Finite Expression Method for Solving High-Dimensional PDEs

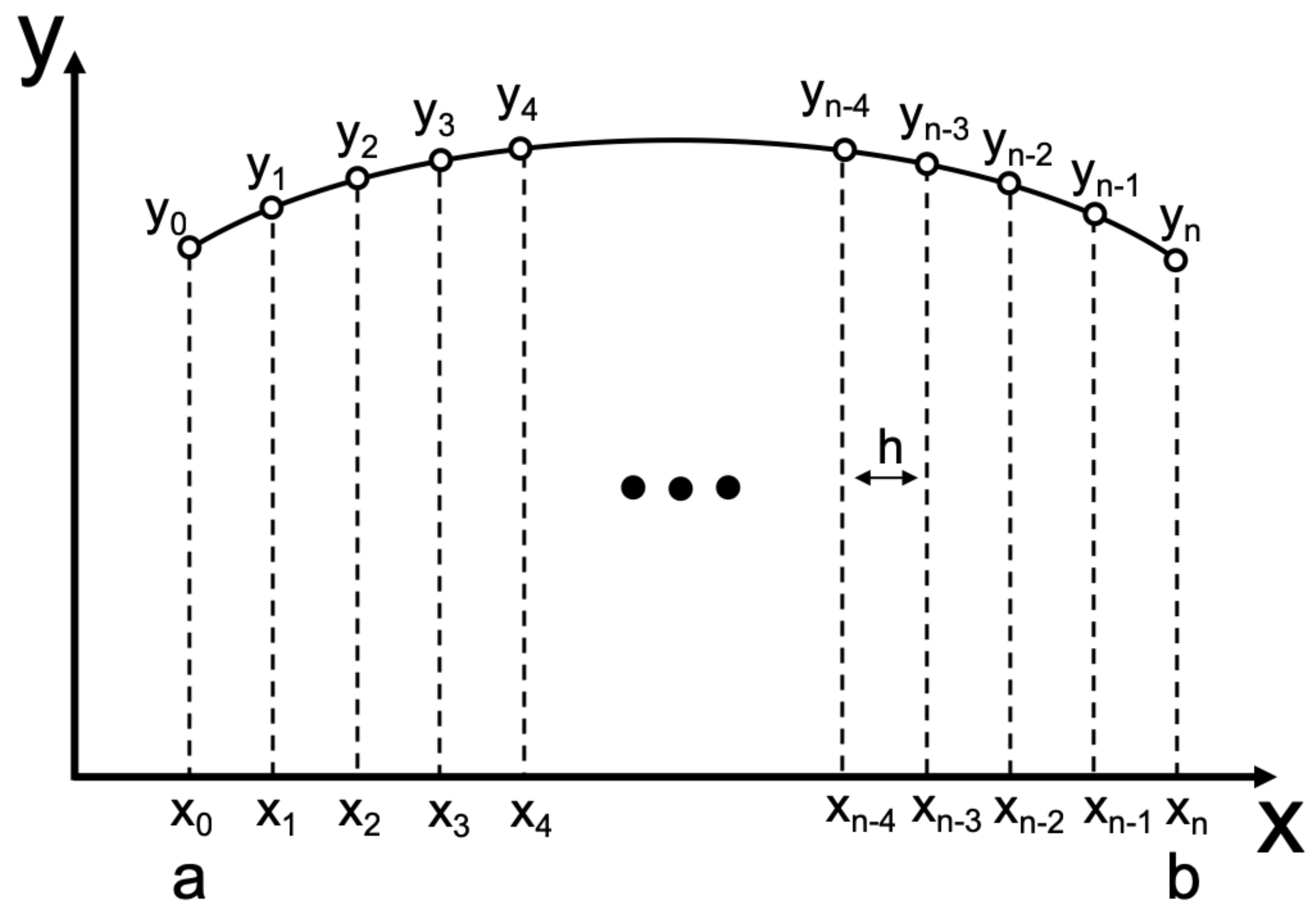
Haizhao Yang
Department of Mathematics
University of Maryland College Park

Applied and Numerical Seminar/SIAM Seminar
University of Florida
Sep 9th, 2022

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 (dating back to 1990s)
  - e.g., $\mathscr{D}(u) = f$ in Ω and $\mathscr{B}(u) = g$ 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

