Finite Expression Method: A Symbolic Approach for Scientific Machine Learning

Haizhao Yang
Department of Mathematics
Department of Computer Science
University of Maryland College Park

CBMS Conference: Deep Learning and Numerical PDEs
June 19, 2023
Morgan State University

Finite Expression Method (FEX)

Solving Problems with Mathematical Expressions

- Learning the solution of high-dimensional PDEs:

$$f : \mathbb{R}^d \to \mathbb{R}^{O(1)}$$

- Learning mathematical operators or governing equations:

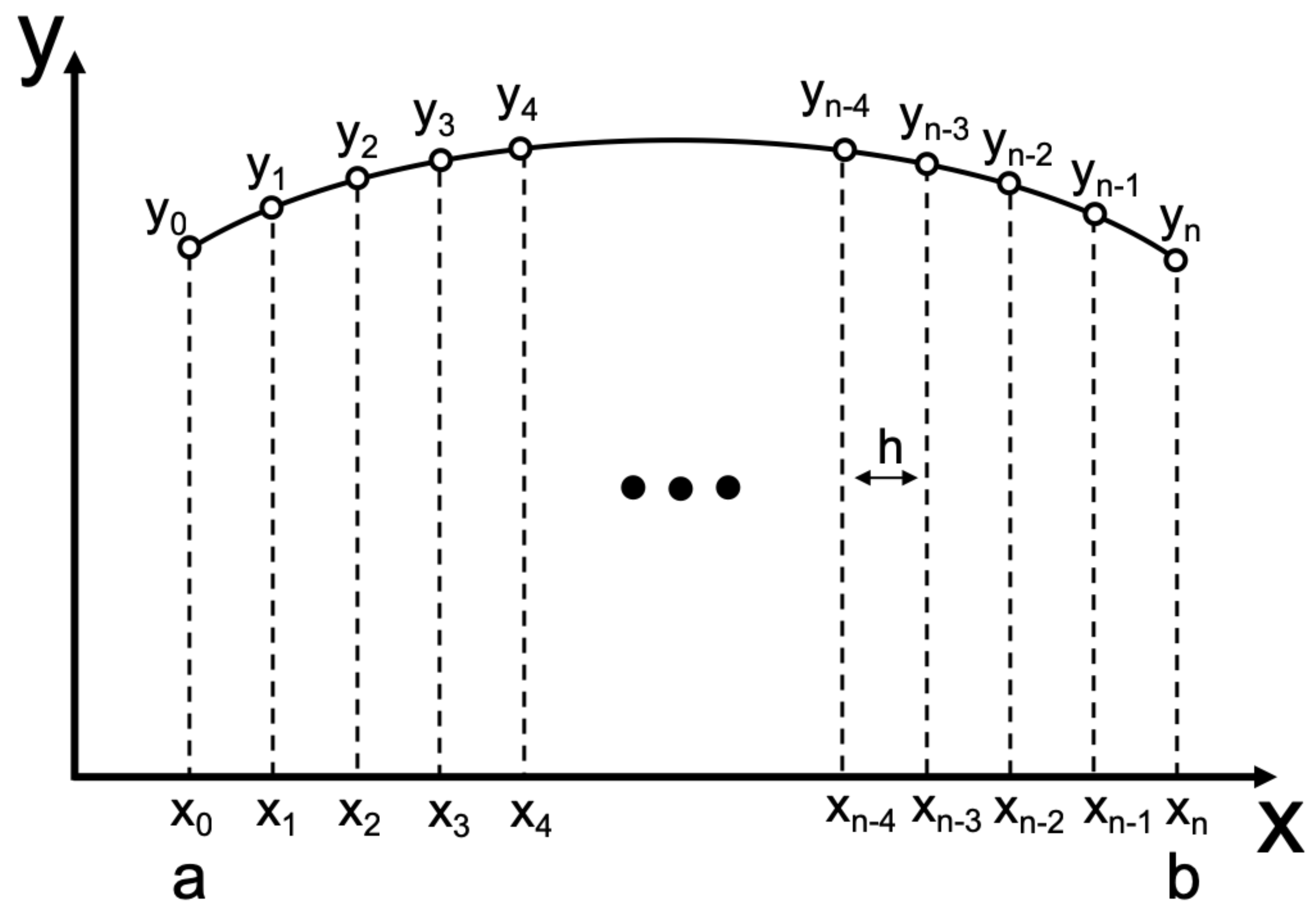
$$f : \mathcal{X} \to \mathcal{Y}$$

- Others in the future

Overview of PDE Solvers

Mesh-based methods:

- Finite difference method, finite element method, etc.
- High accuracy with numerical convergence
- Curse of dimensionality in approximation: $O(1/\epsilon^d)$ parameters



$$\epsilon = h = O\left(\frac{1}{n}\right) \Leftrightarrow n = O\left(\frac{1}{\epsilon}\right)$$

Overview of PDE Solvers

Mesh-free methods:

○ Neural network-based methods

- e.g., $\mathscr{D}(u) = f \quad \text{in } \Omega \qquad \text{and} \qquad \mathscr{B}(u) = g \quad \text{on } \partial\Omega$
- A neural network $\phi(x; \theta^*)$ is constructed to approximate the solution u via least square fitting

$$\theta^* = \arg\min_\theta \mathscr{L}(\theta) := \arg\min_\theta \|\mathscr{D}\phi(x;\theta) - f(x)\|_2^2 + \lambda\|\mathscr{B}\phi(x;\theta) - g(x)\|_2^2$$

or numerically

$$\theta^* = \arg\min_\theta \mathscr{L}(\theta) := \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n} |\mathscr{D}\phi(x_i;\theta) - f(x_i)|^2 + \lambda\frac{1}{m}\sum_{j=1}^{m} |\mathscr{B}\phi(x_j;\theta) - g(x_j)|^2$$

where $\lambda > 0$ is a hyperparameter

Overview of PDE Solvers

Neural network advantage

- No curse of dimensionality in approximation
- $O(d^2)$ parameters to achieve arbitrary accuracy, Shen, Y., Zhang, arXiv:2107.02397, JMLR, 2022

