Dynamic Bayesian Optimization Framework for Instruction Tuning in Partial Differential Equation Discovery

Anonymous ACL submission

Abstract

Large Language Models (LLMs) show promise for equation discovery, yet their outputs are highly sensitive to prompt phrasing—a phenomenon we term *instruction brittleness*. Static prompts cannot adapt to the evolving state of a multi-step generation process, causing models to plateau at suboptimal solutions. To address this, we propose NEUROSYM-BO, which reframes prompt engineering as a *sequential decision problem*. Our method maintains a discrete library of reasoning strategies and uses Bayesian Optimization to select the optimal instruction at each step based on numerical feedback. Experiments on PDE discovery benchmarks show that adaptive instruction selection significantly outperforms fixed prompts, achieving higher recovery rates with more parsimonious solutions.

1 Introduction

The automated discovery of physical laws from data is a central challenge in AI for Science (Wang et al., 2023; Raghu and Schmidt, 2020). While traditional Symbolic Regression (SR) methods like Genetic Programming (e.g., PySR) are effective, they struggle with combinatorial search spaces and lack semantic priors (Cranmer et al., 2020). Large Language Models (LLMs) offer a promising alternative by leveraging pre-trained knowledge of physics and code generation (Chen et al., 2021; Wei et al., 2022). However, applying LLMs to equation discovery faces a critical barrier: *instruction brittleness* (Sclar et al., 2024; Mizrahi et al., 2024)—the phenomenon where small changes in prompt phrasing lead to dramatically different outputs.

Standard frameworks such as LLM-SR (Shojaee et al., 2025) and LLM4ED (Du et al., 2024) typically employ fixed prompt templates (e.g., "Find the equation that fits this data"). This rigid approach mimics a "static researcher" unable to adapt their line of questioning. An LLM prompted with "Find the simplest equation" might over-regularize and miss important terms; the same model prompted with "Find the most accurate equation" might hallucinate spurious terms or fixate on memorized but irrelevant formulas. When faced with complex nonlinear dynamics, static prompts often lead models into local optima from which they cannot escape. This paper makes a key observation: **the optimal instruction depends on the current state of the generation process**. Early in the search, exploratory prompts ("propose novel functional forms") are beneficial; later, refinement prompts ("prune redundant terms") become essential. A fixed prompt cannot capture this state-dependency.

To overcome this, we introduce NEUROSYM-BO, a framework that treats the prompt instruction not as a fixed input, but as a *dynamic hyperparameter* to be optimized during the search. We construct a discrete space of reasoning strategies and employ Bayesian Optimization (BO) to navigate this space, enabling the system to actively switch strategies based on the error profile of current candidates. Our contributions are: **(1) Dynamic Instruction Tuning**: We formalize prompt engineering in scientific discovery as a discrete Bayesian Optimization problem, enabling adaptive control over the LLM's generation mode. **(2) Sample Efficiency**: Unlike Reinforcement Learning approaches to prompt optimization, our BO-based method is highly sample-efficient, making it feasible for computationally expensive scientific evaluations. **(3)Empirical Robustness**: We demonstrate that NEUROSYM-BO achieves higher recovery rates on benchmark PDEs, solving cases where static prompting fails.

2 Background

PDE Discovery We consider the task of discovering a governing partial differential equation (PDE) from observational data (Brunton et al., 2016; Rudy et al., 2017). Given a spatiotemporal dataset