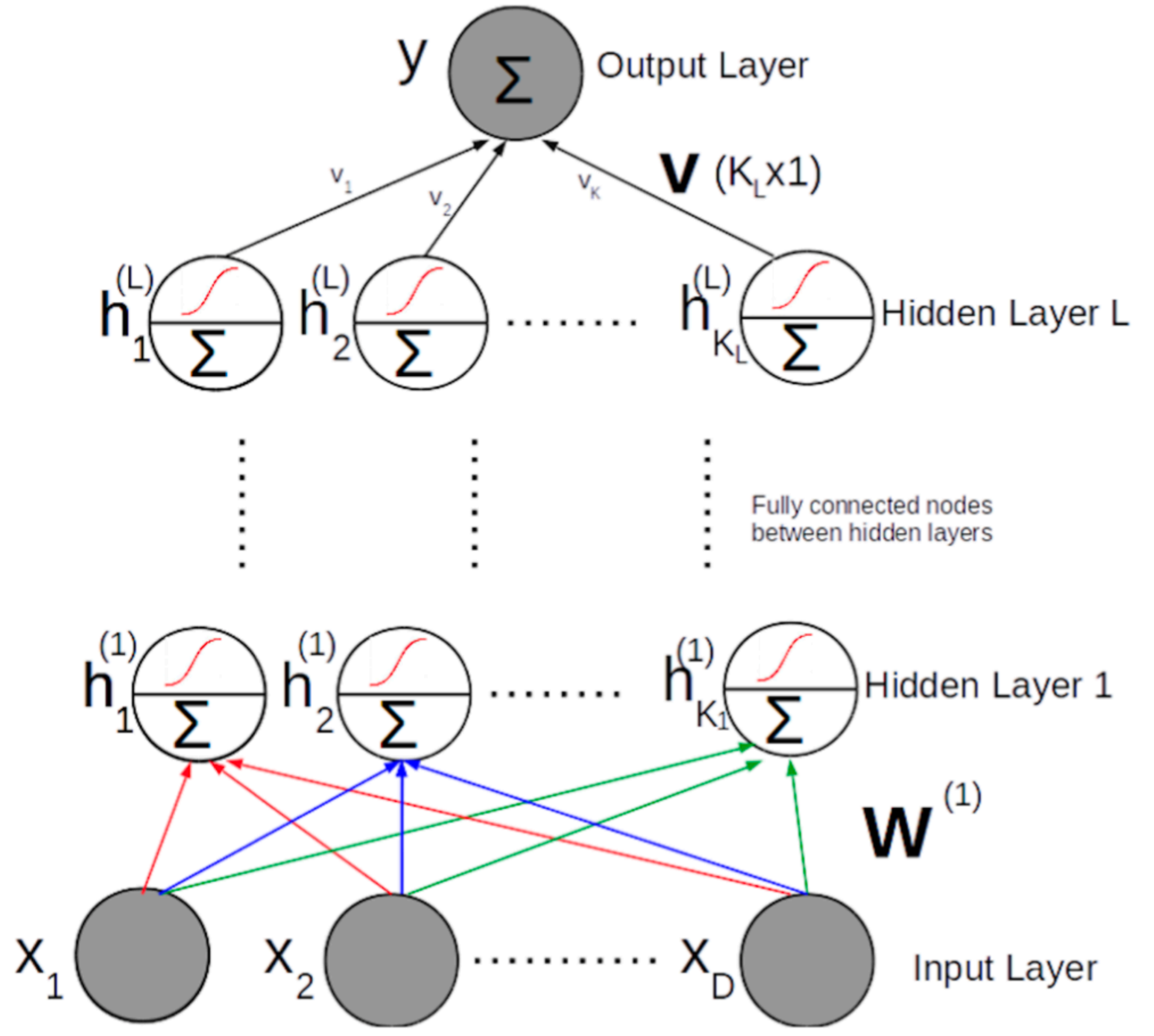
Neural networks

- No curse of dimensionality in approximation
  - $O(d^2)$ parameters to achieve arbitrary accuracy, Shen, Y., Zhang, arXiv:2107.02397
- Curse of dimensionality in numerical computation
  - Optimal nonlinear approximation with continuous parameter selection, DeVore, Howard, Micchelli, 1989

$$y = h(x;\theta) := T \circ \phi(x) := T \circ h^{(L)} \circ h^{(L-1)} \circ \cdots \circ h^{(1)}(x)$$

where

- $h^{(i)}(x) = \sigma({W^{(i)}}^T x + b^{(i)});$
- $T(x) = V^T x;$
- $\theta = (W^{(1)}, \cdots, W^{(L)}, b^{(1)}, \cdots, b^{(L)}, V).$



- **Question:** How to obtain a numerical solver scalable in dimension?
- **Idea:** Find an appropriately small function space with stable computation

- **Question:** What function space is appropriate?
- **Ideas:**

- Barron space: functions with integral representations (Barron, 1993, E et al. 2019, Du et al. 2021, Xu et al. 2021)

  e.g. $f(\mathbf{x}) = \int_{\Omega} a\sigma(\mathbf{b}^T\mathbf{x} + c)\rho(da, d\mathbf{b}, dc), \qquad \mathbf{x} \in X$

  where ρ is a probability distribution or more particularly a Fourier representation

  $$f(\mathbf{x}) = \int_{\mathbb{R}^d} \hat{f}(\omega)\cos(\omega^T\mathbf{x})d\omega = \int_{\mathbb{R}^1\times\mathbb{R}^d} a\cos(\omega^T\mathbf{x})\rho(da, d\omega)$$

- Ours: functions with **finite expressions**

- **Question:** Why finite expressions?
- **Ideas:**
  - Simple, intuitive, and interpretable
  - Sparse or low-complexity structure of a high-dimensional problem

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

- NN: $O(d^2)$ parameters to achieve arbitrary accuracy, Shen, Y., Zhang, arXiv:2107.02397
- NN has finite expressions:
- **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 $\mathscr{H}_\mu^\alpha([0,1]^d)$ and $\varepsilon > 0$, there exists a k-finite expression ϕ in $\mathbb{S}_k$ such that $\|f - \phi\|_{L^p} \le \varepsilon$, if $k \ge \mathcal{O}(d^2(\log d + \log \frac{1}{\varepsilon})^2)$.