Overview of PDE Solvers

Neural network challenges

- Neural network optimization

$$\theta^* = \arg\min_{\theta} \mathscr{L}(\theta) := \arg\min_{\theta} \|\mathscr{D}\phi(x;\theta) - f(x)\|_2^2 + \lambda\|\mathscr{B}\phi(x;\theta) - g(x)\|_2^2$$

or numerically

$$\theta^* = \arg\min_{\theta} \mathscr{L}(\theta) := \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n} |\mathscr{D}\phi(x_i;\theta) - f(x_i)|^2 + \lambda \frac{1}{m}\sum_{j=1}^{m} |\mathscr{B}\phi(x_j;\theta) - g(x_j)|^2$$

where $\lambda > 0$ is a hyper parameter

- Monte Carlo error $\frac{C_d}{\sqrt{n}}$
- Non-convex optimization
- May require exponentially large number of iterations (E and Wojtowytsch, arXiv:2005.10815)

Question: How to obtain a numerical solver accurate in high dimensions?

Idea:

- Solutions with structures
- Machine learning to identify structures

Finite Expression Method (FEX)

Liang and Y. arXiv:2206.10121

Motivating Problem:

- A **structured** high-dimensional Poisson equation

$$-\Delta u = f \quad \text{for } x \in \Omega, \quad u = g \text{ for } x \in \partial\Omega$$

with a solution $u(x) = \frac{1}{2}\sum_{i=1}^{d} x_i^2$ of low complexity $O(d)$, i.e., $O(d)$ operators in this expression

Idea:

- Find an explicit expression that approximates the solution of a PDE
- Function space with finite expressions
    - **Mathematical expressions**: a combination of symbols with rules to form a valid function, e.g., $\sin(2x) + 5$
    - ***k*-finite expression**: a mathematical expression with at most k operators
    - Function space in FEX: $\mathbb{S}_k$ as the set of s-finite expressions with $s \leq k$

Finite Expression Method (FEX)

Liang and Y. arXiv:2206.10121

Advantages: No curse of dimensionality in approximation

Theorem (Liang and Y. 2022) Suppose the function space is $\mathbb{S}_k$ generated with operators including \`\`+", \`\`-", \`\`$\times$", \`\`/", \`\`$\max\{0,x\}$", \`\`sin(x)", and \`\`2^x". Let $p \in [1, +\infty)$. For any f in the Holder function class $\mathcal{H}_\mu^\alpha([0,1]^d)$ and $\varepsilon > 0$, there exists a k-finite expression ϕ in $\mathbb{S}_k$ such that

$$\|f - \phi\|_{L^p} \leq \varepsilon,$$

if

$$k \geq \mathcal{O}(d^2(\log d + \log \frac{1}{\varepsilon})^2).$$

Finite Expression Method (FEX)

Liang and Y. arXiv:2206.10121

Advantages:

- Lessen the curse of dimensionality in numerical computation for structured problems
- To be proved numerically

Finite Expression Method

Least square based FEX

- e.g., $\mathscr{D}(u) = f$ in Ω and $\mathscr{B}(u) = g$ on $\partial\Omega$
- A mathematical expression u^* to approximate the PDE solution via

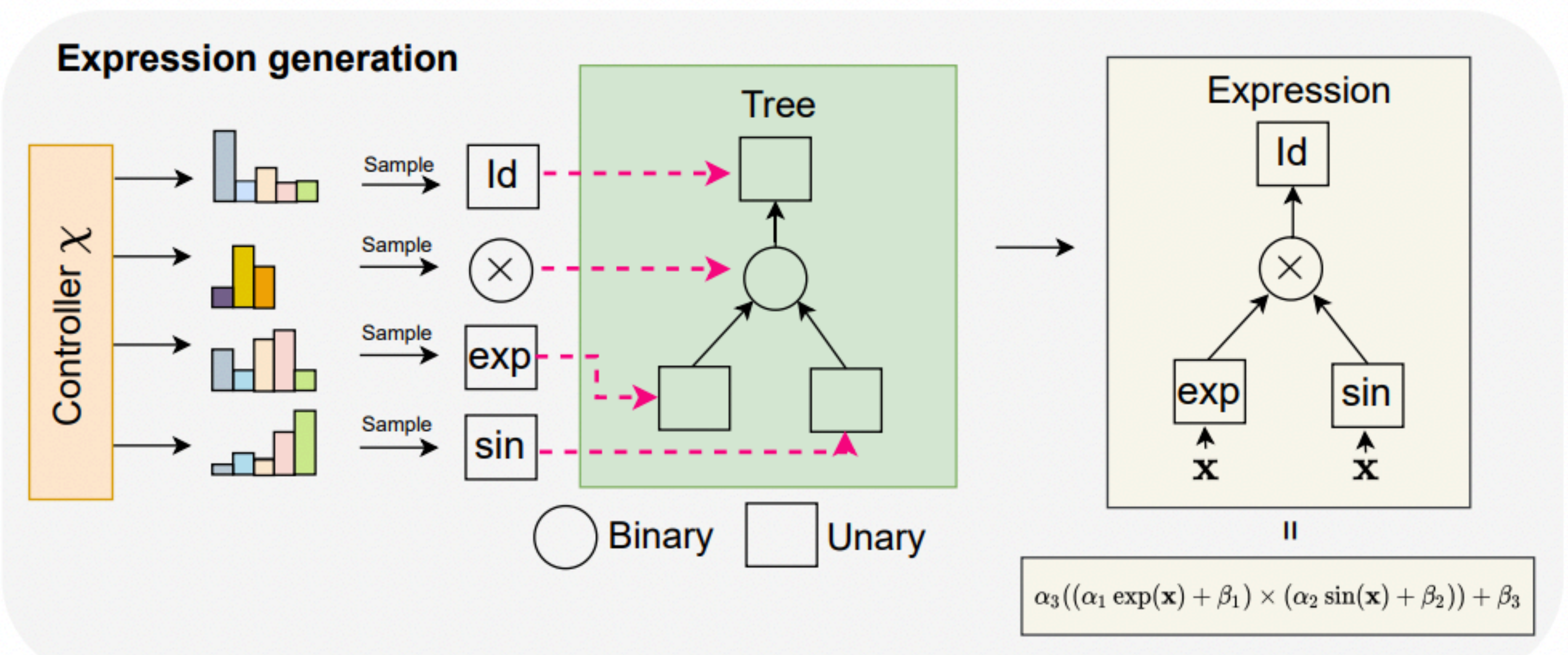
$$u^* = \arg\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \arg\min_{u \in \mathbb{S}_k} \|\mathscr{D}u - f\|_2^2 + \lambda \|\mathscr{B}u - g\|_2^2$$

