Modeling and Computation in the Space of Language: Symbolic and LLM-Based Approaches

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

Statistical and Computational Challenges in Probabilistic Scientific Machine Learning (SciML)

June 10, 2025

Overview

01 Introduction
Modeling & Computing in the space of Language

02 Finite expression method (FEX)
Concept & theory & formulation

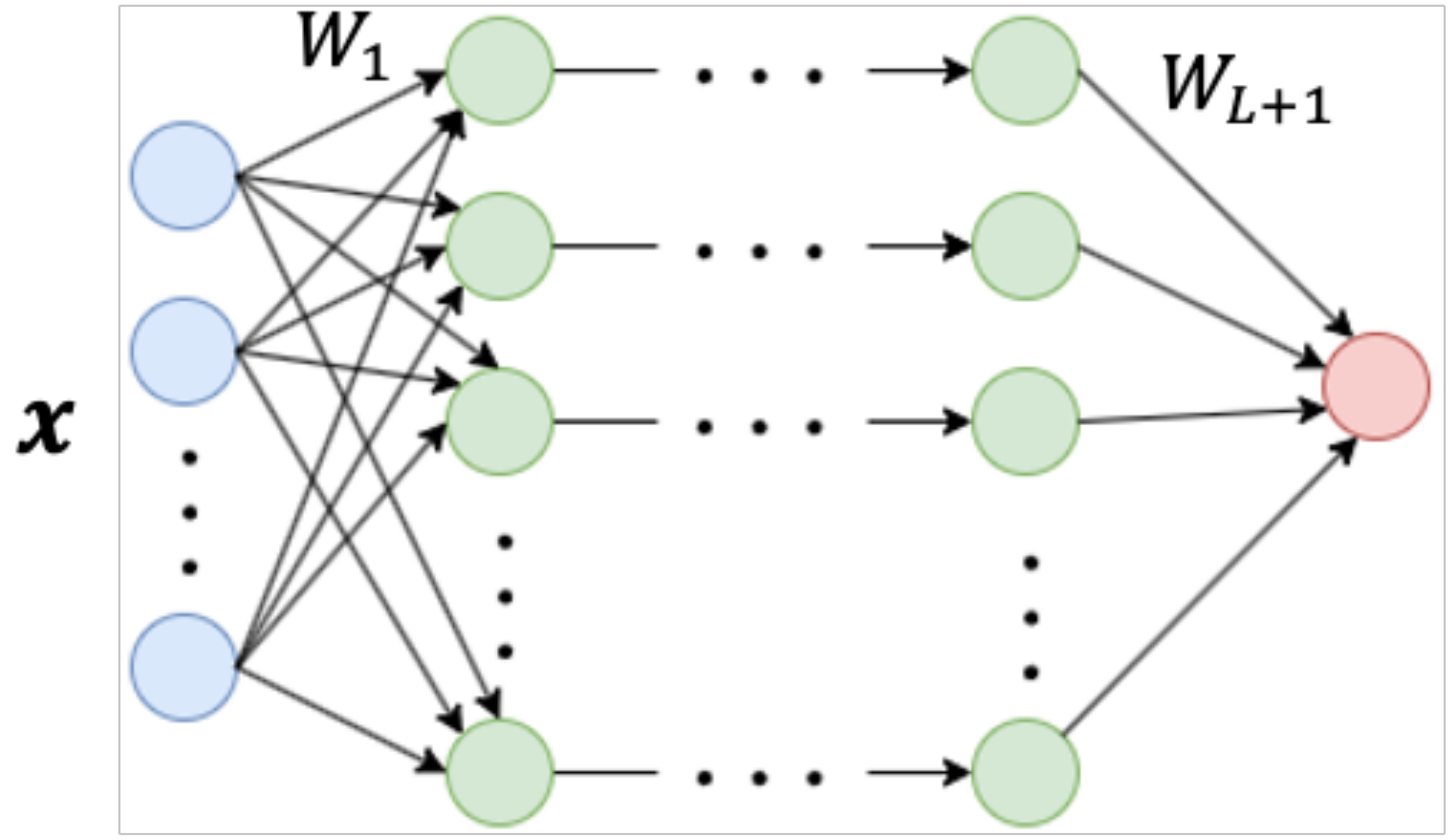
03 Auto-Computing via LLM
Scientific assistant: modeling, planning, coding, computing

04 Summary and future
Efficiency & future vision

Neural network (NN) and its advantages

Neural network

- Fully-connected neural network: $\phi(\boldsymbol{x};\theta) = W_{L+1}\, h_L \circ \cdots \circ h_1(\boldsymbol{x}),\ \ \boldsymbol{x} \in \mathbb{R}^d$
- Hidden layer: $h_\ell(\boldsymbol{x}) = \sigma(W_\ell \boldsymbol{x} + b_\ell)$
- The σ is a nonlinear activation
- Parameter set: $\theta = \{W_\ell\}_{\ell=1}^{L+1} \cup \{b_\ell\}_{\ell=1}^{L}$



Theory and applications

- NN approximation: curse of dimensionality debate (Shen, Y., Zhang, JMLR 2022)
- NN optimization: gap between theory and practice (Na, Y., arXiv:2502.05360)
- Practical applications in high-dimensional problems

Two prototypes for differential equations (DEs)
NN ≈ Target

Learn physical laws as descriptor

Dynamical system: $\frac{dx}{dt} = f(x), \quad x \in \mathbb{R}^d$

- Given historical data $\{x^t\}_{t=0}^T$ and learn a surrogate $\hat{f} \approx f$, e.g, NN $f(x;\theta) \approx f(x)$.
- **Training**: $\min_\theta \frac{1}{\mathrm{T}}\sum_{t=1}^T \left|\left|x^t - \hat{x}^t\right|\right|^2$, with $\hat{x}^t = \mathrm{Integrator}(f(\cdot;\theta), x^{t-1}, \Delta t)$.
- **Prediction**: $x^t = \mathrm{Integrator}(f(\cdot;\theta^*), x^{t-1}, \Delta t), t > T$.

Solve PDE as parametrization

Boundary value problem: $\begin{cases} \mathcal{D}u(x) = f(x), x \in \Omega \subset \mathbb{R}^d \\ \mathcal{B}u(x) = g(x), x \in \partial\Omega \end{cases}$

- Approximate the PDE solution with NN, e.g., $u(x;\theta) \approx u(x)$.
- **Training**: $\min_\theta \left|\left|\mathcal{D}u(x;\theta) - f(x)\right|\right|_{L^2(\Omega)}^2 + \left|\left|\mathcal{B}u(x;\theta) - g(x)\right|\right|_{L^2(\partial\Omega)}^2$.
- Differential operator ← auto-differentiation/finite difference.
- Integral ← Monte Carlo integration.

NN Limitations

- Limited accuracy for engineering applications: $O(10^{-2})$ to $O(10^{-4})$
- Expert knowledge needed/Try-and-error for good architecture
- Poor data efficiency
- Lack of interpretability

Is there an alternative to neural network-based methods that offers

High accuracy

Less data needed

?

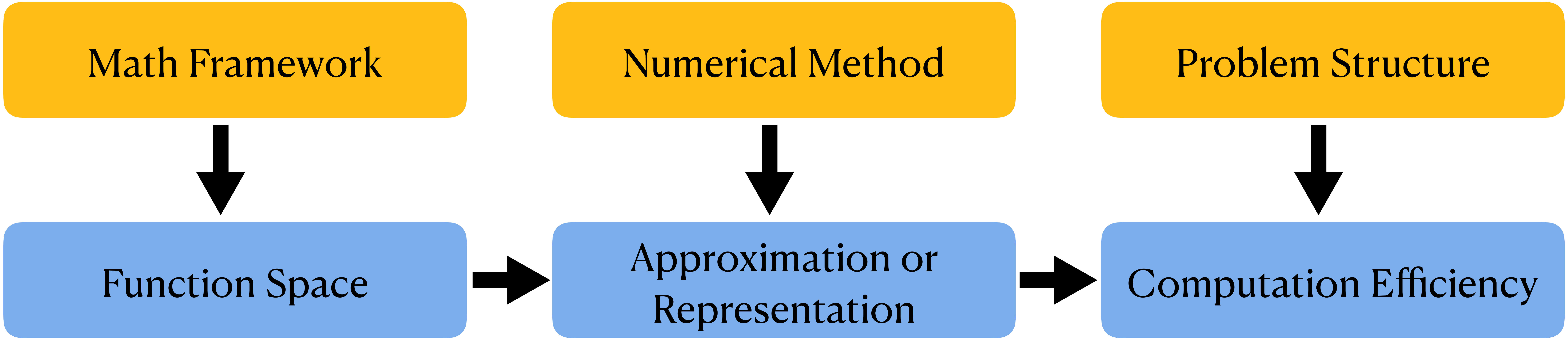
Interpretability

Less reliance on expert

Modeling & Computing in the Right Space

- Sparse Grids
  - "Right Space" - the solution can be approximated well by sparse combinations of basis functions
- Low-Rank Tensor Methods
  - "Right Space" - the solution can be approximated by a sum of products of lower-dimensional functions
- Neural Network Methods
  - "Right Space" - the implicit function space learned by the network during training

Modeling & Computing in the Right Space



- Our research is a big search (optimization) process
- Our "search" is in a space of natural language
- Our "optimization" is mixed-integer programming and gradient-free

Modeling & Computing in the Space of Natural Language

New Paradigm for New Investigation, Method, and Application

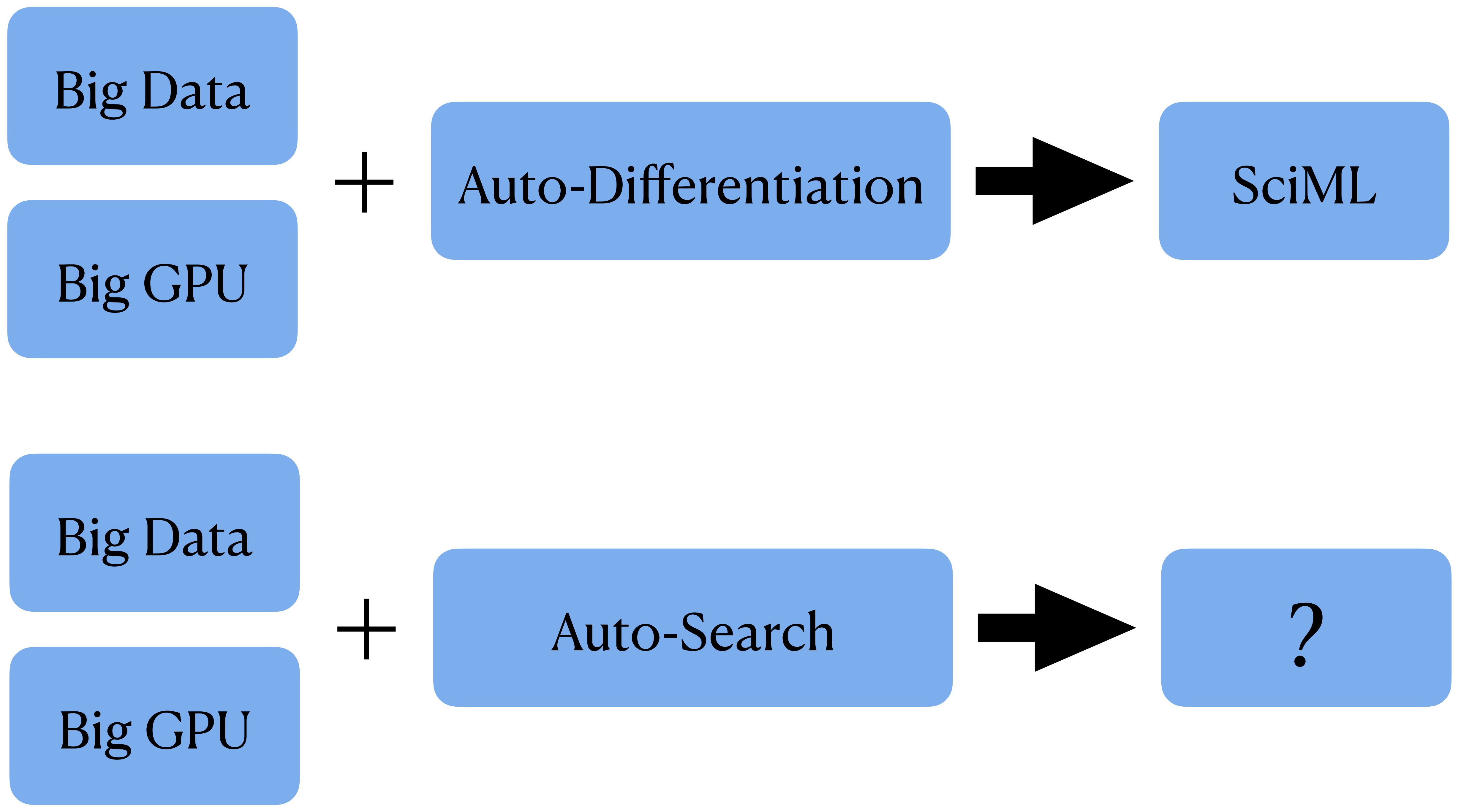
Two Complementary Approaches:

- Symbolic learning (Finite Expression Method)
- Large language model (LLM)

Applications:

- Search for a solution
- Search for a mathematical model
- Search for a computational algorithm
- Search for executable code
- ...