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

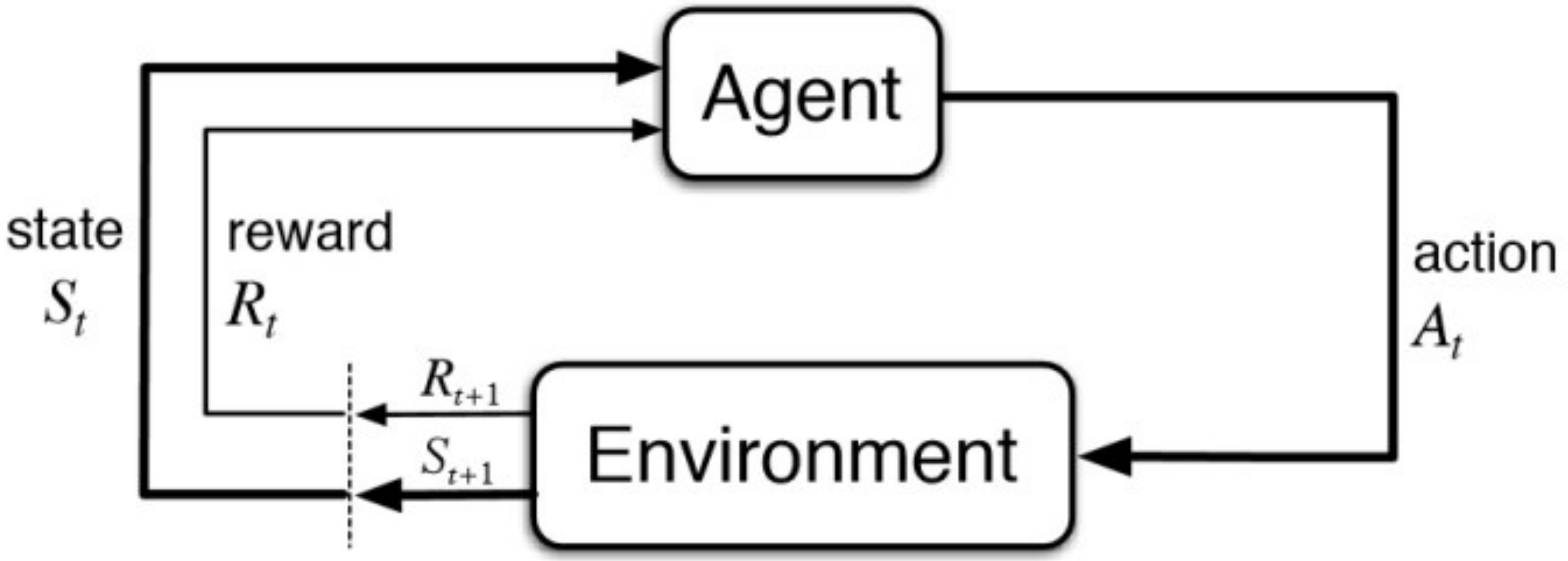
$$u^* = \arg\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \arg\min_{u \in \mathbb{S}_k} \|\mathscr{D}u - f\|_2^2 + \lambda \|\mathscr{B}u - g\|_2^2$$

- 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?

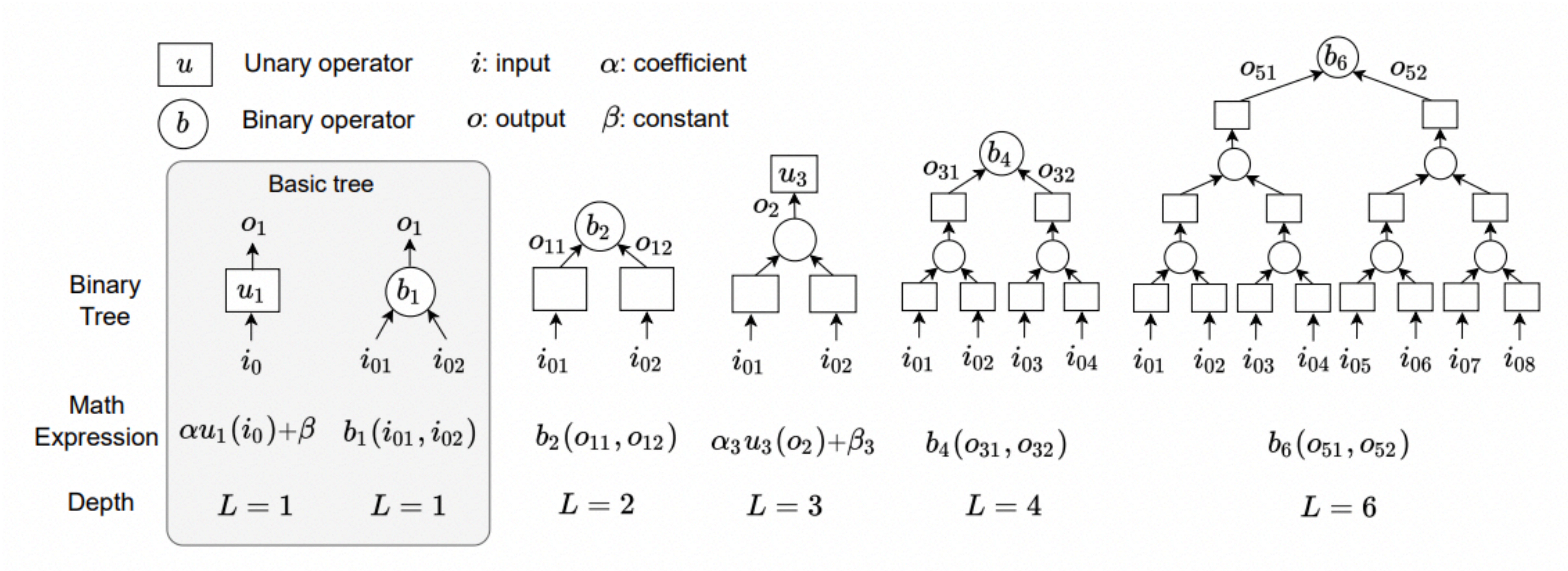
Reinforcement Learning for Combinatorial Optimization



By Richard S. Sutton and Andrew G. Barto.