- Or numerically

$$u^* = \arg\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \arg\min_{u \in \mathbb{S}_k} \frac{1}{n} \sum_{i=1}^{n} |\mathscr{D}u(x_i) - f(x_i)|^2 + \lambda \frac{1}{m} \sum_{j=1}^{m} |\mathscr{B}u(x_j) - g(x_j)|^2$$

- Question: how to solve this combinatorial optimization problem?

Continuous Relaxation of FEX



Finite Expression Method

Least square based FEX

- e.g., $\mathscr{D}(u) = f$ in Ω and $\mathscr{B}(u) = g$ on $\partial\Omega$
- A mathematical expression u^* to approximate the PDE solution via

$$u^* = \arg\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \arg\min_{u \in \mathbb{S}_k} \|\mathscr{D}u - f\|_2^2 + \lambda \|\mathscr{B}u - g\|_2^2$$

- Continuous relaxation with k probability distributions for selecting k operators

$$\begin{aligned}(P_1^*, \ldots, P_k^*) &= \arg\min_{P_1, \ldots, P_k} \mathbb{E}_{u \sim (P_1, \ldots, P_k)} \left[\mathscr{L}(u)\right] \\ &= \arg\min_{P_1, \ldots, P_k} \mathbb{E}_{u \sim (P_1, \ldots, P_k)} \left[\|\mathscr{D}u - f\|_2^2 + \lambda \|\mathscr{B}u - g\|_2^2\right]\end{aligned}$$

and gradient descent in the space of probability distributions

- Finally, $u^* \sim (P_1^*, \ldots, P_k^*)$

Numerical Comparison

- NN method:
  - Neural networks with a ReLU^2-activation function
  - ResNet with depth 7 and width 50

- FEX method:
  - Depth 3 binary tree
  - Binary set $\mathbb{B} = \{+, -, \times\}$
  - Unary set $\mathbb{U} = \{0, 1, \mathrm{Id}, (\cdot)^2, (\cdot)^3, (\cdot)^4, \exp, \sin, \cos\}$

Poisson Equation

- Boundary value problem:

