf{x}')}\left[\mathbb{I}\left(\mathbf{y}_m(\mathbf{x}), \mathbf{y}_M(\mathbf{x}') | \mathcal{D}\right)\right]$$

- Batch acquisition under budget *B*

$$a_{\text{batch}}(\mathcal{M}, \mathcal{X}) = \mathbb{E}_{p(\mathbf{x}')}\left[\mathbb{I}\left(\{\mathbf{y}_{m_j}(\mathbf{x}_j)\}_{j=1}^n, \mathbf{y}_M(\mathbf{x}') | \mathcal{D}\right)\right], \quad \text{s.t. } \sum_{j=1}^n \lambda_{m_j} \le B$$
$$\mathcal{M} = \{m_1, \ldots, m_n\}, \mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$$

Intractable expectation

- Monte-Carlo Approximation of Target Acquisition

$$\widehat{a}_{\text{batch}}(\mathcal{M}, \mathcal{X}) = \frac{1}{A}\sum_{l=1}^{A} \mathbb{I}\left(\{\mathbf{y}_{m_j}(\mathbf{x}_j)\}_{j=1}^n, \mathbf{y}_M(\mathbf{x}'_l) | \mathcal{D}\right) \quad \text{s.t. } \sum_{j=1}^n \lambda_{m_j} \le B$$

Incurs combinatorial search over multiple fidelities

- Weighed Greedy Optimization

$$\hat{a}_{k+1}(\mathbf{x}, m) = \frac{1}{A}\sum_{l=1}^{A} \frac{\mathbb{I}\left(\mathcal{Y}_k \cup \{\mathbf{y}_m(\mathbf{x})\}, \mathbf{y}_M(\mathbf{x}'_l)|\mathcal{D}\right) - \mathbb{I}\left(\mathcal{Y}_k, \mathbf{y}_M(\mathbf{x}'_l)|\mathcal{D}\right)}{\lambda_m}$$
$$\text{s.t. } \lambda_m + \sum_{\widehat{m} \in \mathcal{M}_k} \lambda_{\widehat{m}} \le B,$$

Theorem 3.1. *At any step of Weighted-Greedy (Algorithm 1) before any choice of fidelity would exceed the budget, and the total budget used to that point is $B' < B$, then the mutual information of the current solution is within $(1 - 1/e)$ of OPT(B').*

Corollary 3.1. *If Weighted-Greedy (Algorithm 1) is run until input-fidelity pair $(\mathbf{x}, m)$ that corresponds with the maximal acquisition function $\hat{a}_{k+1}(\mathbf{x}, m)$ would exceed the budget, it selects that input-fidelity pair anyways (the solution exceeds the budget B) and then terminates, the solution obtained is within $(1 - 1/e)$ of OPT(B).*

Algorithm 1 Weighted-Greedy($\{\lambda_m\}$, budget B)

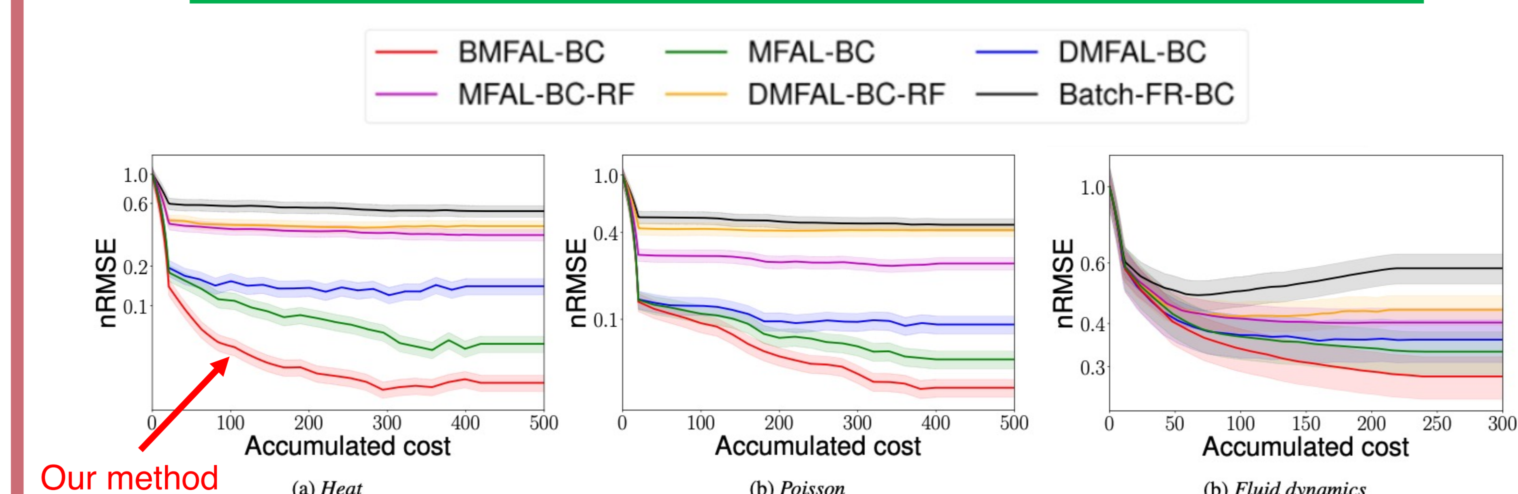
1: $k \leftarrow 0, \mathcal{Q}_k \leftarrow \emptyset, C_k \leftarrow 0$
2: **while** $C_k \le B$ **do**
3: Optimize the weighted incremental acquisition function in (6):
$$(\mathbf{x}^*, m^*) = \operatorname*{argmax}_{\mathbf{x}\in\Omega, 1\le m\le B-C_k} \hat{a}_{k+1}(\mathbf{x}, m)$$
4: **if** Infeasible **then**
5: break
6: **end if**
7: $k \leftarrow k+1$
8: $\mathcal{Q}_k \leftarrow \mathcal{Q}_{k-1} \cup \{(\mathbf{x}^*, m^*)\}$
9: $C_k \leftarrow C_{k-1} + \lambda_{m^*}$
10: **end while**
11: Return $\mathcal{Q}_k$