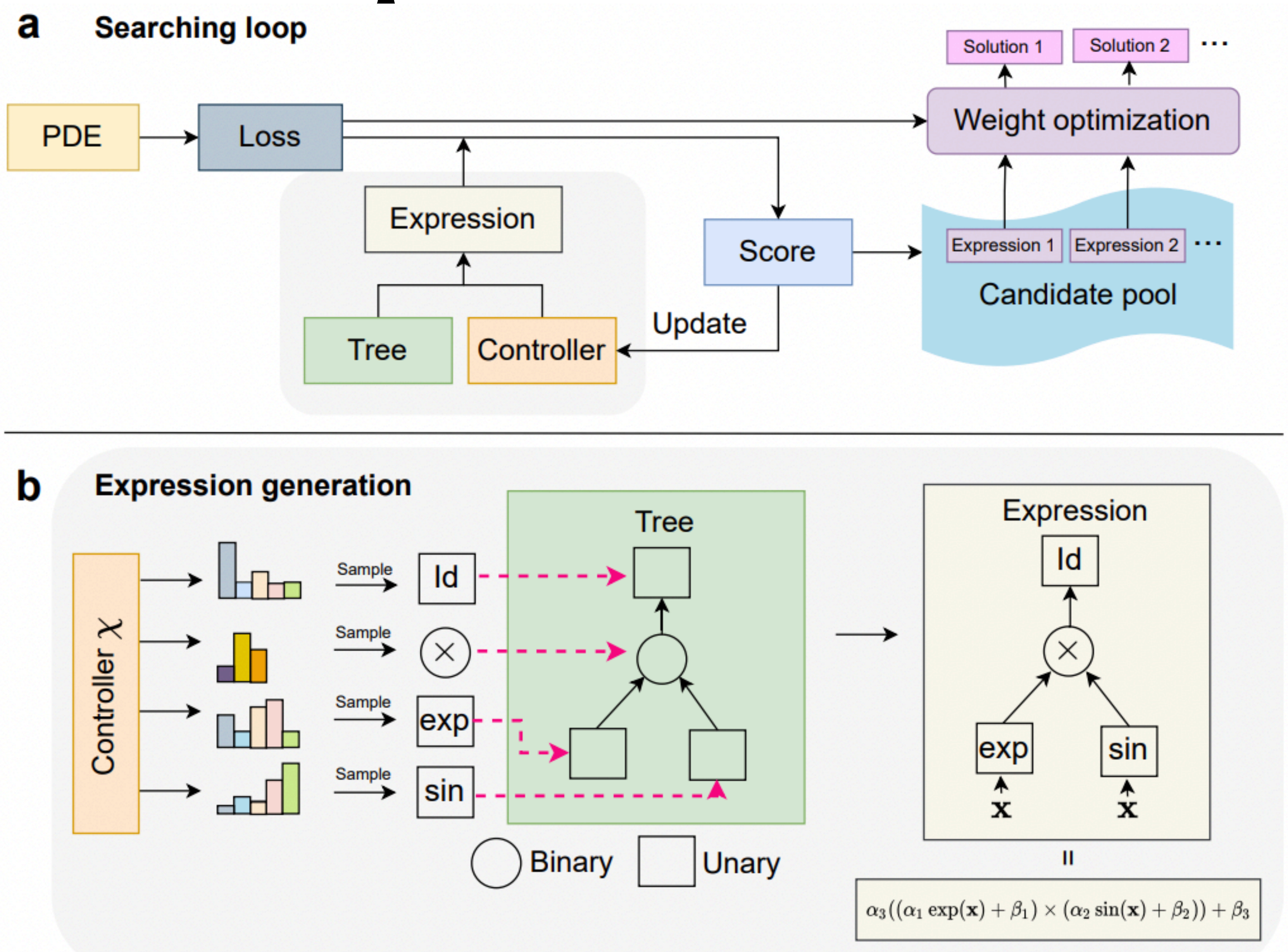
- **Goal:** Apply reinforcement learning to select mathematical expressions to solve a PDE
- **Ideas:**
    1. Reformulate the sequential (selection, realization, evaluation) procedure as a sequence of (action, state, reward)
    2. Reformulate the decision strategy for selection as the policy to take actions
    3. The PDE regression quality as the reward

Expression Generation



An expression tree as a sequence of node values by using its pre-order traversal, e.g., $2\sin(x) + 3$ and $x + y$

Computation Flow of FEX



Learning to Regress in FEX

- **State** at time t:

  The expression tree

- **Action** at time t:

  The operators, variables, and constants drawn from the policy

- **Reward** at time t: $R(a_t) = 1/(1 + \mathscr{L}(u))$

- **Policy (controller)**: $p(a \mid \theta)$ is the probability specified by a deep neural network

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 $\mathbb{U} = \{0, 1, \mathrm{Id}, (\cdot)^2, (\cdot)^3, (\cdot)^4, \exp, \sin, \cos\}$

○ Fex NN method:

- Apply FEX to obtain an estimated solution structure
- Design NN adaptively with this structure,
- e.g., $u(x)=\exp(NN(x;\theta))$

Poisson Equation

- Boundary value problem:

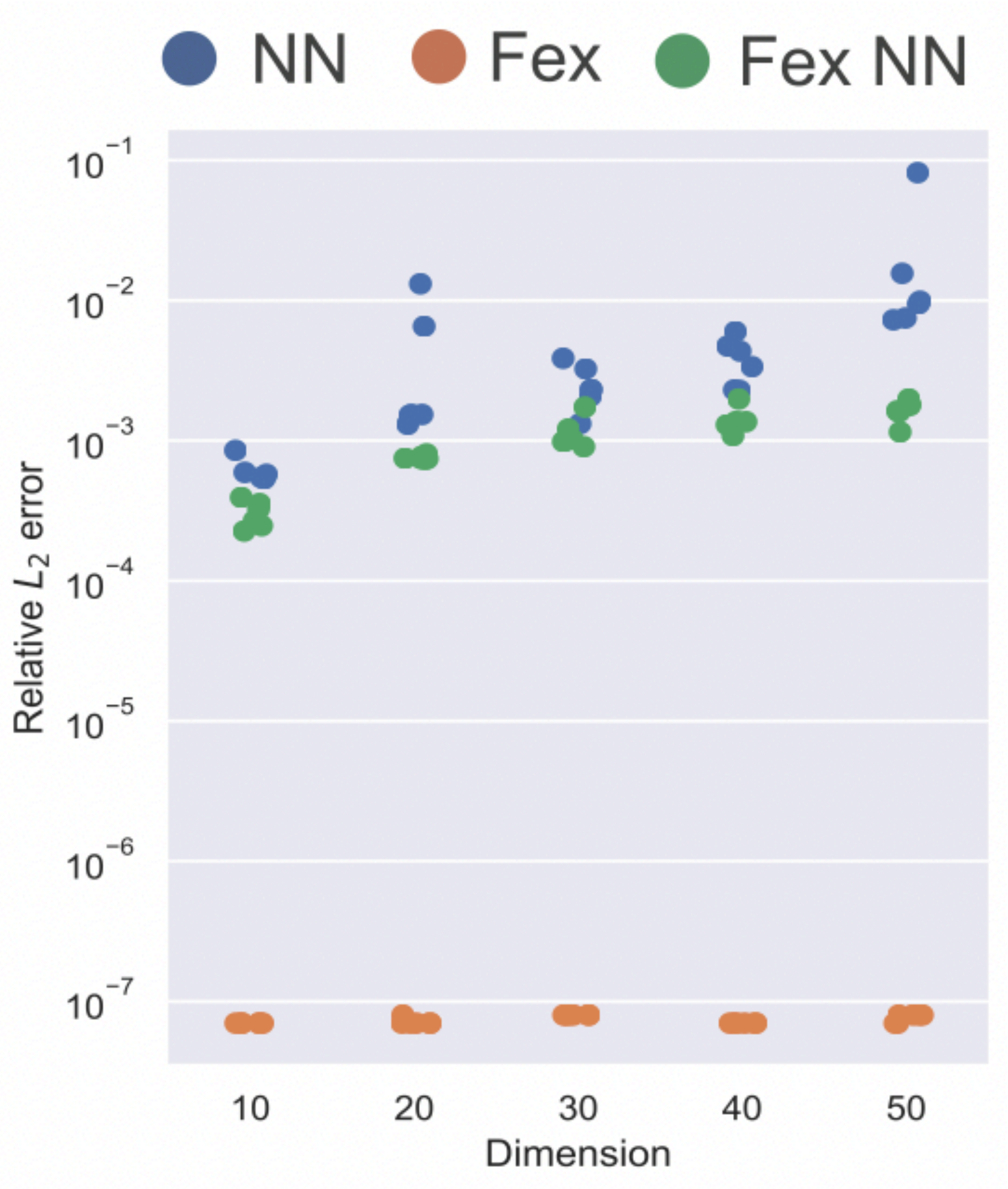
$$\begin{aligned} -\Delta u = f &\quad \text{for } x \in \Omega \\ u = g &\quad \text{for } x \in \partial\Omega \end{aligned}$$