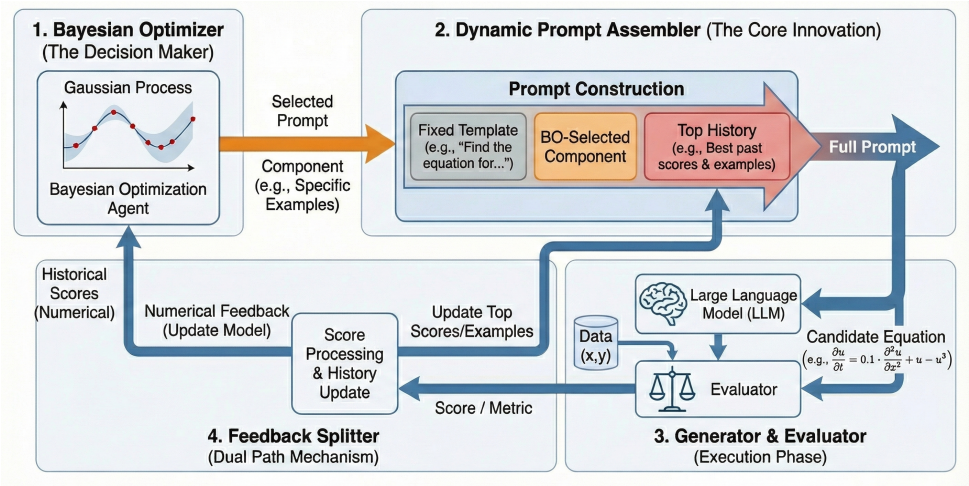


Figure 1: Overview of NEUROSYM-BO. A Bayesian Optimizer (1) selects instruction strategies that are assembled into prompts (2) with historical context. The LLM generates candidate equations for evaluation (3), and feedback updates both the history and the optimizer (4), forming a closed-loop system.

$\mathcal{D} = \{(x_i, t_i, u_i)\}_{i=1}^{N}$ representing measurements of a physical quantity u at spatial locations x and times t, our goal is to recover the symbolic form of the underlying PDE: $u_t = \mathcal{N}(u, u_x, u_{xx}, \ldots)$ where $\mathcal{N}$ is an unknown nonlinear operator composed of spatial derivatives and algebraic terms. The search space of possible equations is combinatorially large—even with a modest library of operators (addition, multiplication, differentiation), the number of candidate expressions grows exponentially with equation length. This makes exhaustive enumeration intractable and motivates the use of intelligent search strategies.

Bayesian Optimization(BO) Bayesian Optimization (BO) is a principled framework for optimizing expensive black-box functions (Snoek et al., 2012; Jones et al., 1998; Shahriari et al., 2016). The key idea is to maintain a probabilistic surrogate model (typically a Gaussian Process) (Rasmussen and Williams, 2006) that estimates both the expected value and uncertainty of the objective function at unobserved points. An acquisition function then uses this surrogate to decide which point to evaluate next, balancing the desire to exploit regions with high predicted performance against the need to explore uncertain regions that might contain better solutions.

3 Methodology

NEUROSYM-BO addresses the challenge of PDE discovery through a three-agent closed-loop architecture (Figure 1). The **Symbolic Proposer** (an LLM) generates candidate equation structures by leveraging its pre-trained knowledge of mathematical forms. The **Numerical Evaluator** fits coefficients to each candidate and computes a fitness score measuring how well the equation explains the observed data. The **Prompt Optimizer** (a Bayesian Optimization agent) analyzes feedback from the evaluator and selects the optimal instruction strategy for the next generation round. This closed-loop design allows the system to adaptively refine its search strategy based on accumulated evidence, rather than relying on a fixed generation policy.

Dynamic Prompt Construction The key innovation of our framework is treating the LLM's instruction as a dynamic, optimizable component. Instead of using a static prompt string throughout the discovery process, we dynamically assemble the prompt P_t at each iteration t:

$$P_t = I_{\text{task}} \oplus \mathcal{H}_{\text{top}} \oplus I_{\text{strategy}}^{(k_t)} \quad (1)$$

where $\oplus$ denotes string concatenation. Each component serves a distinct purpose: (1) **Static Context** (I_{task}): This fixed preamble defines the problem setting, including the state variables (e.g., u, x, t), the admissible operator library (e.g., $\{+, \times, \partial_x, \partial_{xx}, \sin, \exp\}$), and output formatting requirements. This ensures the LLM understands the symbolic constraints of the task. (2) **Dynamic Memory** ($\mathcal{H}_{\text{top}}$): To enable learning from past iterations, we include the top-N best-performing equations discovered so far, along with their fitness scores. This in-context history provides the LLM with implicit "gradient" information—by observing which structural patterns achieve high scores, the model can identify promising directions for further exploration. (3) **Optimizable Instruction**