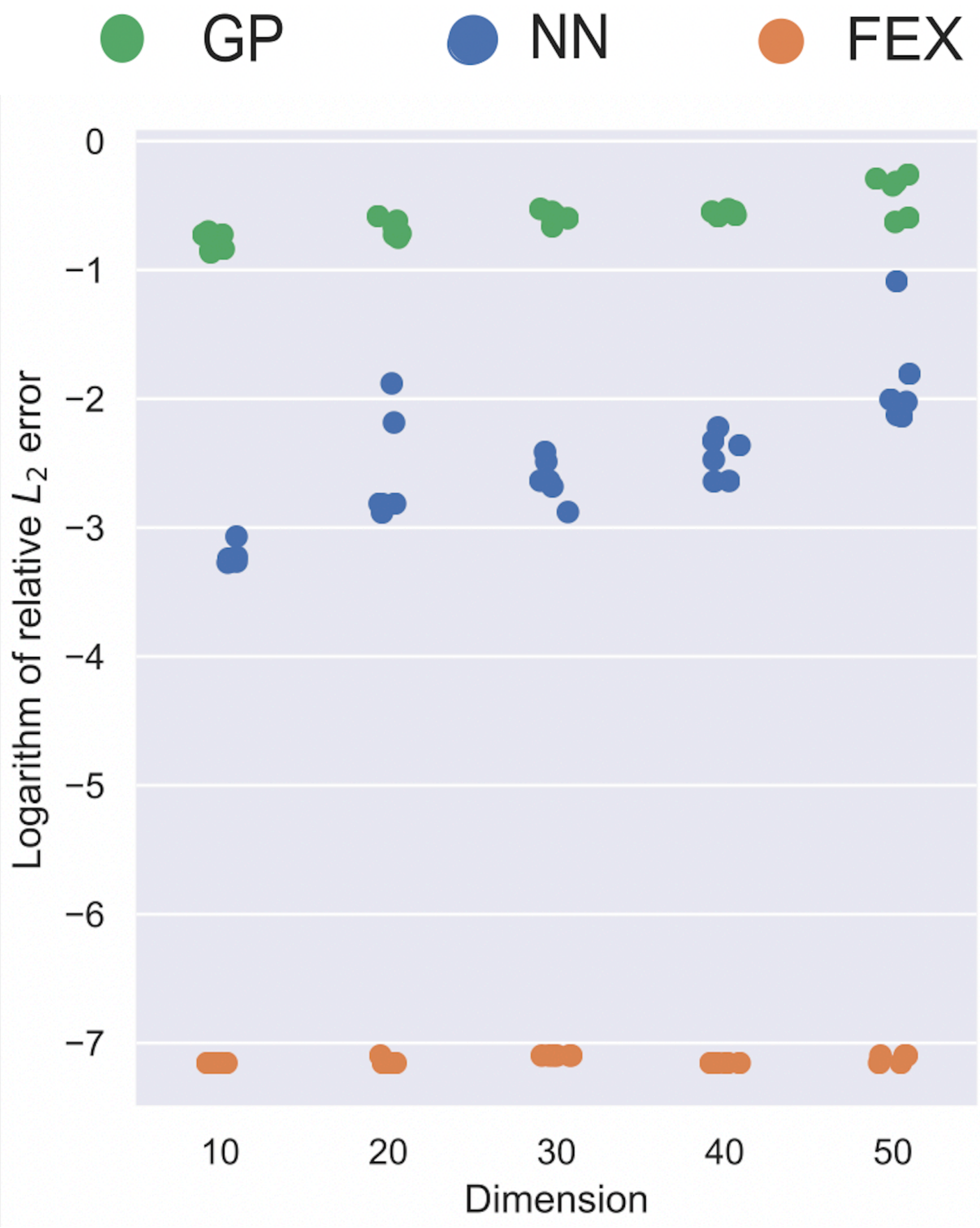
$$-\Delta u = f \quad \text{for } x \in \Omega$$
$$u = g \quad \text{for } x \in \partial\Omega$$

- $\Omega = [-1,1]^d$
- True solution $u(x) = \dfrac{1}{2}\sum_{i=1}^{d} x_i^2$
- Stochastic optimization:

$$\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \min_{u \in \mathbb{S}_k} \| -\Delta u(x) - f(x)\|^2_{L^2(\Omega)} + \lambda \|u(x) - g(x)\|^2_{L^2(\partial\Omega)}$$

with Monte Carlo discretization of high-dimensional integrals

Poisson Equation



Nonlinear Schrodinger Equation

- Consider

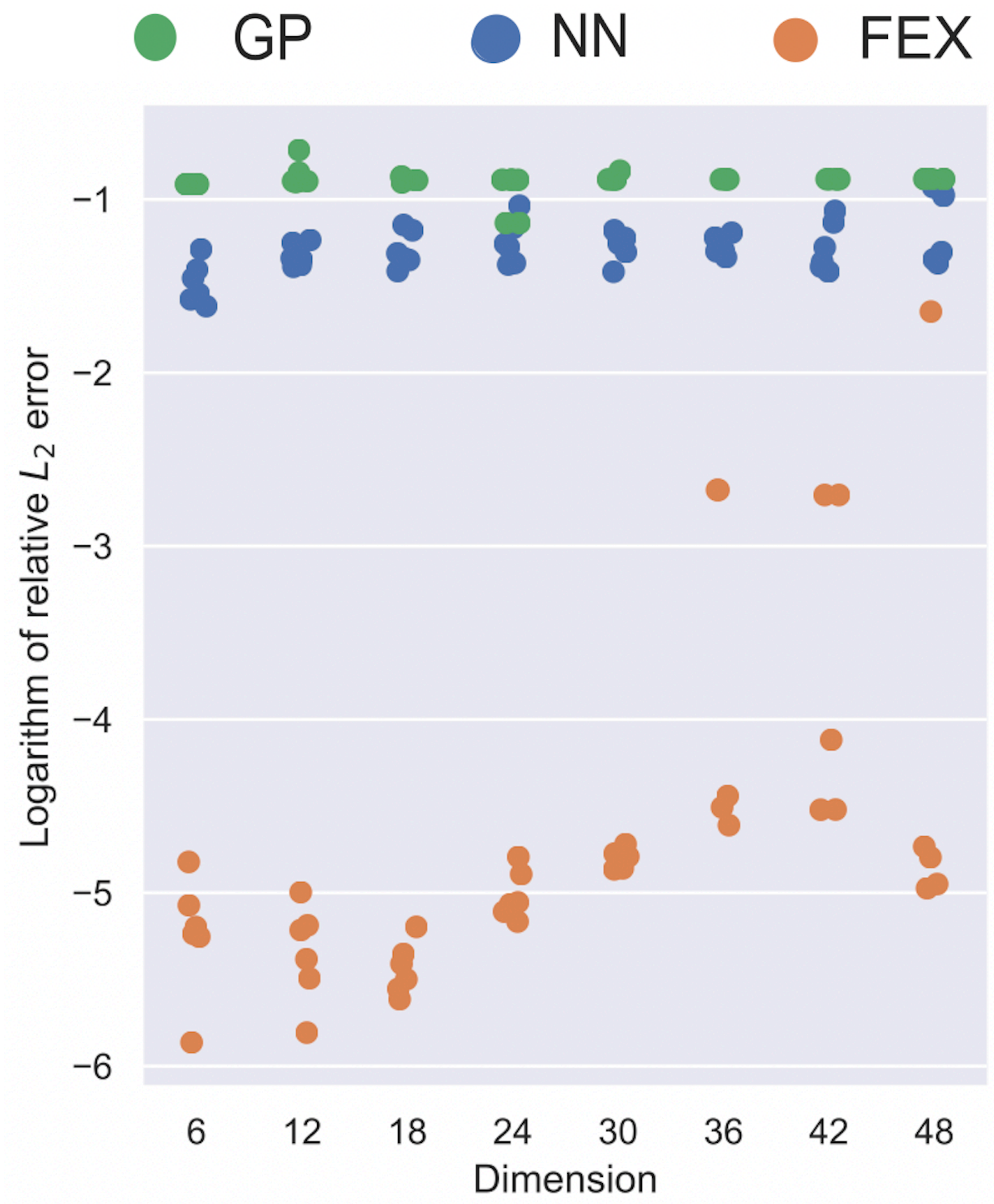
$$-\Delta u + u^3 + Vu = 0 \quad \text{for } x \in \Omega$$