Our vision:



Two Complementary Approaches:

- **Symbolic learning (Finite Expression Method)**
- Large language model (LLM) for modeling and computing assistant

Finite Expression Method (FEX) Methodology

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) Theory

Liang and Y. arXiv:2206.10121

Advantages in Real Analysis: "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) Practice

Advantages in Practice:

- Leverage the power of descriptive structures of problems

Question:

- How to do computation with description?

Answers:

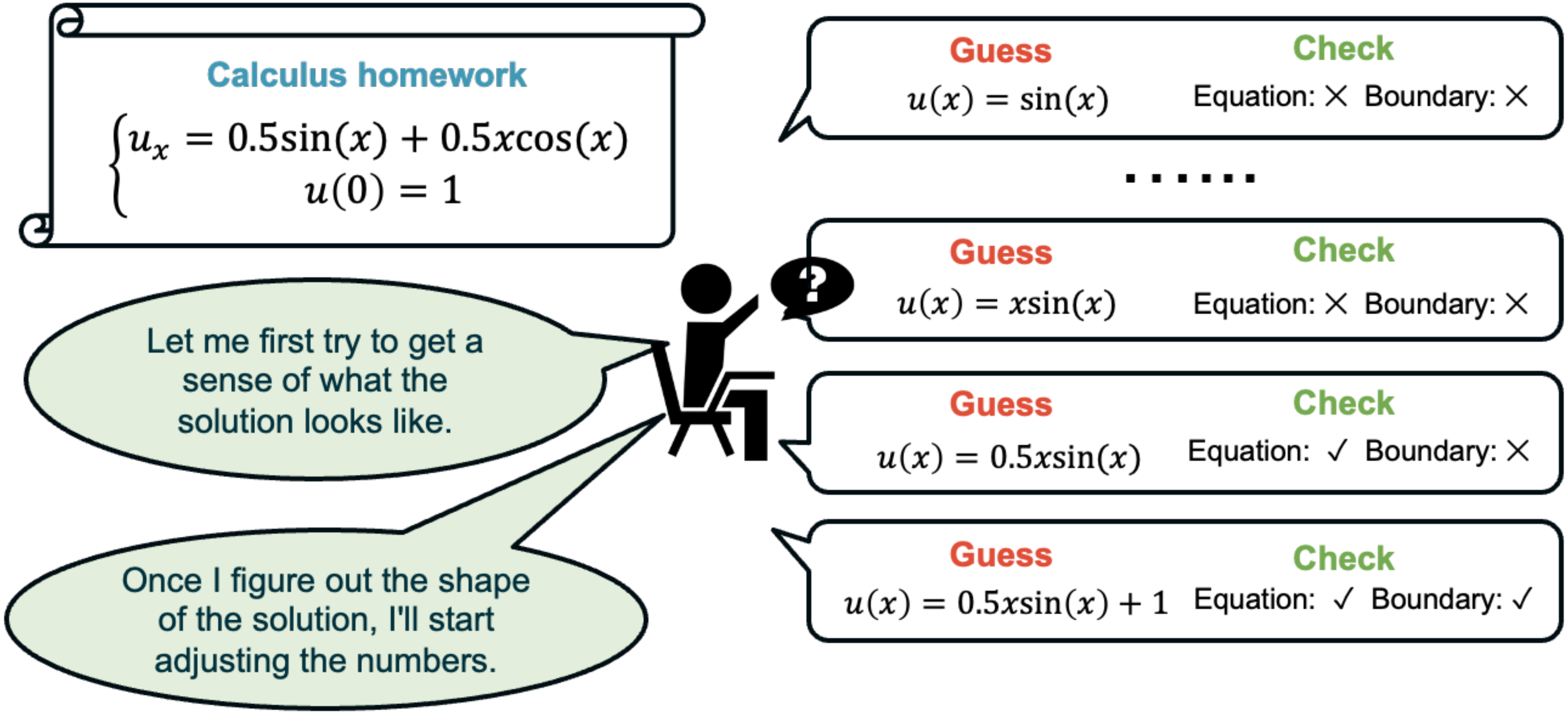
- **Symbolic machine learning**
- Large language models
- Bayesian perspective

Ideas: Automatic Trial-and-Error for Structures and Refinement

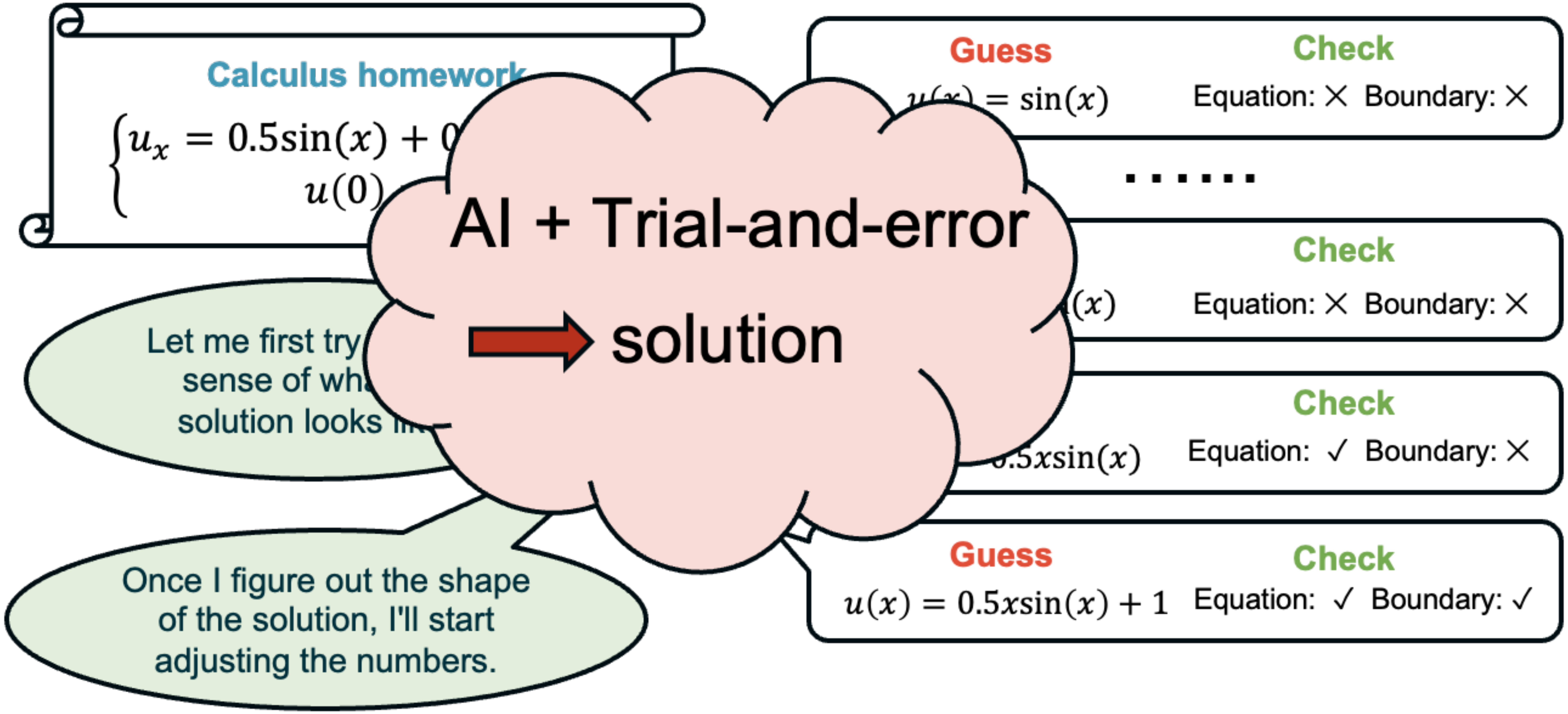
Calculus homework

$$\begin{cases} u_x = 0.5\sin(x) + 0.5x\cos(x) \\ u(0) = 1 \end{cases}$$

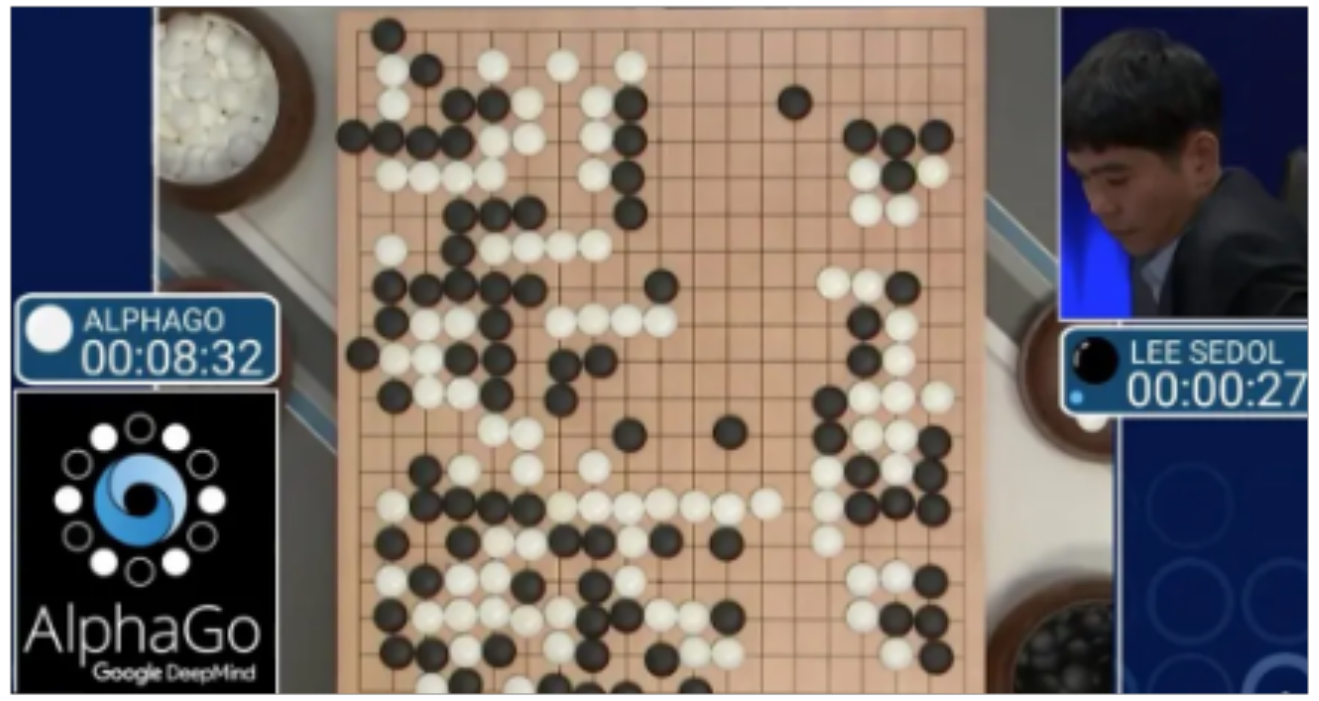
Ideas: Automatic Trial-and-Error for Structures and Refinement