($I^{(k_t)}_{\text{strategy}}$): This is the core optimizable component. Rather than using a generic directive like "find the best equation," we select a specific reasoning strategy from a pre-constructed *Strategy Bank* (Zhou et al., 2023; Sun et al., 2024) $\mathcal{B} = \{s_1, \ldots, s_K\}$. We generate $K = 100$ diverse instruction variants using a meta-LLM (GPT-4o), covering different aspects of the discovery process. These strategies span a spectrum from *exploration* ("Ignore previous bests. Propose a completely new mathematical structure that has not been tried.") to *parsimony* ("The current best equations are too complex. Identify and remove terms that contribute least to the fit."), and include *mutation* directives ("Keep the core structure of the best equation but replace the nonlinear interaction terms with alternatives.") as well as *refinement* instructions ("The structure looks correct. Focus on adjusting the functional form of individual terms.").

Bayesian Optimization for Strategy Selection Selecting the optimal strategy at each iteration is itself an optimization problem. However, evaluating any strategy is expensive: it requires generating candidates from the LLM, fitting their coefficients, and computing residuals against the data. This rules out gradient-based or exhaustive search methods. We model the mapping from strategy index $k \in \{1, \ldots, K\}$ to the resulting best equation fitness as a black-box function $f : \{1, \ldots, K\} \to \mathbb{R}$. A Gaussian Process surrogate $\mathcal{GP}$ is fitted to the history of (strategy, fitness) pairs observed so far, providing posterior estimates of the mean $\mu(k)$ and uncertainty $\sigma(k)$ for each strategy. We select the next strategy by maximizing the Expected Improvement (EI) acquisition function:

$$k_{t+1} = \arg\max_k \mathbb{E}_{y \sim \mathcal{GP}} \left[\max(y - y^*, 0)\right] \quad (2)$$

where y^* is the best fitness score observed so far. EI quantifies the expected gain from trying strategy k: it assigns high values to strategies that either have high predicted performance (exploitation) or high uncertainty (exploration). This principled balance allows the system to efficiently navigate the discrete strategy (Baptista and Poloczek, 2018) space without exhaustively trying all options.

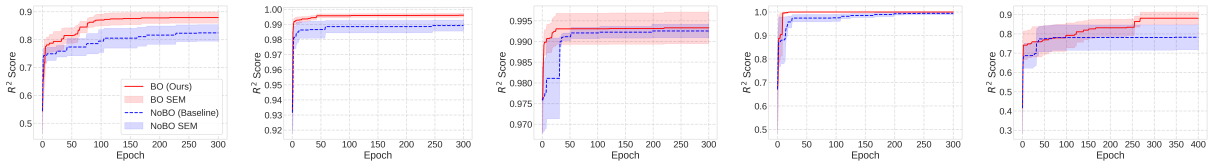
Numerical Evaluation with Parsimony Penalty Once the LLM generates candidate symbolic skeletons, we must evaluate their quality. Each candidate is parsed into a symbolic expression, and its free coefficients are optimized using sparse regression (specifically, STRidge) (Tibshirani, 1996) to minimize the residual against the observed data $\mathcal{D}$. To prevent overfitting through overly complex equations—a common failure mode in symbolic regression—we design a composite fitness function that balances accuracy against parsimony (Bartlett et al., 2024; Burlacu et al., 2019):

$$S(\hat{u}) = \frac{1 - \lambda \cdot \text{complexity}(\hat{u})}{1 + \text{NRMSE}(\hat{u}, \mathcal{D})} \quad (3)$$

where $\text{complexity}(\hat{u})$ counts the number of terms in the equation, λ is a penalty coefficient (set to 0.01 in our experiments), and $\text{NRMSE} = \sqrt{\text{MSE}/\text{Var}(u)}$ is the normalized root mean square error. This formulation ensures that among equations with similar accuracy, simpler ones receive higher scores. A complete algorithmic description with complexity bounds is given in Appendix A.