- $V(x) = -\frac{1}{9}\exp(\frac{2}{d}\sum_{i=1}^{d}\cos x_i) + \sum_{i=1}^{d}(\frac{\sin^2 x_i}{d^2} - \frac{\cos x_i}{d})$ for $x = (x_1, \cdots, x_d)$
- $\Omega = [-1,1]^d$
- True solution $u(x) = \exp(\frac{1}{d}\sum_{j=1}^{d}\cos(x_j))/3$
- Stochastic optimization:

$$\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \min_{u \in \mathbb{S}_k} \| -\Delta u + u^3 + Vu \|^2_{L_2(\Omega)} / \|u\|^3_{L_2(\Omega)}$$

with Monte Carlo discretization of high-dimensional integrals

Nonlinear Schrodinger Equation

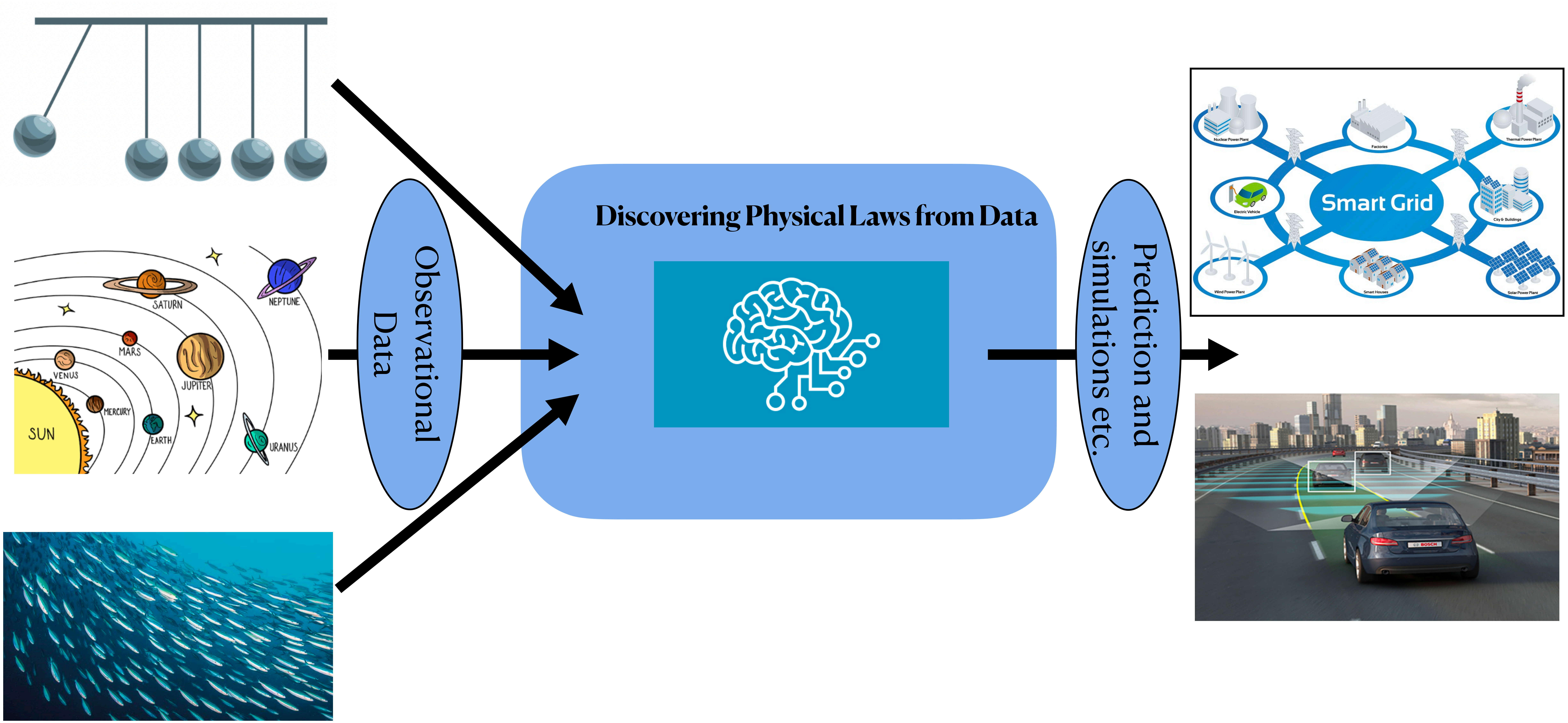


- Learning the solution of high-dimensional PDEs:

$$f : \mathbb{R}^d \to \mathbb{R}^{O(1)}$$

- **Learning mathematical operators or governing equations:**

$$f : \mathscr{X} \to \mathscr{Y}$$


Discovering Physical Laws from Data
Observational Data
Prediction and simulations etc.
Smart Grid
SATURN
NEPTUNE
MARS
VENUS
JUPITER
MERCURY
SUN
EARTH
URANUS