Proved near 1-1/e optimal

Experiment

- **BMFAL-BC**: our weighted greedy strategy
- DMFAL-BC: DMFAL with original acquisition
- MFAL-BC: DMFAL with modified acquisition
- DMFAL-BC-RF: DMFAL with original acquisition but random fidelity selection
- MFAL-BC-RF: DMFAL with modified acquisition but random fidelity selection
- Batch-RF-BC: Pure random selections of fidelities and query inputs

For all compared methods, the surrogate is updated in a batch fashion until the budgets are exhausted.



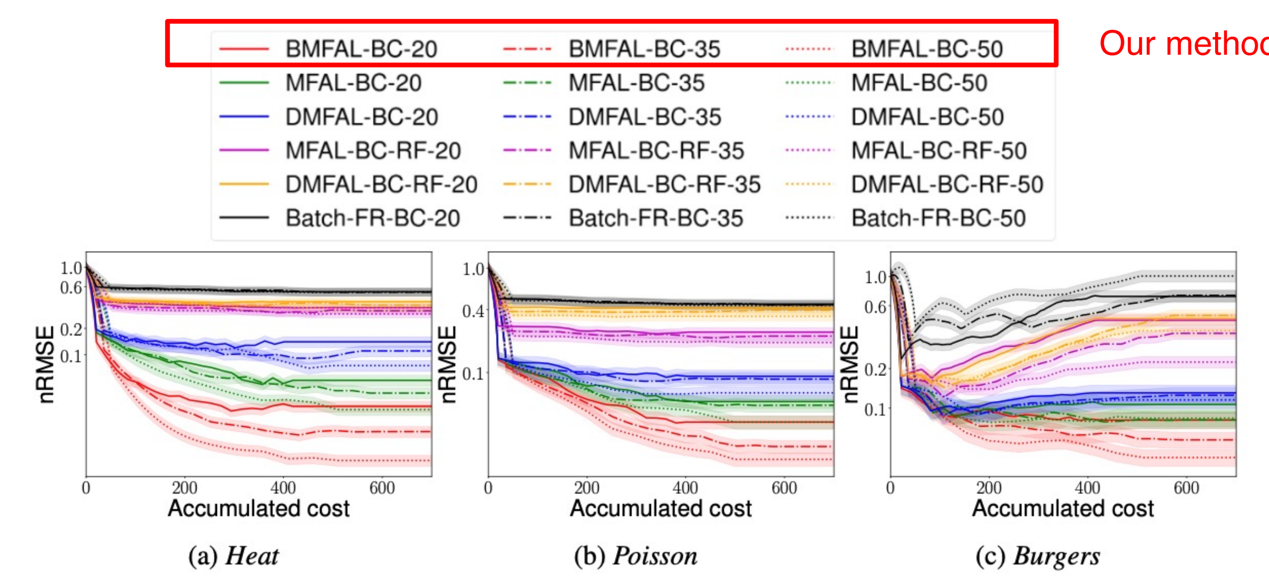


Figure 4: nRMSE *vs.* the accumulated cost under different budgets per batch: $B \in \{20, 35, 50\}$.