4 Related Work

Classical equation discovery follows two main approaches. Sparse regression methods like SINDy (Brunton et al., 2016; Messenger and Bortz, 2021; Fasel et al., 2022) and PDE-FIND (Rudy et al., 2017) select active terms from predefined libraries via sparsity-promoting optimization. Genetic programming methods—Eureqa (Schmidt and Lipson, 2009), PySR (Cranmer, 2023), Operon (Burlacu et al., 2020)—evolve expression trees, with physics-informed variants adding domain constraints (Zhang et al., 2024b). Neural approaches include AI Feynman (Udrescu and Tegmark, 2020; Udrescu et al., 2020; Kamienny et al., 2022; Mundhenk et al., 2021; Landajuela et al., 2022) exploiting physical symmetries, Deep Symbolic Regression (Petersen et al., 2021) using risk-seeking policy gradients, and transformer models like NeSymReS (Biggio et al., 2021) and SymFormer (Vastl et al., 2022). KBASS (Long et al., 2023) combines Bayesian spike-and-slab priors with kernel methods. SRBench (La Cava et al., 2021) and Feynman equations (Udrescu and Tegmark, 2020) provide standard benchmarks. Recent LLM-based methods leverage pre-trained scientific knowledge: LLM-SR (Shojaee et al., 2025; Boiko et al., 2023; Ma et al., 2024; Zhang et al., 2024a) uses evolutionary refinement on Python-represented equations, LLM4ED (Du et al., 2024) alternates self-improvement and evolutionary phases, FunSearch (Romera-Paredes et al., 2024) achieved novel mathematical discoveries, LaSR (Grayeli et al., 2024) builds reusable concept libraries, and ICSR (Merler et al., 2024) applies in-context learning. However,



(a) Burgers (b) Chafee (c) Divide (d) Fisher (e) Allen-Cahn

Figure 2: Optimization trajectories (test R^2) across five PDEs: red solid: NEUROSYM-BO; blue dashed: Fixed Prompt baseline. Shaded regions indicate ±SEM over 5 trials. Our method exhibits step-wise improvements from adaptive strategy switching, while the baseline plateaus early.

these methods use static prompts, unable to adapt instructions based on search progress. Prompt optimization methods include APE (Zhou et al., 2023) using bandit selection, OPRO (Yang et al., 2024) with LLMs as meta-optimizers, and evolutionary approaches like EvoPrompt (Guo et al., 2024) and PromptBreeder (Fernando et al., 2024). Bayesian prompt optimization has also been explored (Sabbatella et al., 2024). Yet existing methods seek a single optimal prompt, whereas we argue *different prompts are optimal at different stages*. Our work introduces closed-loop instruction optimization via Bayesian Optimization, dynamically selecting strategies based on numerical feedback.

5 Experiments

We evaluate NEUROSYM-BO on five benchmark PDEs: Burgers, Fisher, Chafee-Infante, Divide, and Allen-Cahn (details in Appendix B). Both methods use Llama-3.2-3B-Instruct [1] with identical in-context history (top-5 equations). The **Fixed Prompt** baseline employs a static instruction throughout, while **NEUROSYM-BO** dynamically selects from the 100-strategy library. We implement the prompt optimizer using BoTorch [2] with a standard GP surrogate and EI acquisition. All experiments run for 300 iterations across 5 trials; we report the mean R^2 on held-out test data. Table 1 shows NEUROSYM-BO consistently outperforms the fixed-prompt baseline, with improvements of 5-11% on challenging cases (Allen-Cahn, Burgers) and near-perfect recovery on Fisher ($R^2 = 0.9999$). Figure 2 visualizes optimization trajectories across all five PDEs. Several patterns emerge: (1) The fixed-prompt baseline (blue, dashed) plateaus early, typically within 50-100 iterations, struggling to escape local optima. (2) NEUROSYM-BO (red, solid) exhibits characteristic step-wise improvements, where sudden jumps correspond to the BO agent successfully switching strategies—for instance, transitioning from exploration-focused prompts to simplification directives when the current best equation becomes overly complex. (3) The performance gap widens over time, demonstrating that dynamic instruction selection provides compounding benefits as the search progresses. The shaded regions (±SEM) indicate that our method also achieves lower variance across trials, suggesting more robust convergence.