Problem Statement

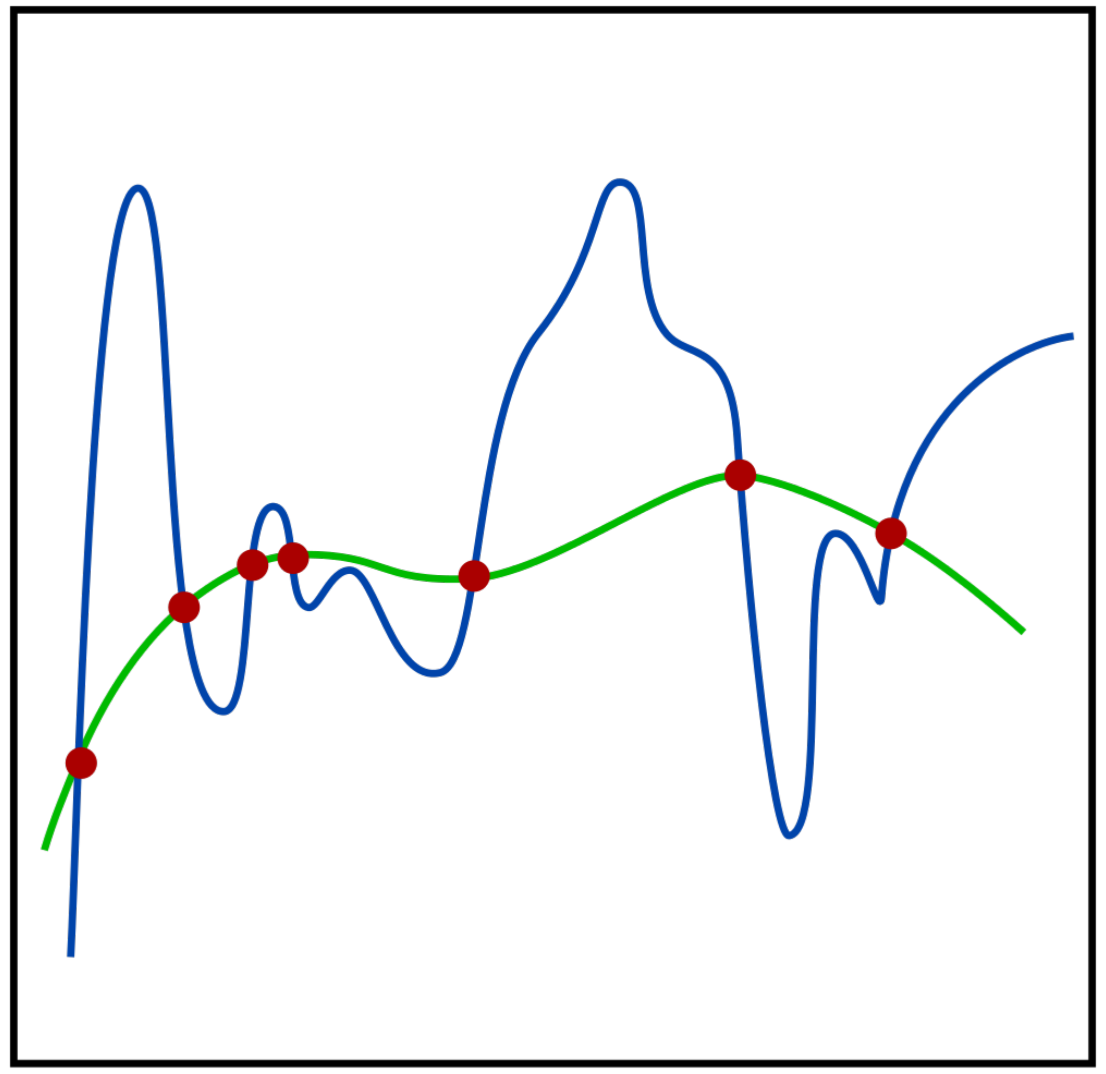
A Concrete Example: 1D Burgers Equation

- **Given** nonlinear operator $F : \mathcal{X} \to \mathcal{Y}$, e.g., $F(u) = \dfrac{\partial u}{\partial t}$
- **Unknown** nonlinear operator $G : \mathcal{X} \to \mathcal{Y}$, e.g., $G(u) = -u \cdot u_x + \nu u_{xx}$
- Assumption: a function $u(x, t) \in \mathcal{X}$ satisfies

$$F(u) = G(u) \quad \Leftrightarrow \quad \frac{\partial u}{\partial t} = -u \cdot u_x + \nu u_{xx}$$

- **Given** discrete data observations $u(x_i, t_j), i = 1, \ldots, m, j = 1, \ldots, n$
- **Goal**: identify G with $G \neq F$
- **Challenges**: 1) non-uniqueness of G (due to data fitting and discretization)