Ideas: Automatic Trial-and-Error for Structures and Refinement



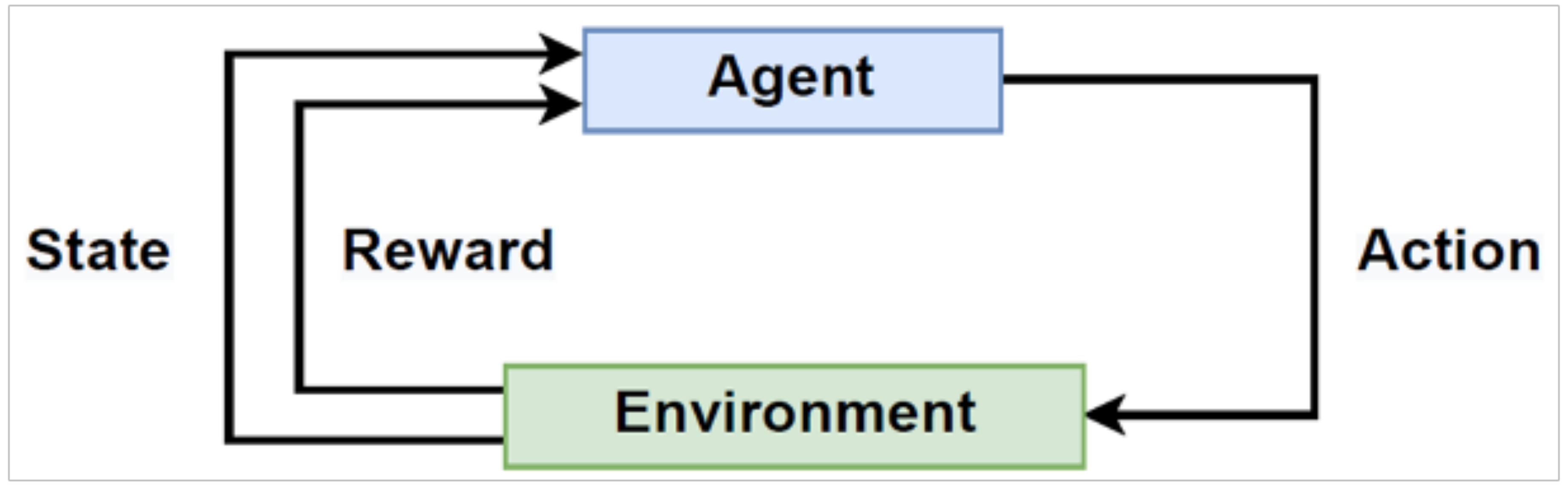
Reinforcement Learning (RL)



AlphaGo (source: bbc.com)

Playing video games (source: CLVR lab @ USC)

- Reinforcement learning: train AI agent to make decision



- Objective is to learn π : $\max_{\pi} \mathbb{E}_{\tau \sim \pi} R(\tau)$

  π is a policy, τ is an episode, $R(\tau)$ is cumulative reward

- Methods: Policy gradient, etc.

Finite Expression Method for Solving PDEs

Liang and Y. arXiv:2206.10121

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

Numerical Comparison

Liang and Y. arXiv:2206.10121

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

○ The right space: solutions with simple descriptive structures

Solving High-Dimensional PDEs with FEX

Poisson equation

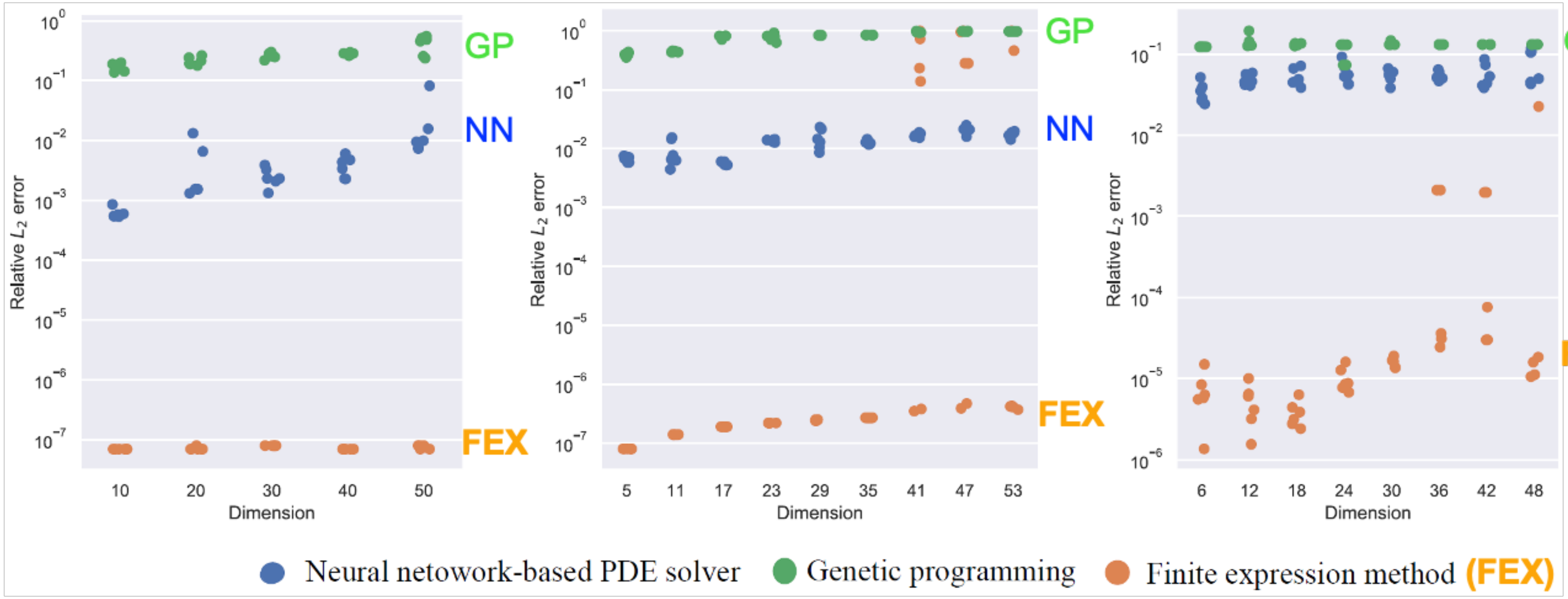
$$\Omega = [0,1]^d$$
$$-\Delta u = f \text{ for } x \in \Omega \text{ and } u = g \text{ on } \partial\Omega$$
True solution: $u(\boldsymbol{x}) = \frac{1}{2}\sum_{i=1}^{d} x_i^2$

Linear conservation law

$$T\times\Omega = [0,1]\times[-1,1]^d$$
$$\frac{\pi d}{4}u_t - \sum_{i=1}^{d} u_{x_i} = 0 \text{ and } u(0,x) = \sin\left(\frac{\pi}{4}\sum_{i=1}^{d} x_i\right)$$
True solution: $u(\boldsymbol{x}) = \sin(t + \frac{\pi}{4}\sum_{i=1}^{d} x_i)$

Nonlinear Schrödinger equation

$$\Omega = [-1,1]^d$$
$$-\Delta u + u^3 + Vu = f \text{ for } x \in \Omega$$
True solution: $u(\boldsymbol{x}) = \exp(\sum_{i=1}^{d} \cos(x_i)/d)$



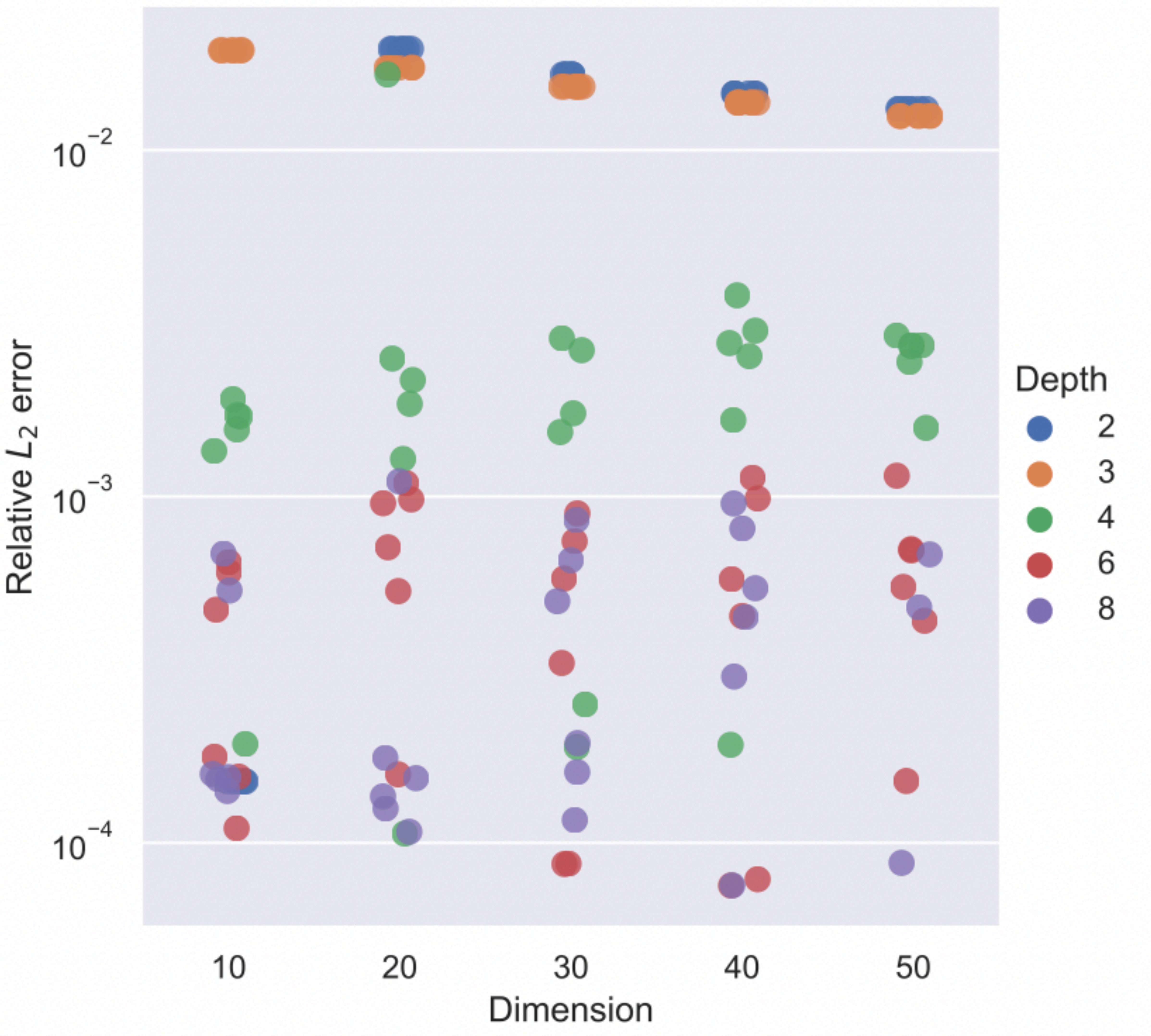
Liang and Y. arXiv:2206.10121

Poisson Equation

Liang and Y. arXiv:2206.10121

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)



FEX for Partial Integral Differential Equations

Hardwick, Liang, Y., arxiv:2410.00835

$$\frac{\partial u}{\partial t} + b \cdot \nabla u + \frac{1}{2}\mathrm{Tr}(\sigma\sigma^T H(u)) + \mathrm{A}u + f = 0$$

$$u(T, \cdot\,) = g(\,\cdot\,)$$

$$\mathrm{A}u(t,x) = \int_{\mathbb{R}^n} (u(t, x + G(x,z)) - u(t,x) - G(x,z)\cdot \nabla u(t,x))\nu(dz)$$

$G(x,z) \in \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d$, and ν is a Levy measure associated with a Poisson random measure.