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

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

with Monte Carlo discretization of high-dimensional integrals

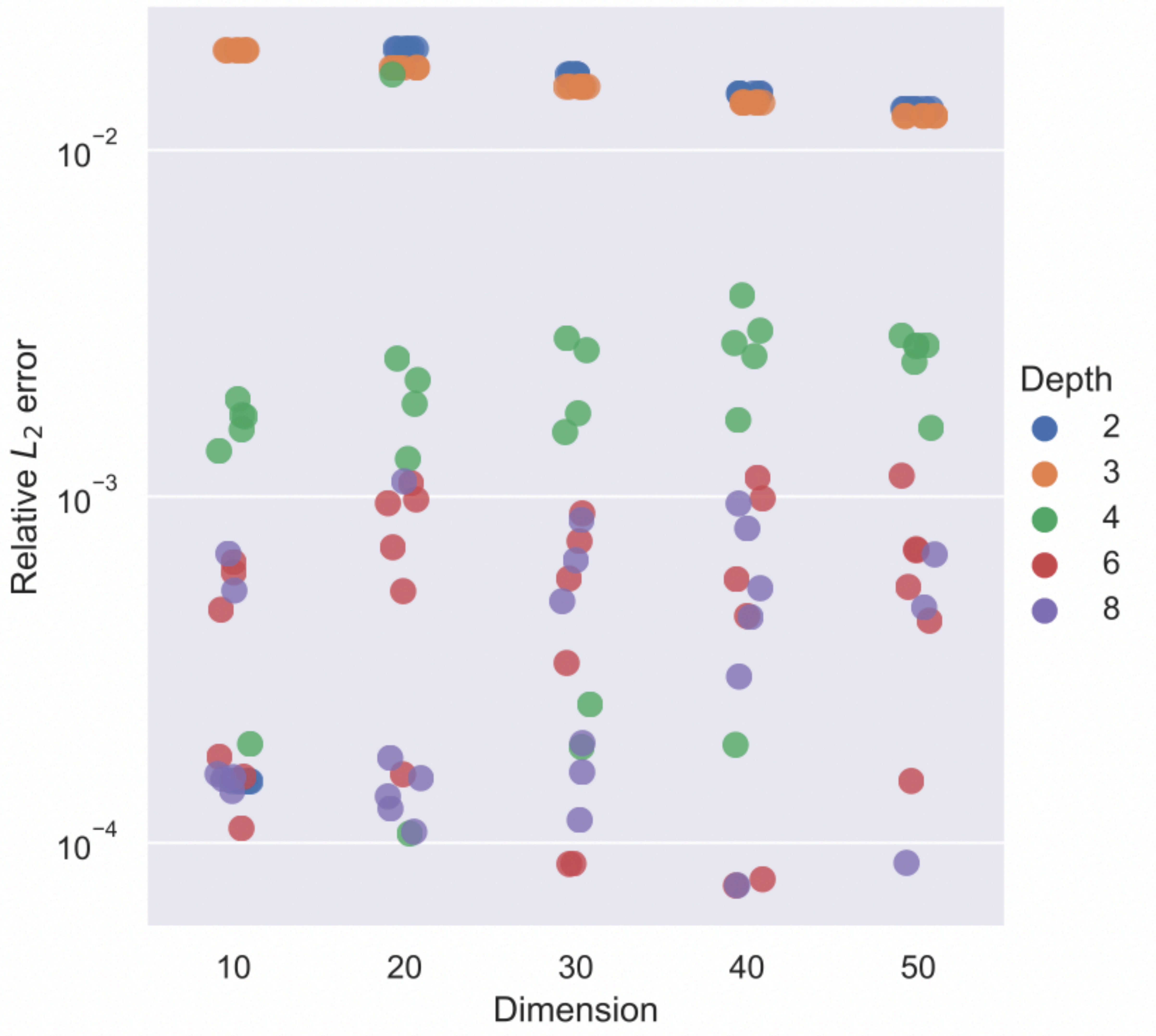
Poisson Equation



Poisson Equation

Convergence Test:

- True solution $u(x) = \dfrac{1}{2}\sum_{i=1}^{d} x_i^2$
- 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)



Linear Conservation Law

- Consider

$$\frac{\pi d}{4} u_t - \sum_{i=1}^{d} u_{x_i} = 0 \quad \text{for } x = (x_1, \cdots, x_d) \in \Omega, t \in [0,1]$$

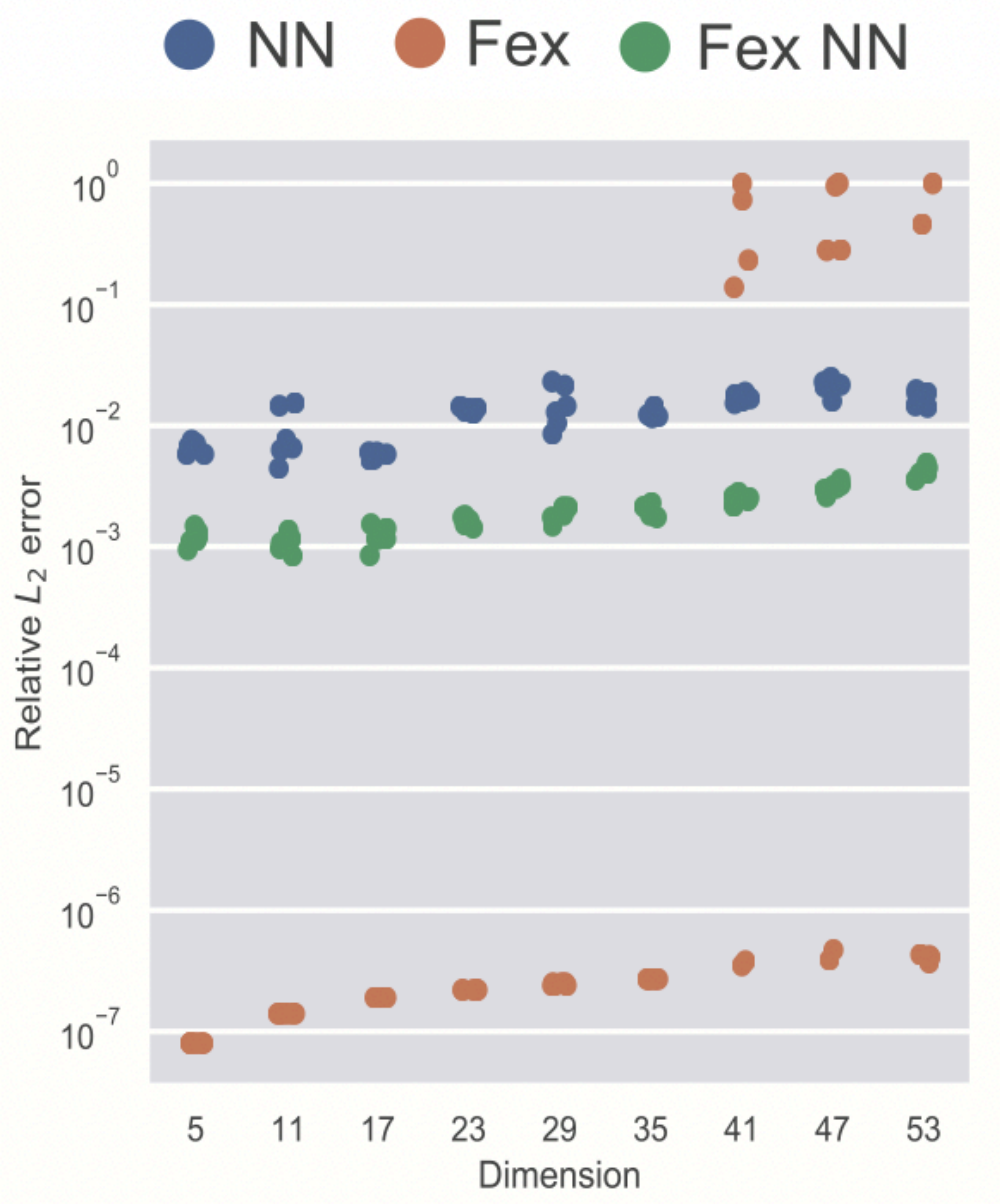
$$u(0,x) = \sin(\frac{\pi}{4} \sum_{i=1}^{d} x_i) \quad \text{for } x \in \Omega$$