  2) noisy data



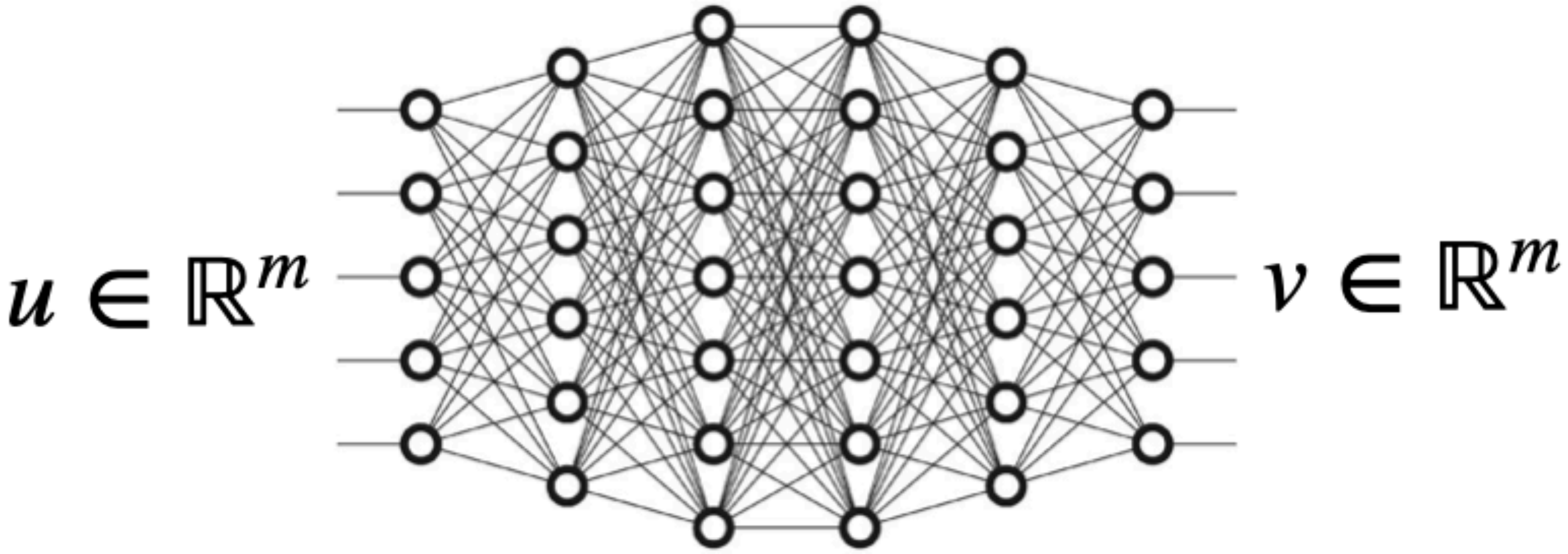
Problem Statement

A Concrete Example: 1D Burgers Equation

- **Goal**: identify $G(u) = -u \cdot u_x + \nu u_{xx}$
- What's operator G after discretization?

$$G : \mathbb{R}^m \to \mathbb{R}^m$$

  - A high-dimensional function
  - Traditional parametrization methods: **the curse of dimensionality**
  - Neural network parametrization: **no interpretability**



Finite Expression Method (FEX)

Jiang, Wang, Y. arXiv:2305.08342

Idea:

- Find an explicit expression that approximates the unknown operator G
- Function space with finite expressions
  - k-**finite expression**: a mathematical expression with at most k operators

    e.g., $\sin(2x) + 5e^x$ and $5\frac{\partial}{\partial t}(u)$
  - Function space: $\mathbb{S}_k$ as the set of s-finite expressions with $s \leq k$

Finite Expression Method

Jiang, Wang, Y. arXiv:2305.08342

Least square based FEX

- e.g., $\dfrac{\partial u}{\partial t} = G(u) = -u \cdot u_x + \nu u_{xx}$
- A mathematical expression G^* to approximate the unknown operator via

$$G^* = \arg\min_{G \in \mathbb{S}_k} \mathscr{L}(G) := \arg\min_{G \in \mathbb{S}_k} \|G(u) - u_t\|_2^2$$

- Or numerically

$$G^* = \arg\min_{G \in \mathbb{S}_k} \mathscr{L}(G) := \arg\min_{G \in \mathbb{S}_k} \frac{1}{mn} \sum_{j=1}^{n} \sum_{i=1}^{m} |G(u)(x_i, t_j) - u_t(x_i, t_j)|^2$$

- Continuous relaxation to solve this combinatorial optimization problem

Key Features of FEX

- **No curse of dimensionality** in approximation theory v.s. traditional methods
- **Interpretable** learning outcomes v.s. blackbox neural networks
- **Higher accuracy** v.s. existing symbolic regression tools
- A nonlinear approach to generate **a large set** of expressions from **a small collection** of operators
    - SINDy[1]: require a large manually designed dictionary
    - PDE-Net[2]: only capable of polynomials of operators
    - GP: Genetic programming with poor accuracy
    - SPL[3]: Monte Carlo tree search with poor accuracy

1. Brunton, Proctor, Nathan, Discovering governing equations from data by sparse identification of nonlinear dynamical systems, PNAS, 2016
2. Long, Lu, Dong, PDE-Net 2.0: Learning PDEs from data with a numeric-symbolic hybrid deep network, Journal of Computational Physics 2019
3. Sun et al. Symbolic Physics Learner: Discovering governing equations via Monte Carlo tree search. ICLR 2023

Numerical Example 1:

2D Burgers equation with periodic boundary conditions on $(x, y, t) \in [0,2\pi]^2 \times [0,10]$:

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} + \nu(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2})$$

$$\frac{\partial v}{\partial t} = -u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} + \nu(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2})$$

$$u(x, y,0) = u_0(x, y)$$

$$v(x, y,0) = v_0(x, y)$$

$$\nu = 0.1$$

	PDE-Net 2.0	SINDy	GP	SPL	FEX
Mean Absolute Error	1.086×10^{-3}	3.239×10^{-1}	4.973×10^{-1}	2.1×10^{-1}	2.021×10^{-4}

Numerical Example 1:

Noise Robustness

Numerical Results of the Burger's equation by PDE-Net with different levels of noise