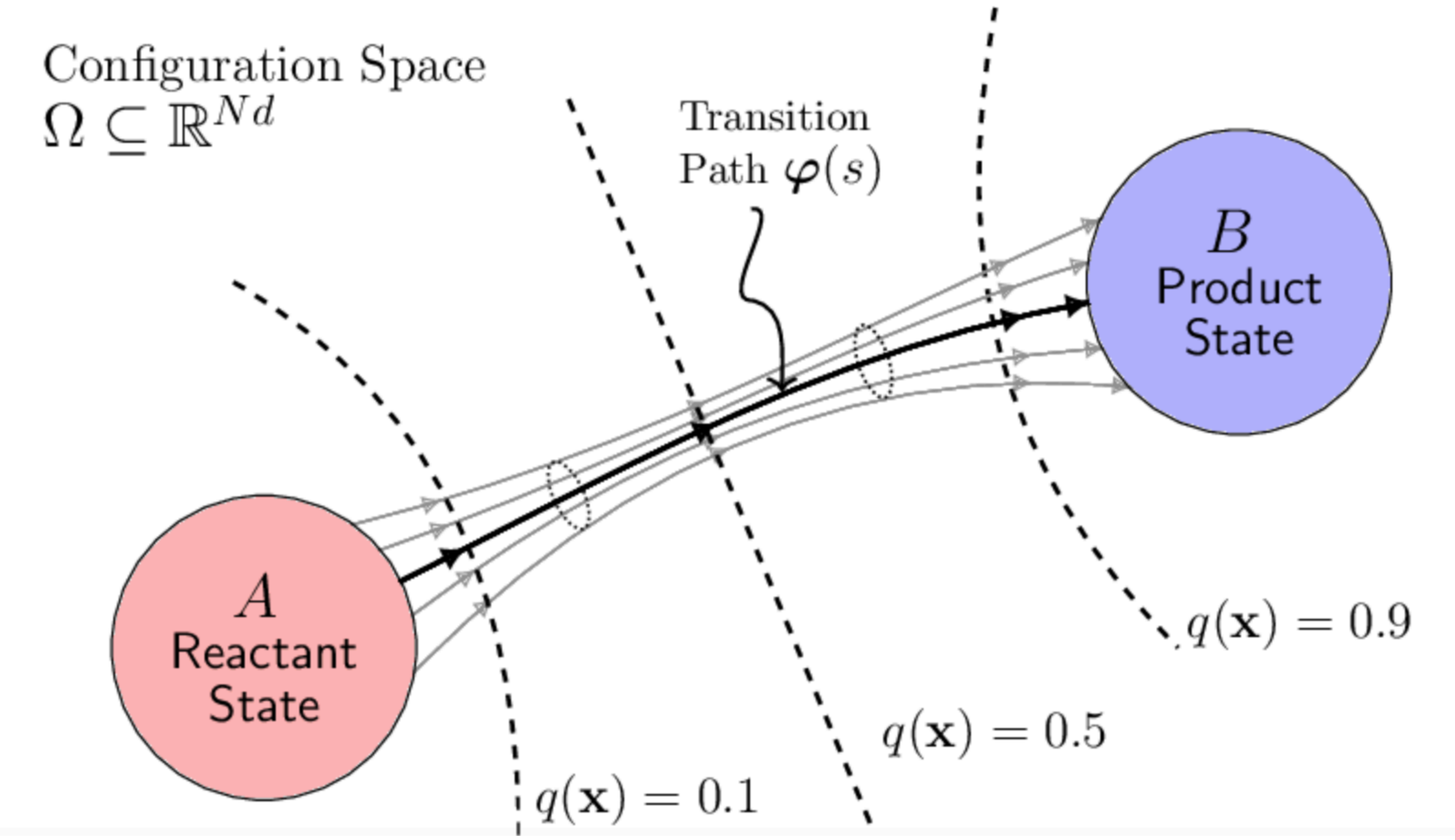
Dimension	2	4	6	8	10	20	30
FEX-PG	2.99e-7	3.17e-7	5.16e-7	7.26e-7	2.05e-7	8.02e-7	4.49e-7
TD-NN [23]	0.00954	0.00251	0.00025	0.00671	0.01895	0.00702	0.01221

Dimension	40	50	60	70	80	90	100
FEX-PG	9.05e-7	4.27e-7	4.55e-7	3.54e-7	5.89e-7	6.44e-7	5.64e-7
TD-NN [23]	0.00956	0.00219	0.00944	0.00044	0.00277	0.00460	0.00548

Committor Function for Rare Events

Song, Cameron, Yang arXiv:2306.12268, SISC, 2025

$$
\begin{cases}
(Lq)(\mathbf{x}) = 0 & \text{for } x \notin A \cup B \\
q(\mathbf{x}) = 0 & \text{for } \boldsymbol{x} \in A \\
q(\mathbf{x}) = 1 & \text{for } \boldsymbol{x} \in B\,.
\end{cases}
$$



where L is the infinitesimal generator of the process

defined as:

$$Lq = -\beta^{-1}\Delta q + \nabla V \cdot \nabla q$$

Previous work

- Diffusion map, Coifman et al. (2008), Lai & Lu (2018), Evans et al. (2023)
- Neural network, Khoo et al. (2019), Li et al. (2019), Li et al. (2022)
- Tensor network, Chen et al. (2023)

Committor Function for Rare Events

Song, Cameron, Yang arXiv:2306.12268, SISC, 2025

Difficulty

- Curse of dimensionality: dimension $\propto$ number of atoms

Physical Structure

- Low-dimensional structure: a small number of collective variables

Machine Learning

- FEX to identify the low-dimensional structure
- Transfer a high-dimensional problem into a low-dimensional one

Committor Function for Rare Events

Song, Cameron, Yang arXiv:2306.12268, SISC, 2025

Example: Double-Well potential

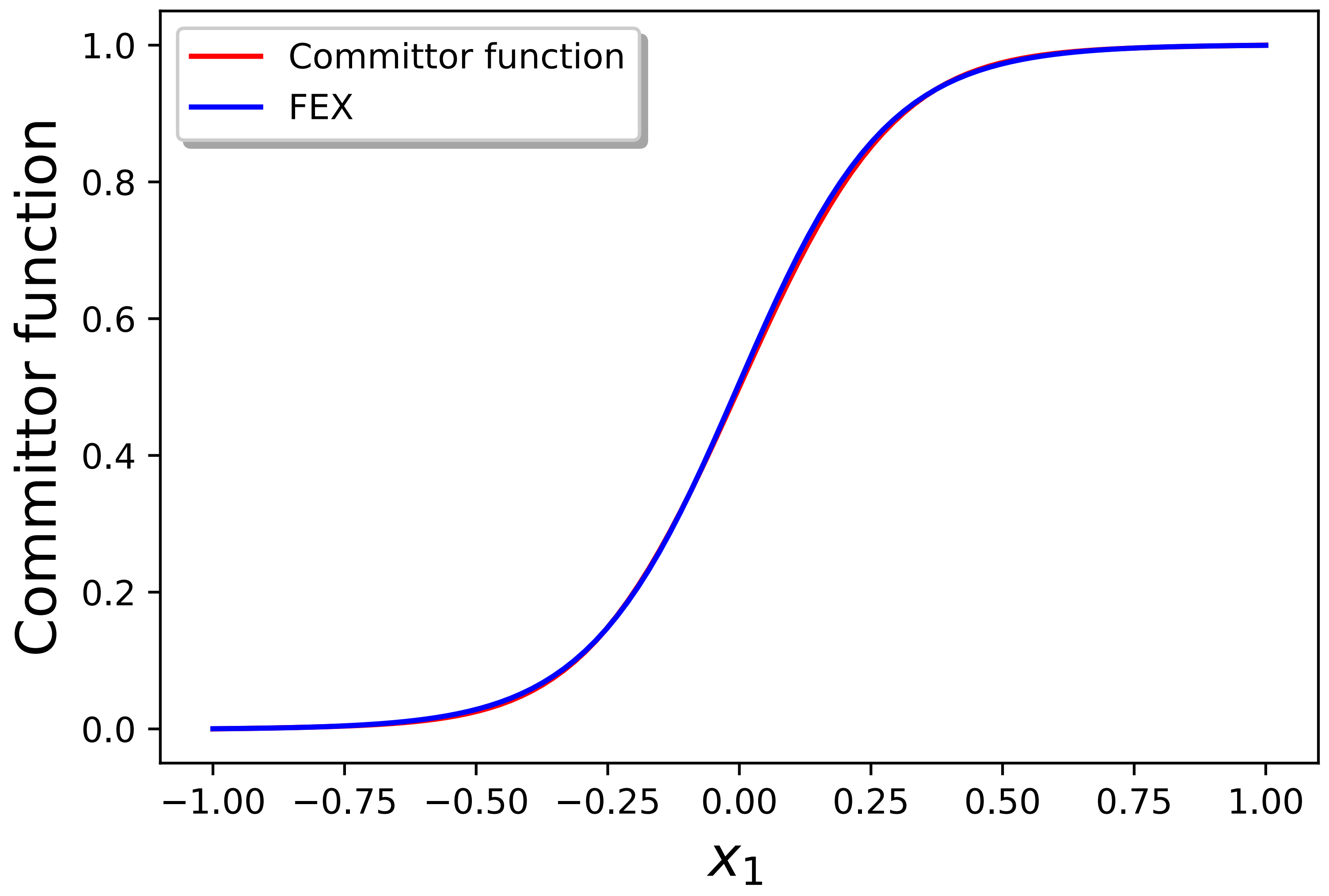
$$V(\mathbf{x}) = \underbrace{\left(x_1{}^2 - 1\right)^2}_{\text{collective variable}} + 0.3\sum_{i=2}^{d} x_i^2$$