- $T \times \Omega = [0,1] \times [-1,1]^d$
- True solution $u(t,x) = \sin(t + \frac{\pi}{4} \sum_{i=1}^{d} x_i)$
- Stochastic optimization:

$$\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \min_{u \in \mathbb{S}_k} \|u_t - \sum_{i=1}^{d} u_{x_i}\|^2_{L^2(T\times\Omega)} + \lambda \|u(0,x) - \sin(\frac{\pi}{4} \sum_{i=1}^{d} x_i)\|^2_{L^2(\Omega)}$$

with Monte Carlo discretization of high-dimensional integrals

Linear Conservation Law



Nonlinear Schrodinger Equation

- Consider

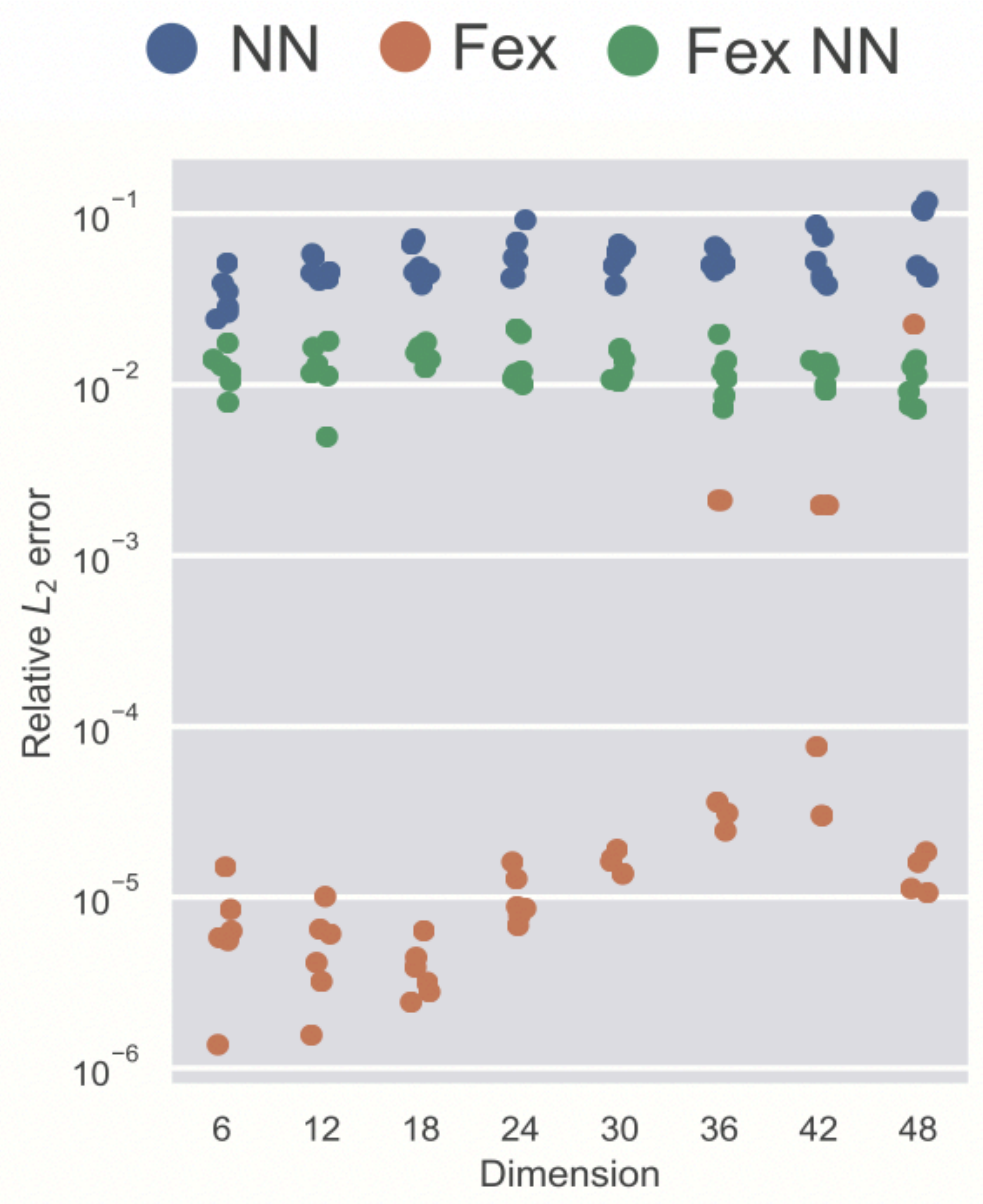
$$-\Delta u + u^3 + Vu = 0 \quad \text{for } x \in \Omega$$