Correct PDE	$C = 0.001$	$C = 0.005$	$C = 0.01$
$-uu_x$	$-1.00uu_x$	$-1.01uu_x$	$-0.88uu_x$
$-vu_y$	$-1.00vu_y$	$-0.92vu_y$	$-0.80vu_y$
$0.05u_{xx}$	$0.0503u_{xx}$	$0.01u_{xx}$	$0.01u_{xx}$
$0.05u_{yy}$	$0.0503u_{yy}$	$0.02u_{yy}$	$0.01u_{yy}$

Numerical Results of the Burger's equation by FEX with different levels of noise

Correct PDE	$C = 0.001$	$C = 0.005$	$C = 0.01$
$-uu_x$	$-1.00uu_x$	$-1.006uu_x$	$-1.025uu_x$
$-vu_y$	$-1.00vu_y$	$-1.002vu_y$	$-0.926vu_y$
$0.05u_{xx}$	$0.0498u_{xx}$	$0.0534u_{xx}$	$0.0617u_{xx}$
$0.05u_{yy}$	$0.0502u_{yy}$	$0.0543u_{yy}$	$0.0612u_{yy}$

Numerical Example 2:

PDE with varying coefficients and periodic boundary conditions:

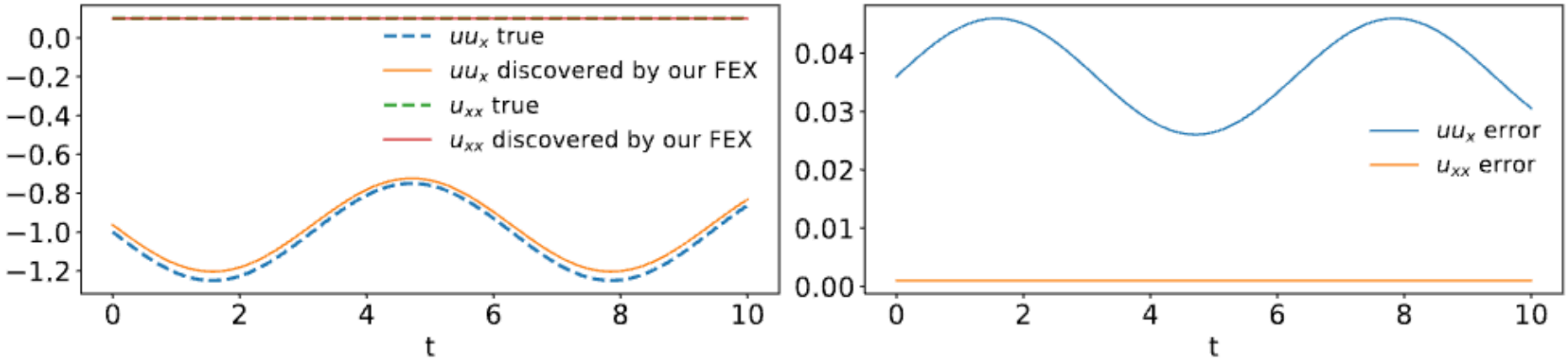
$$u_t(x,t) = a(t)uu_x + \nu u_{xx}, \qquad \forall (x,t) \in [-8,8] \times [0,10],$$

$$u(x,0) = \exp(-(x+1)^2)$$

$$a(t) = 1 + \frac{1}{4}\sin t$$

$$\nu = 0.1$$

Numerical Example 2:



Visualization of the recovery error of varying coefficients

Numerical Example 3:

Johnson-Mehl-Avrami-Kolmogorov nonlinear equation:

$$y = 1 - \exp\left(-kt^n\right)$$

Correct function	$y = 1 - \exp\left(-0.6t^2\right)$
PDE-Net 2.0	$y = 0.0538t + 0.5013t^2 - 0.0888t^3 + 0.0048t^4 - 0.0024$
SINDy	$y = 0.6015t^2 - 0.2042t^4 + 0.0537t^5$
GP	$y = 0.994 - \exp\left(-0.58t^2\right)$
SPL	$y = 0.5165t^2$
FEX	$y = 1.000 - \exp\left(-0.6001t^2\right)$

Finite Expression Method

Summary

- Curse of dimensionality in computation with finite precision
  - Addressed in theory for all continuous functions
  - Lessened for structured problems numerically
- Monte Carlo error
  - Lessened for structured problems numerically
- Challenging optimization
  - Continuous relaxation for mix-integer optimization
  - Randomized algorithms with multiple trials for non-convex optimization
  - Good performance for structured problem numerically

Acknowledgement

Collaborators

Zhongyi Jiang (U. Of Delaware), Senwei Liang (LBNL), Zuowei Shen (NUS), Chunmei Wang (U. Of Florida), Shijun Zhang (Duke)

Reference

- Shen, Y., Zhang. Deep Network Approximation: Achieving Arbitrary Accuracy with Fixed Number of Neurons. JMLR, 2022, arXiv:2107.02397
- Liang and Y., Finite Expression Method for Solving High-Dimensional Partial Differential Equations, arxiv:2206.10121
- Jiang, Wang, Y., Finite Expression Methods for Discovering Physical Laws from Data, arXiv:2305.08342

Funding Support

Poisson Equation

Convergence Test:

- True solution $u(x) = \frac{1}{2}\sum_{i=1}^{d} x_i^2$
- Binary set $\mathbb{B} = \{\, + \,, - \,, \times \,\}$
- Unary set $\mathbb{U} = \{0, 1, \mathrm{Id}, (\,\cdot\,)^3, (\,\cdot\,)^4, \exp, \sin, \cos\}$
- No expression tree to exactly represent u(x)