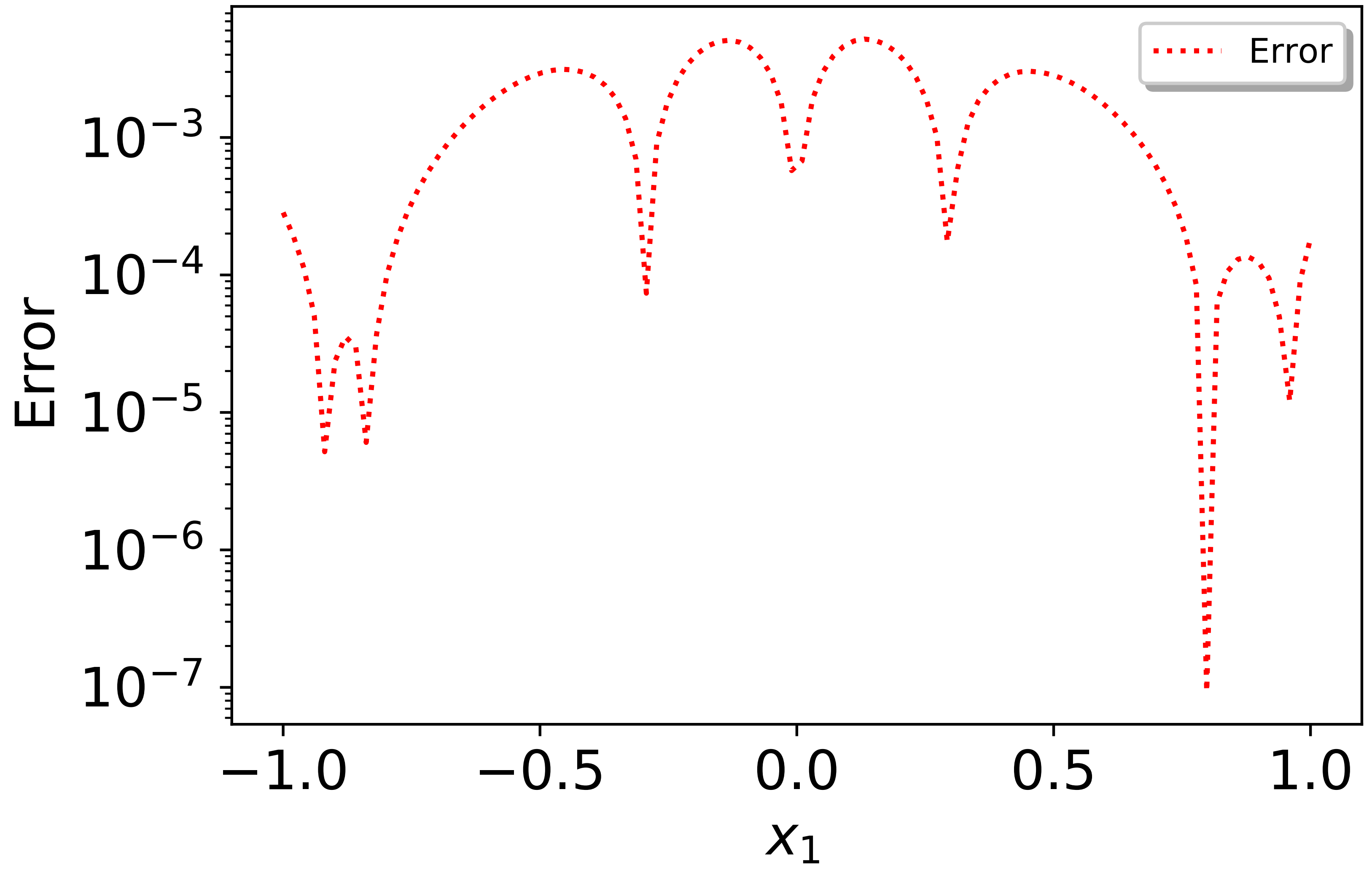
with

$$A = \left\{x \in \mathbb{R}^d \mid x_1 \leq -1\right\}, \quad B = \left\{x \in \mathbb{R}^d \mid x_1 \geq 1\right\}$$

The ground truth solution is $q(\mathbf{x}) = f(x_1)$

$$\frac{d^2 f\left(x_1\right)}{dx_1^2} - 4x_1\left(x_1^2 - 1\right)\frac{df\left(x_1\right)}{dx_1} = 0, \quad f(-1) = 0, \quad f(1) = 1$$





Committor Function for Rare Events

Song, Cameron, Yang arXiv:2306.12268, SISC, 2025

FEX identifies the following representation

Eqn 1: $\alpha_{1,1}x_1 + \ldots + \alpha_{1,10}x_{10} + \beta_1$

Eqn 2: $\alpha_{2,1}\tanh(x_1) + \ldots + \alpha_{2,10}\tanh(x_{10}) + \beta_2$

$\mathcal{J}(\mathbf{x}) = \alpha_3 \tanh(\text{Eqn 1} + \text{Eqn 2}) + \beta_3$

where $\alpha_3 = 0.5,\ \beta_3 = 0.5$

	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8	α_9	α_{10}	β
Eqn 1	**1.6798**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Eqn 2	**1.9039**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

FEX discovers that $q(\mathbf{x}) = f(x_1)$ and hence transfers a high-dimensional problem into a low-dimensional one.

FEX for Learning Physical Laws

- **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

FEX for Learning Physical Laws

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}

Finite Expression Method (FEX) Practice

Advantages:

- Leverage the power of descriptive structures of problems

Question:

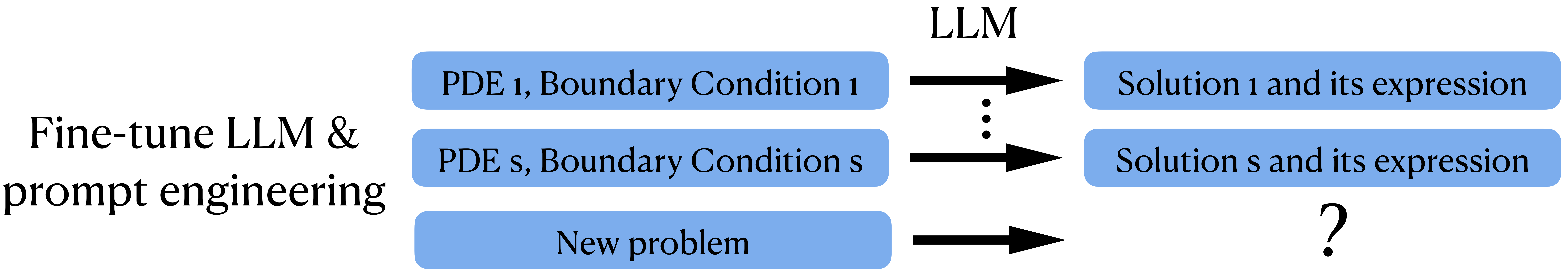
- How to do computation with description?

Answers:

- Symbolic machine learning
- **Large language models**
- Bayesian perspective

Unraveling Symbolic Structures in FEX with LLMs

Bhatnagar, Liang, Patel, Y., arXiv:2503.09986



$u(x)$	Method	Binary Size	Unary Size	Iters	Time [m]	Error
$4\cos(4x^2\cos(x_0))$	**LLM-informed**	1	2	4.25	8.25	0
	Uninformed	3	9	167	340	0
$4x_1^3+4x_1^2+2\cos(4x_1^3\cos(x_0))$	**LLM-informed**	2	4	102	286	10^{-8}
	Uninformed[6]	3	9	2000+	2400+	N/A
$128x_2^3+2x_2e^{4x_2^4}$	**LLM-informed**	2	4	34.5	40.5	4×10^{-7}
	Uninformed	3	9	90	186	6×10^{-7}
$64x_1^2e^{2x_2}$	**LLM-informed**	1	3	15.5	21	3×10^{-8}
	Uninformed	3	9	103.5	161	3×10^{-8}

Finite Expression Method (FEX) Practice

Advantages:

- Leverage the power of descriptive structures of problems

Question:

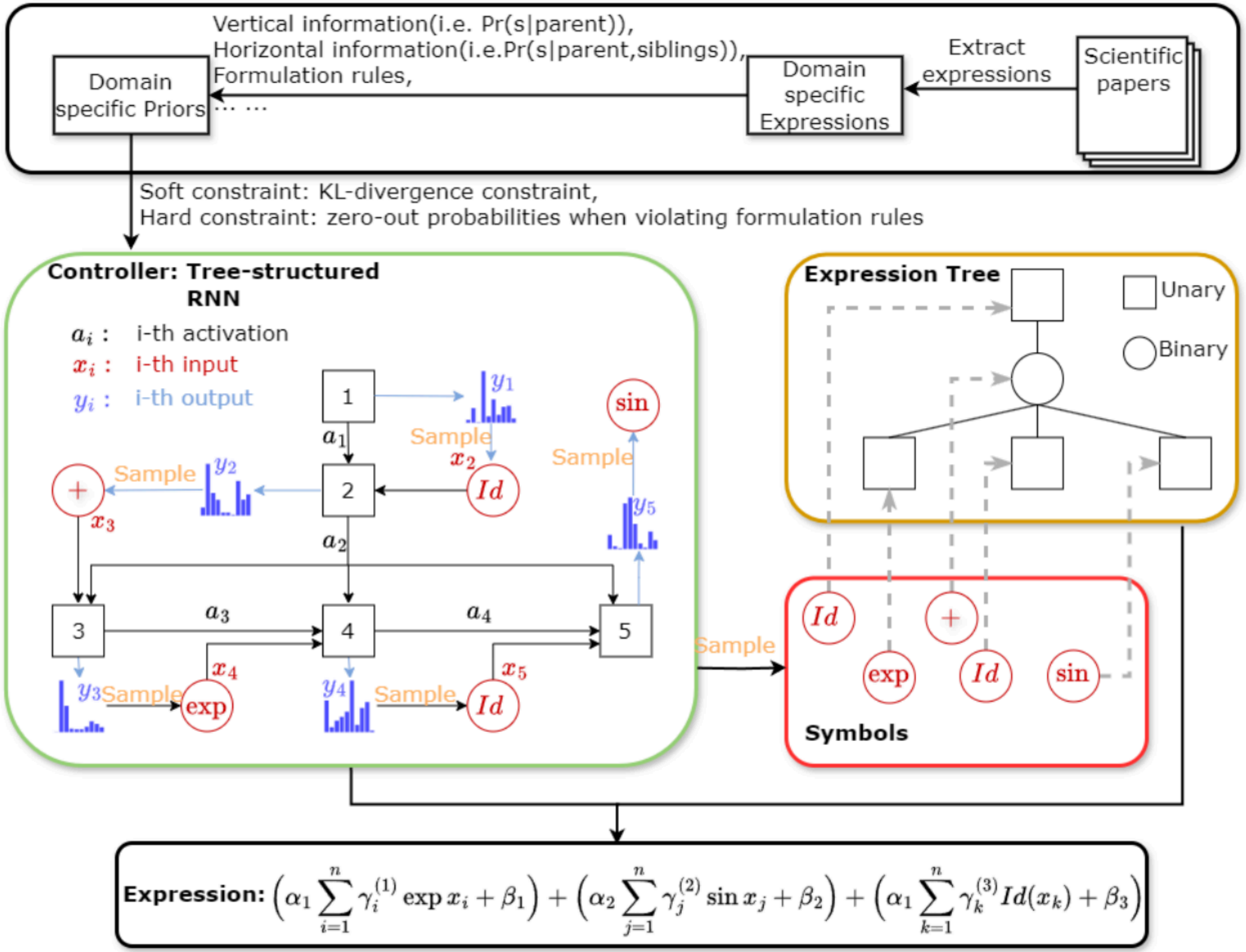
- How to do computation with description?

Answers:

- Symbolic machine learning
- Large language models
- **Bayesian perspective**

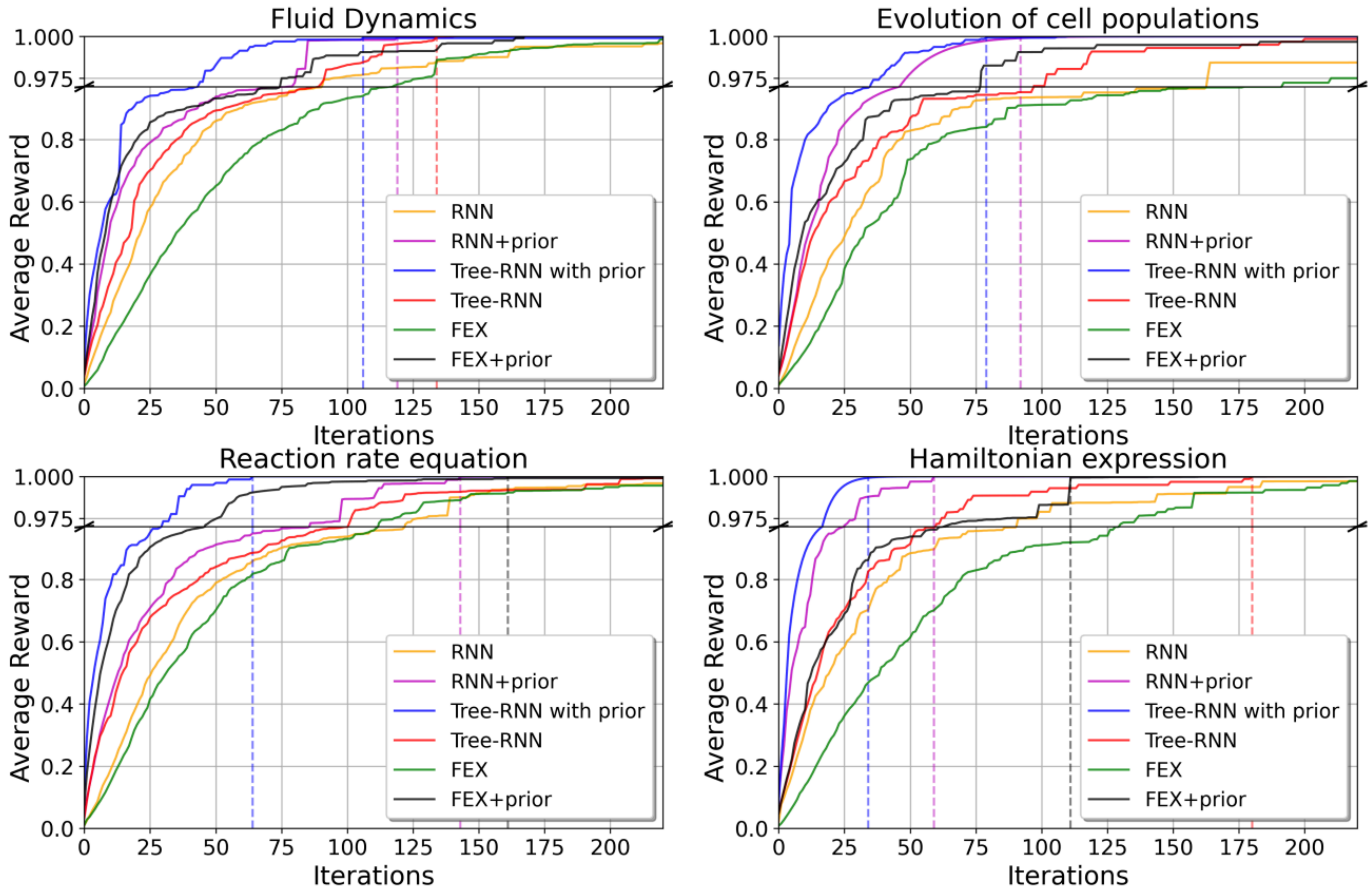
Bayesian Symbolic Learning

Huang, Wen, Adusumilli, Choudhary, Y., arXiv:2503.09592

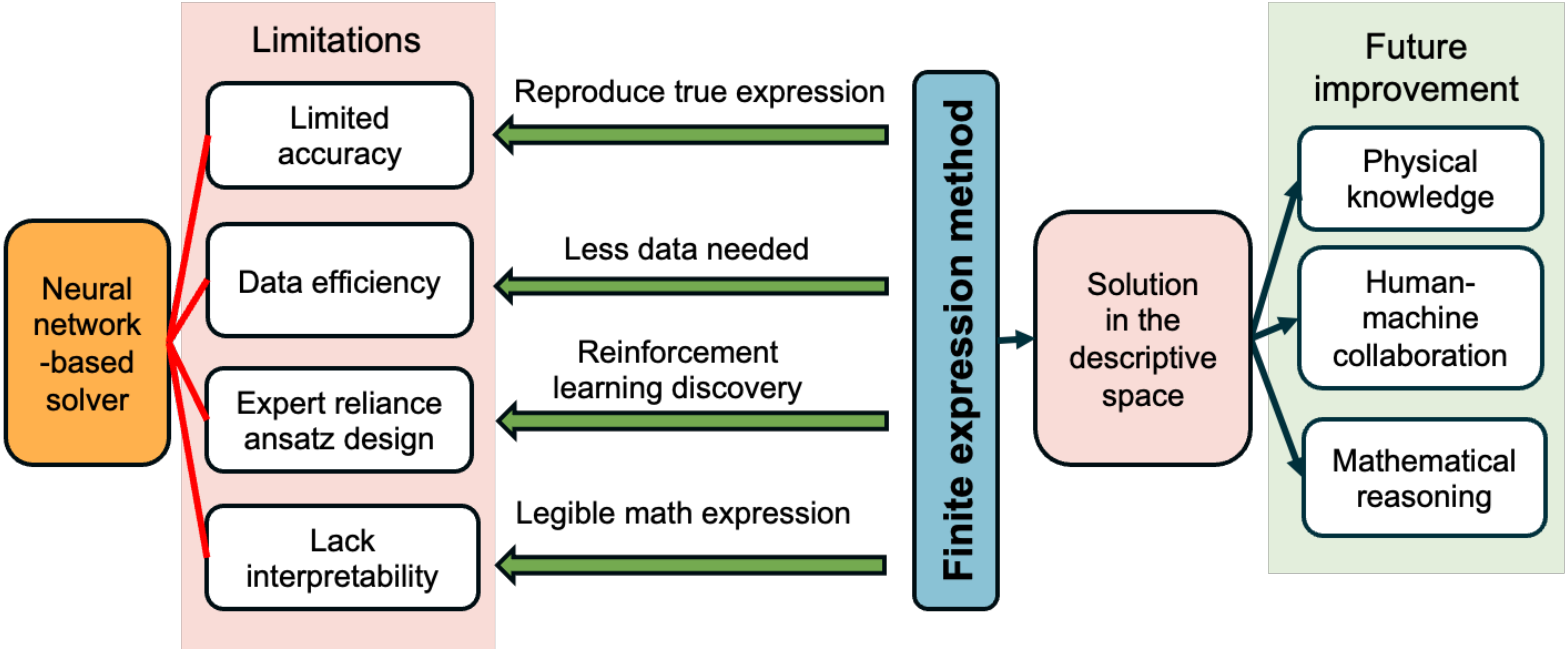


Bayesian Symbolic Learning

Huang, Wen, Adusumilli, Choudhary, Y., arXiv:2503.09592



Summary & Discussion



Two Complementary Approaches:

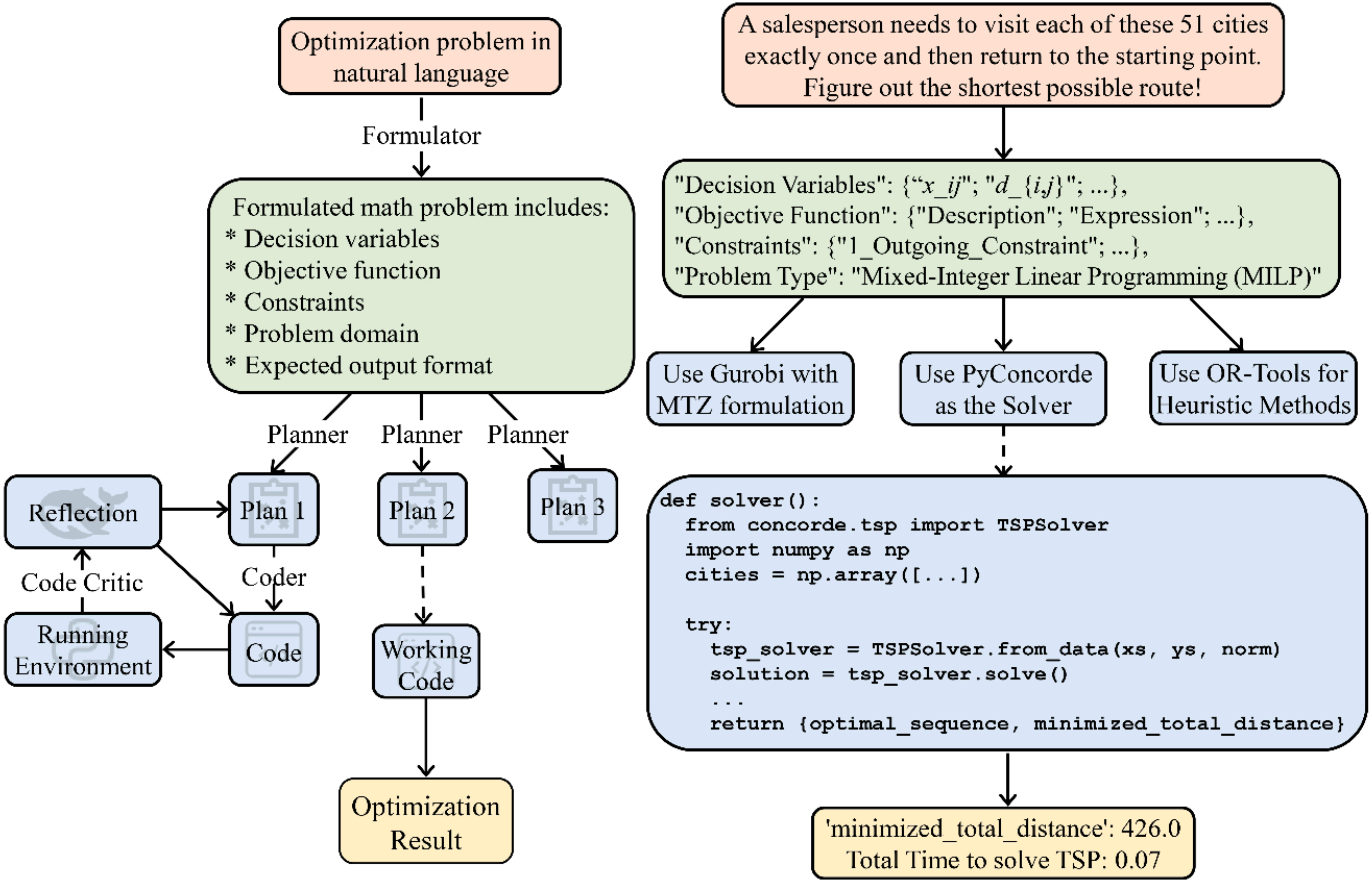
- Symbolic learning (Finite Expression Method)
- **Large language model (LLM) for modeling and computing assistant**

LLM Agents for Modeling & Computing from Natural Language

OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

- Our research is a big search (optimization) process
- Our “search” is in a space of natural language
- Our “optimization” is mixed-integer programming and gradient-free
- **Example**: automatic optimization modeling, solving, and testing

Overview and New Features of OptimAI



LLM Agents for Modeling & Computing from Natural Language

OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

Table 1: Comparison of Functional Capabilities between OptimAI and Prior Methods.

Functional Capabilities	OptiMUS	Optibench	CoE	OptimAI
Natural language input	✗	✓	✓	✓
Planning before coding	✗	✗	✓	✓
Multi-solver support	✗	✗	✗	✓
Switching between plans	✗	✗	✗	✓
Code generation	✓	✓	✓	✓
Distinct LLM collaboration	✗	✗	✗	✓

LLM Agents for Modeling & Computing from Natural Language

OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

Table 2: Previous work on using LLMs for optimization.

Work	Dataset Proposed	Size	Problem Type(s)
NL4Opt Competition [8]	NL4Opt	289	LP
Chain-of-Experts (CoE) [9]	ComplexOR	37	LP, MILP
OptiMUS [10, 11, 12]	NLP4LP	67	LP, MILP
Optibench [13]	Optibench	605	LP, NLP, MILP, MINLP
OR-LLM-Agent[14]	OR-LLM-Agent	83	LP, MILP

Abbreviations: LP - Linear Programming, NLP - Nonlinear Programming, MI - Mixed-Integer.

LLM Agents for Modeling & Computing from Natural Language

OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

Table 3: Accuracy comparison between OptimAI and state-of-the-art methods.

Agent \ Dataset	NLP4LP	Optibench Linear w/o Tab.	Optibench Linear w/ Tab.	Optibench Nonlin. w/o Tab.	Optibench Nonlin. w/ Tab.
OptiMUS [11]	71.6%	-	-	-	-
Optibench [13]	-	75.4%	62.5%	42.1%	32.0%
Ours w/ GPT-4o	79.1%	81.2%	73.8%	72.0%	48.0%
Ours w/ GPT-4o+o1-mini	**88.1%**	84.2%	**80.0%**	77.3%	56.0%
Ours w/ QwQ (by Qwen)	79.1%	86.2%	77.5%	**81.6%**	50.0%
Ours w/ DeepSeek-R1	82.1%	**87.4%**	78.8%	79.5%	**60.0%**

All evaluations were conducted under a zero-shot prompting setting. GPT-4o+o1-mini refers to using o1-mini as the planner while employing GPT-4o for all other roles.

LLM Agents for Modeling & Computing from Natural Language

OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

Table 4: Generalization of OptimAI across NP-hard combinatorial optimization problems.

	Math Programming	TSP	JSP	Set Covering
OptimAI	✓	✓	✓	✓
OptiMUS	✓	✗	✗	✗
Optibench	✓	✗	✗	✗

Traveling salesman problem (TSP), job shop scheduling problem (JSP), and set covering problem.