- $V(x) = -\frac{1}{9}\exp(\frac{2}{d}\sum_{i=1}^{d}\cos x_i) + \sum_{i=1}^{d}(\frac{\sin^2 x_i}{d^2} - \frac{\cos x_i}{d})$ 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



Eigenvalue Problem

- Consider

$$-\Delta u + w \cdot u = \gamma u, \quad x \in \Omega$$
$$u = 0, \quad x \in \partial\Omega$$

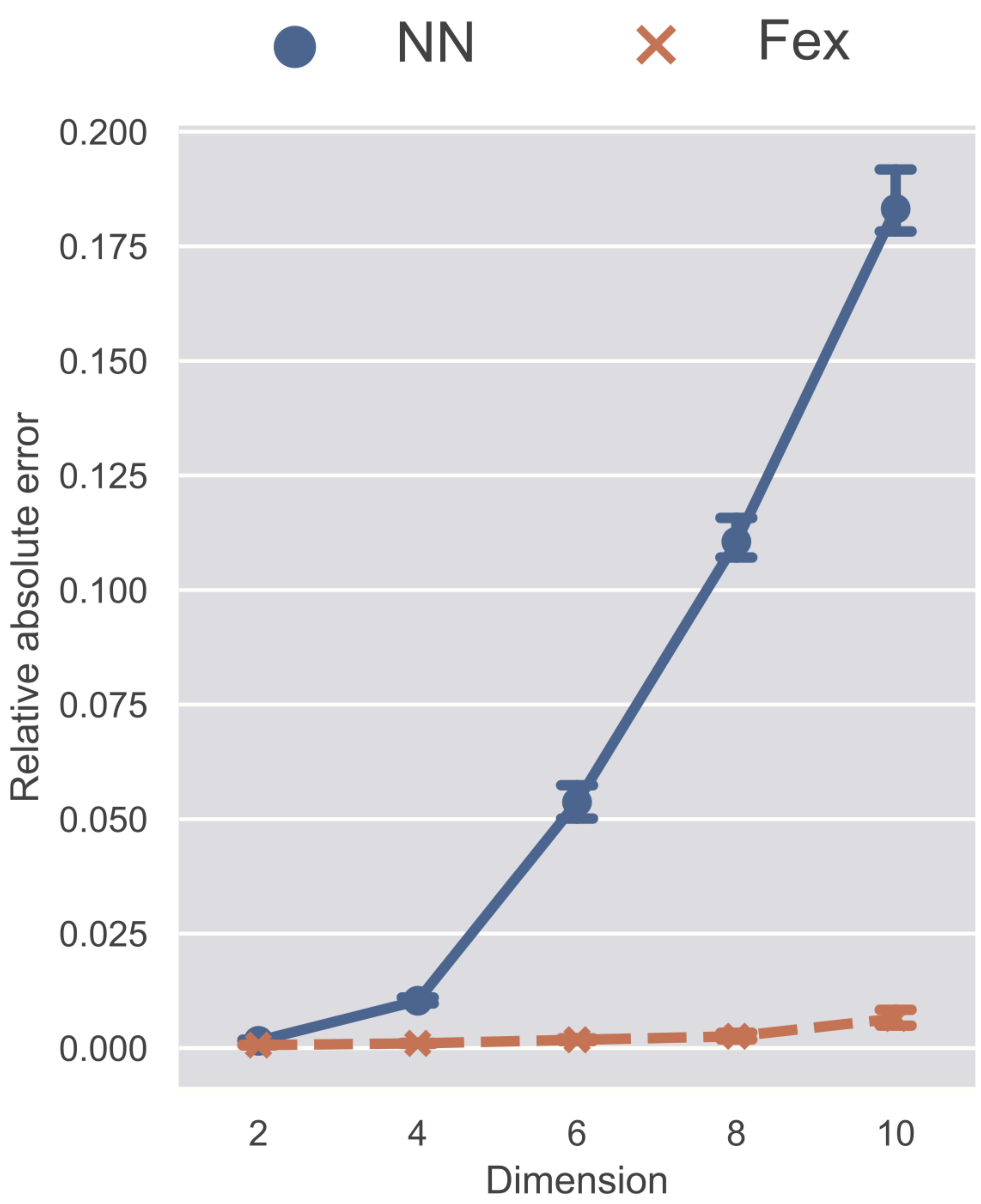
- $\Omega = [-3,3]^d$ and $w = \|x\|_2^2$
- The smallest eigenfunction is $u(x) = \exp(-2\|x\|_2^2)$
- Stochastic optimization (DeepRitz, Weinan E and Bing Yu, 2017):

$$\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \min_{u \in \mathbb{S}_k} \mathscr{I}(u) + \lambda_1 \int_{\partial\Omega} u^2 dx + \lambda_2 \Big(\int_{\Omega} u^2 dx - 1 \Big)^2$$

with Rayleigh quotient

$$\mathscr{I}(u) = \frac{\int_\Omega \|\nabla u\|_2^2 dx + \int_\Omega w \cdot u^2 dx}{\int_\Omega u^2 dx}$$

Eigenvalue Problem



Finite Expression Method

Conclusion

- **Theory:** $O(d^2)$ finite expressions approximate d-dimensional continuous functions to arbitrary accuracy
- **Algorithm**: reinforcement learning solve combinatorial optimization to identify expressions to solve PDEs
- **Advantage:** PDE solver scalable in dimension with high accuracy
- Preprint: Liang and Y. arXiv:2206.10121

Acknowledgement

NSF

Office of Naval Research
USN ONR
Science & Technology

ORACLE LABS