LLM Agents for Modeling & Computing from Natural Language

OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

Table 5: Synergistic effects of combining heterogeneous LLMs.

Planner \ Remaining Roles	Llama 3.3 70B	DeepSeek-R1 14B	Gemma 2 27B
Llama 3.3 70B	59%	54%	54%
DeepSeek-R1 14B	**68%**	50%	41%
Gemma 2 27B	**77%**	**59**%	54%

LLM Agents for Modeling & Computing from Natural Language

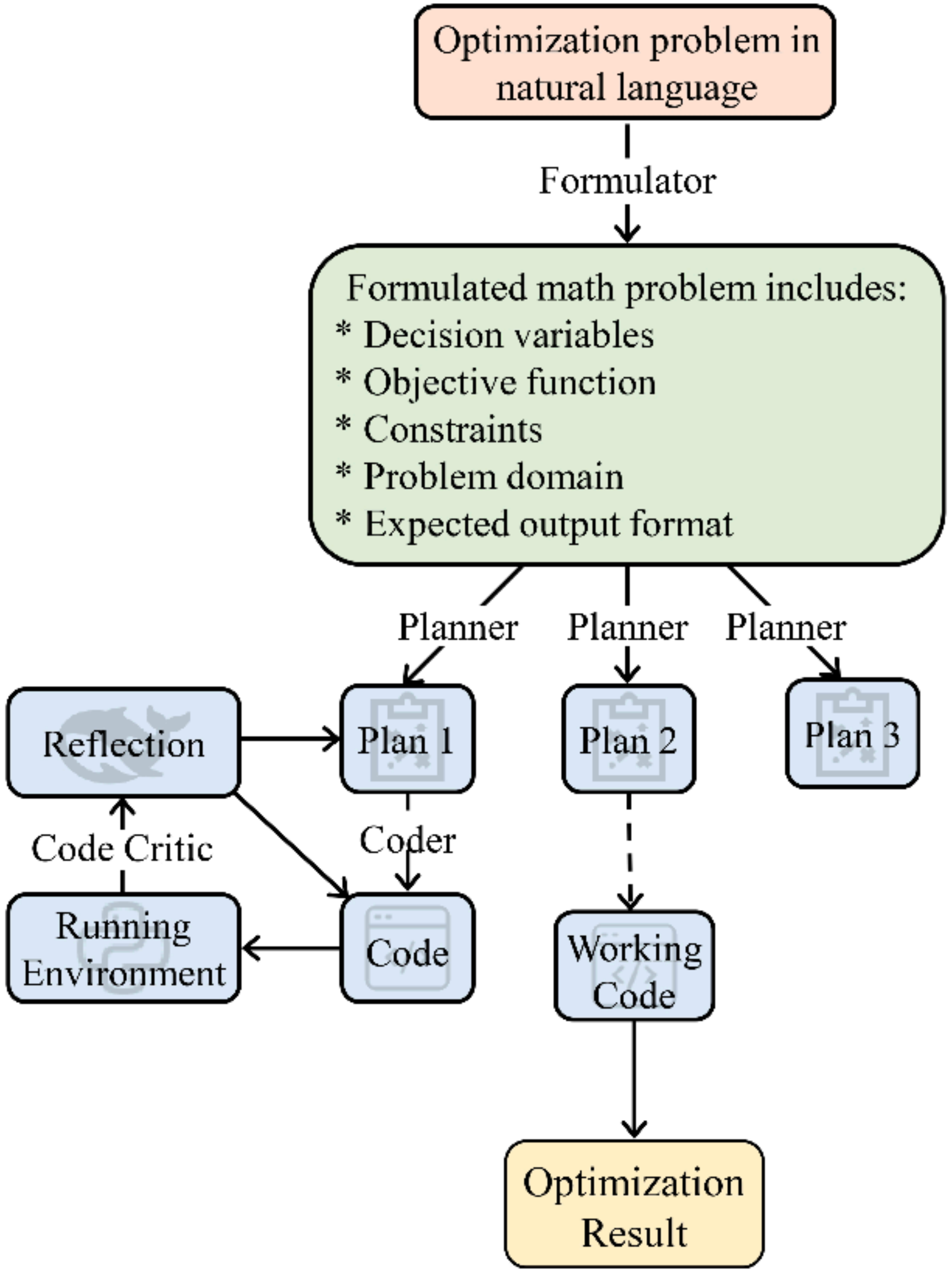
OptimAI: Thind, Sun, Liang, Y., arXiv:2504.16918

Table 6: Ablation study of OptimAI design.

Formulator	Planner	Code Critic	Revisions	Executability	Productivity
✓	✓	✓	**1.7**	**3.6**	**6.8**
✗	✓	✓	2.0	3.2	6.3
✓	✗	✓	7.8	3.1	1.2
✓	✓	✗	6.2	3.3	2.2

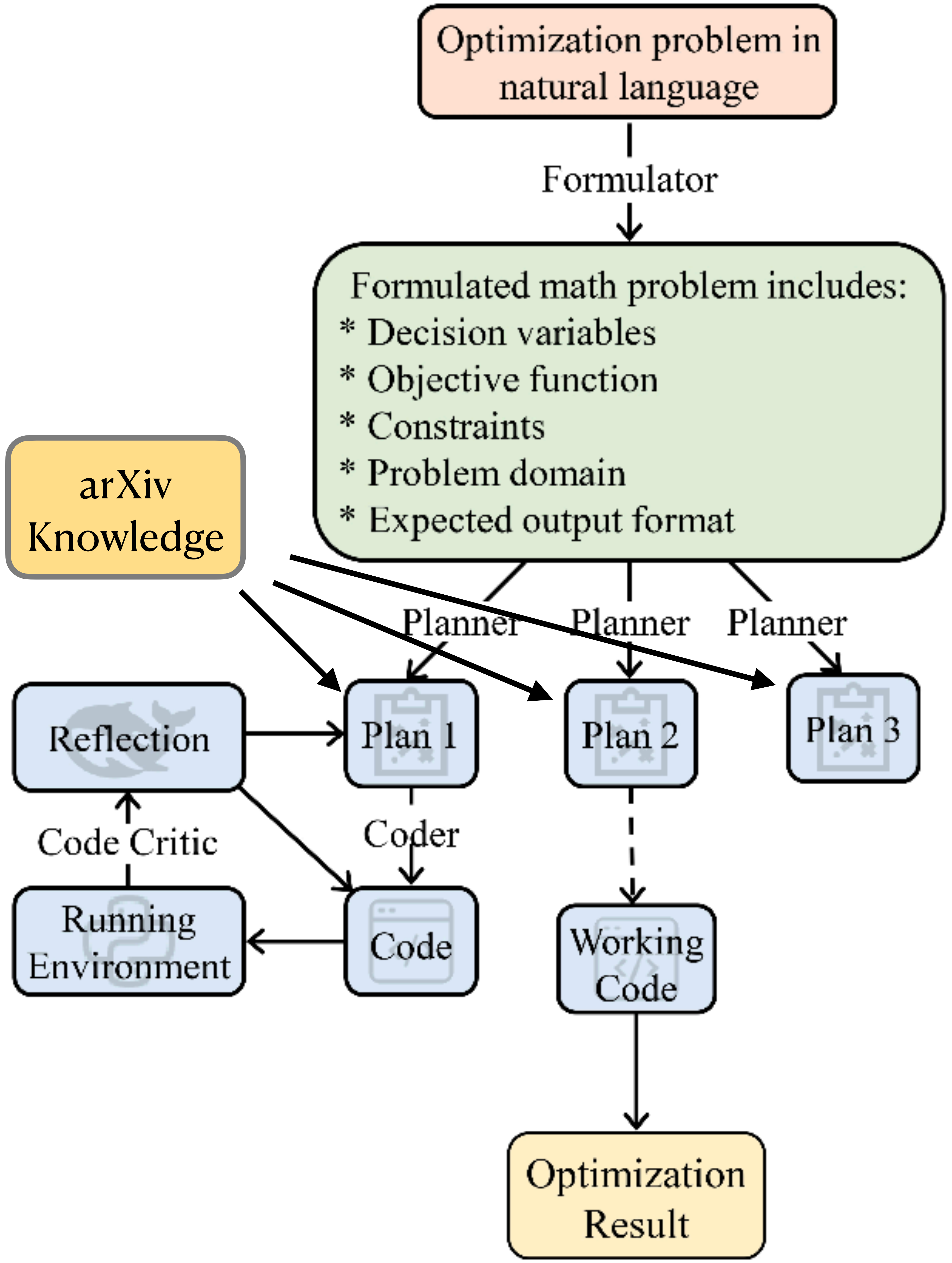
OptimAI Roadmap:

- **Non-expert users for LP, NLP, MILP, MINLP**
- Bayesian approach with arXiv knowledge
- Reasoning with chain of thoughts
  - Algorithm analysis
  - Coding analysis
- Optimization AI assistant with human in the loop



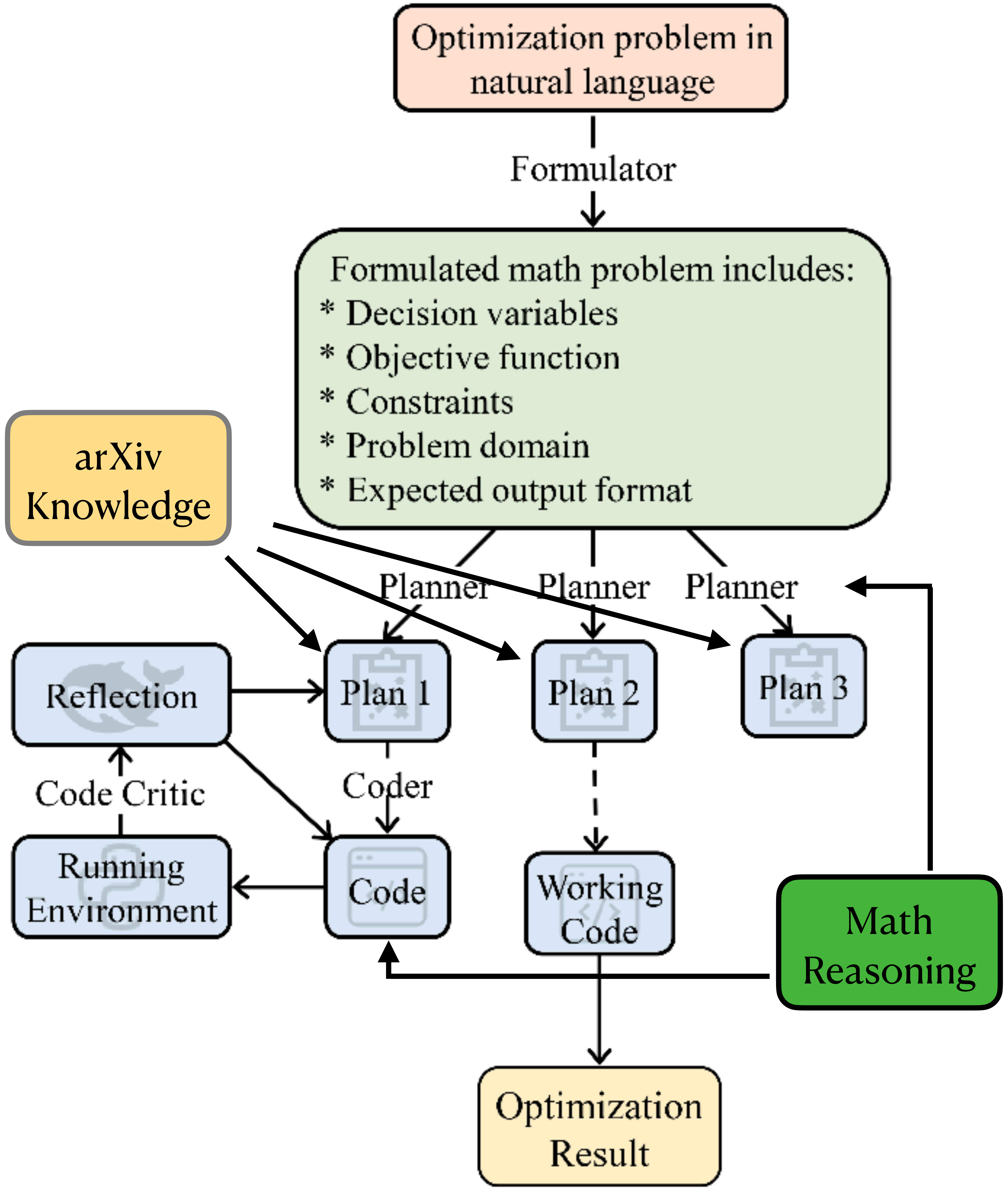
OptimAI Roadmap:

- Non-expert users for LP, NLP, MILP, MINLP
- **Bayesian approach with arXiv knowledge**
- Reasoning with chain of thoughts
  - Algorithm analysis
  - Coding analysis
- Optimization AI assistant with human in the loop



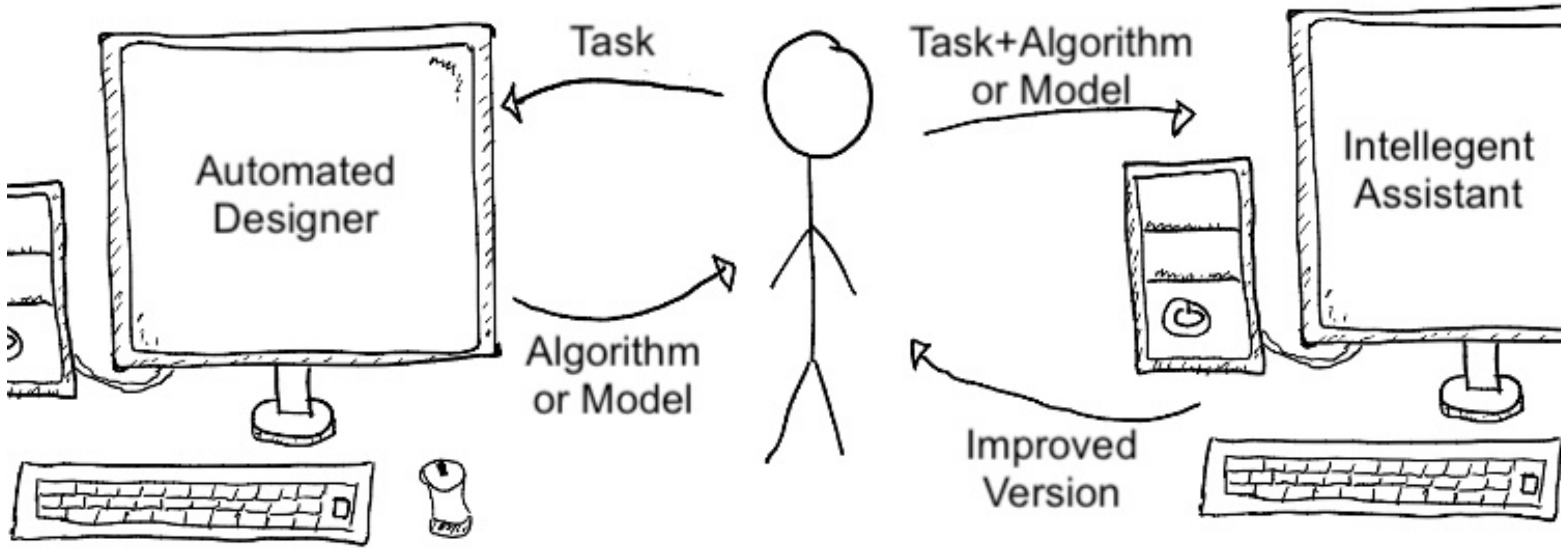
OptimAI Roadmap:

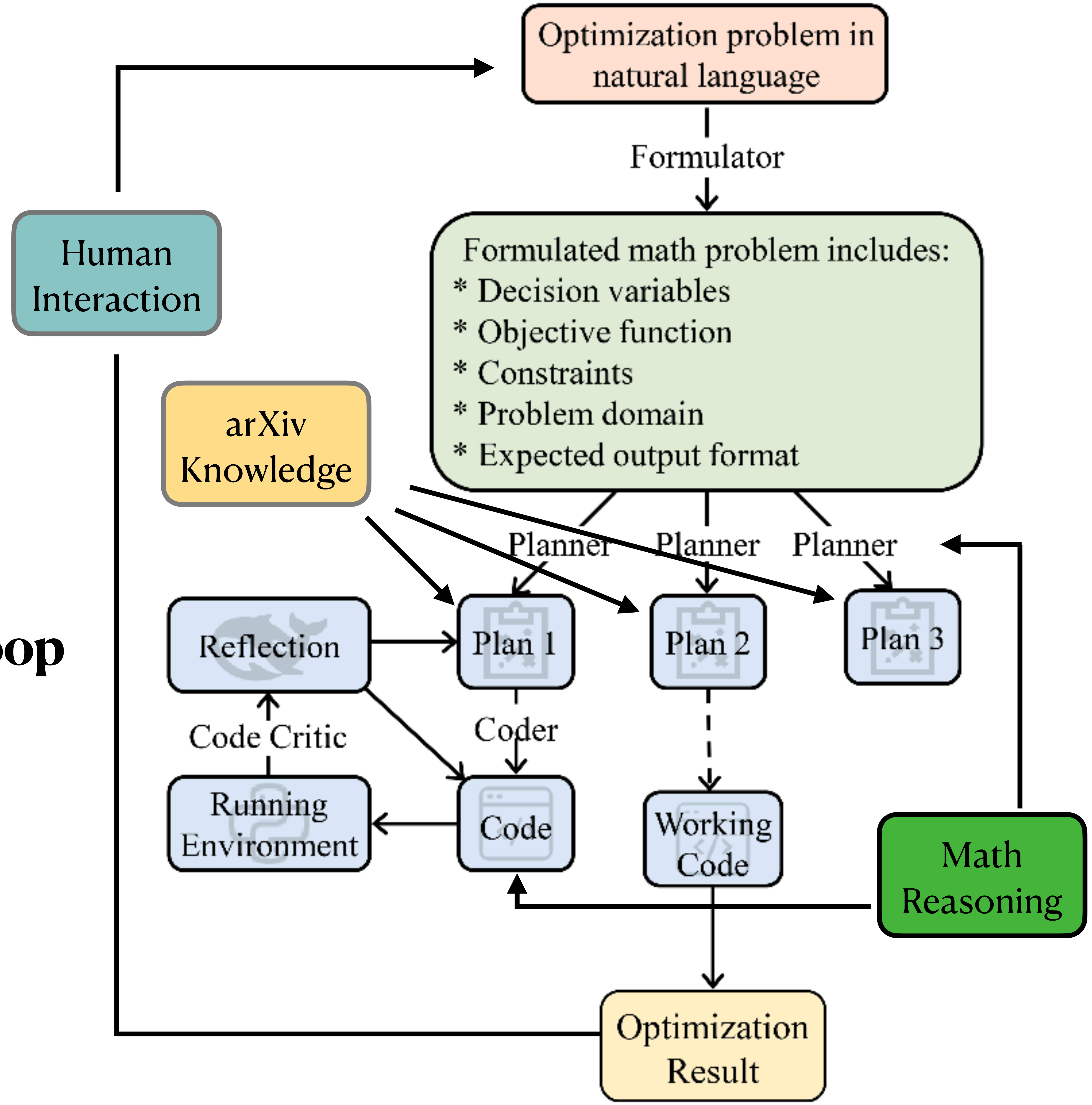
- Non-expert users for LP, NLP, MILP, MINLP
- Bayesian approach with arXiv knowledge
- **Reasoning with chain of thoughts**
  - Algorithm analysis
  - Coding analysis
- Optimization AI assistant with human in the loop



OptimAI Roadmap:

- Non-expert users for LP, NLP, MILP, MINLP
- Bayesian approach with arXiv knowledge
- Reasoning with chain of thoughts
  - Algorithm analysis
  - Coding analysis
- **Optimization AI assistant with human in the loop**





Take Home Messages

- Modeling and Computing **in description**
- Leverage the power of **descriptive structures** of challenging problems
- Leverage the power of **automatic big** search

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

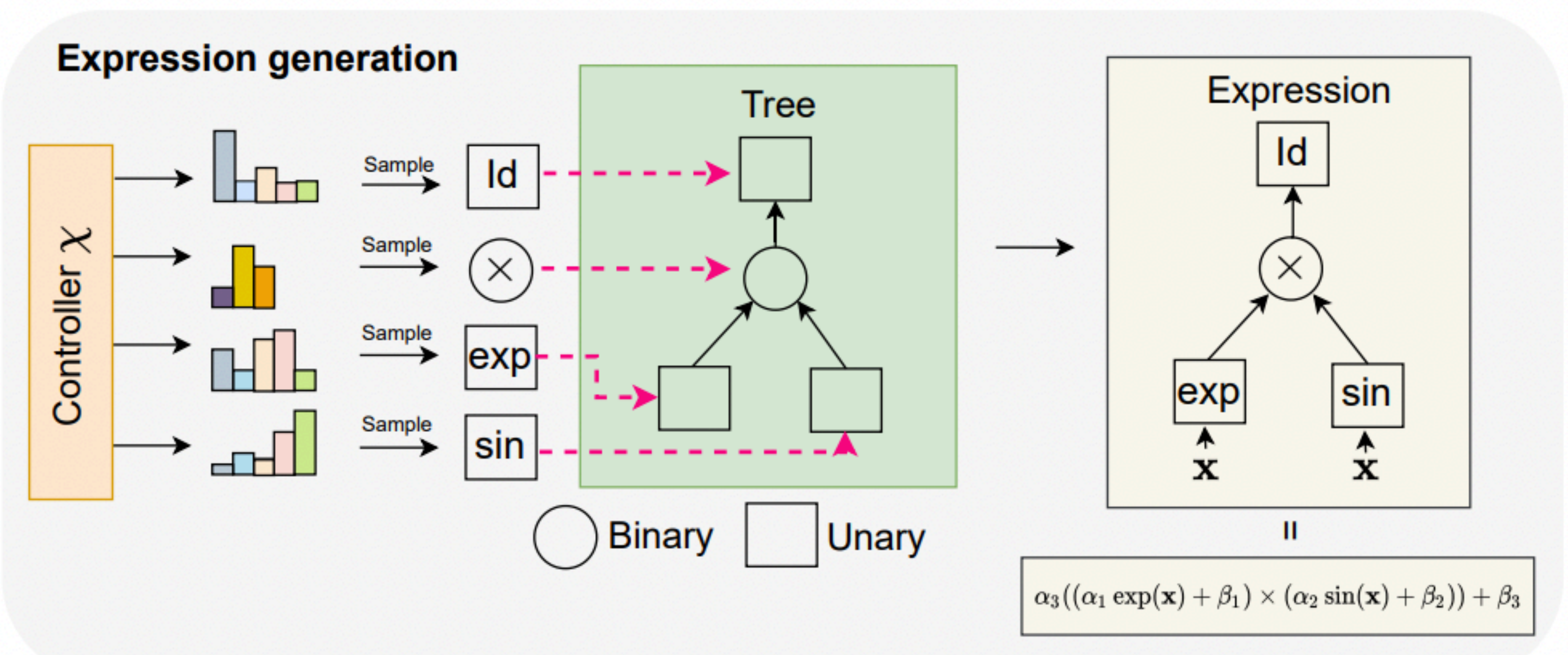
$$u^* = \arg\min_{u \in \mathbb{S}_k} \mathscr{L}(u) := \arg\min_{u \in \mathbb{S}_k} \|\mathscr{D}u - f\|_2^2 + \lambda \|\mathscr{B}u - g\|_2^2$$

- 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

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

and gradient descent in the space of probability distributions

- Finally, $u^* \sim (P_1^*, \ldots, P_k^*)$ with the optimal parameters α^* and β^*