PDE	Method	Train R^2	Test R^2
Allen-Cahn	Fixed	0.7968	0.7824
	Ours	**0.9107**	**0.8914**
Burgers	Fixed	0.8102	0.8242
	Ours	**0.8699**	**0.8791**
Chafee	Fixed	0.9894	0.9886
	Ours	**0.9951**	**0.9947**
Divide	Fixed	0.9927	0.9922
	Ours	**0.9942**	**0.9941**
Fisher	Fixed	0.9952	0.9953
	Ours	**0.9999**	**0.9999**

Table 1: Performance comparison across five benchmark PDEs (5-trial average). NEUROSYM-BO (Ours) consistently outperforms the Fixed Prompt baseline on both training and test R^2.

6 Conclusion

In this work, we introduced NEUROSYM-BO, a closed-loop framework that leverages Bayesian Optimization to dynamically optimize instructions for LLM-based equation discovery. By treating prompt engineering as a search problem over a discrete strategy space, our method effectively overcomes the prompt sensitivity bottleneck inherent in static approaches. Experiments on benchmark PDEs demonstrate that our framework recovers correct symbolic structures with higher success rates while improving sample efficiency. Our findings suggest that combining LLM generative reasoning with numerical optimization feedback is a promising direction for automated scientific discovery.

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct

[2] https://botorch.org/

Limitations

While NEUROSYM-BO demonstrates promising results, it currently relies on the inherent mathematical capabilities of the backbone LLM (e.g., Llama-3). If the LLM lacks fundamental knowledge of specific mathematical operators, prompt optimization alone cannot solve the problem. Furthermore, our current evaluation focuses on 1D PDEs; scaling to higher-dimensional systems with chaotic behavior remains future work.

Ethics Statement

This work introduces a framework for automated scientific discovery. The primary application is to accelerate research in physics and engineering. We do not foresee any immediate negative societal impacts or potential for misuse. The energy consumption of our experiments is minimal compared to training large models, as we perform inference and lightweight BO updates.

References

Mark J Ablowitz and Anthony Zeppetella. 1979. Explicit solutions of Fisher's equation for a special wave speed. *Bulletin of Mathematical Biology*, 41(6):835–840.

Samuel M Allen and John W Cahn. 1979. A microscopic theory for antiphase boundary motion and its application to antiphase domain coarsening. *Acta Metallurgica*, 27(6):1085–1095.

Ricardo Baptista and Matthias Poloczek. 2018. Bayesian optimization of combinatorial structures. In *International Conference on Machine Learning*, pages 462–471.

Deaglan J Bartlett, Harry Desmond, and Pedro G Ferreira. 2024. Exhaustive symbolic regression and model selection by minimum description length. *IEEE Transactions on Evolutionary Computation*.

Harry Bateman. 1915. Some recent researches on the motion of fluids. *Monthly Weather Review*, 43(4):163–170.

Peter W Bates and Paul C Fife. 1990. Spectral comparison principles for the Cahn-Hilliard and phase-field equations, and time scales for coarsening. *Physica D: Nonlinear Phenomena*, 43(2-3):335–348.

Luca Biggio, Tommaso Bendinelli, and 1 others. 2021. Neural symbolic regression that scales. In *International Conference on Machine Learning*, pages 936–945.

Daniil A Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. 2023. Autonomous chemical research with large language models. *Nature*, 624:570–578.

John P Boyd. 2001. *Chebyshev and Fourier Spectral Methods*. Dover Publications.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937.

Johannes Martinus Burgers. 1948. A mathematical model illustrating the theory of turbulence. *Advances in Applied Mechanics*, 1:171–199.

Bogdan Burlacu, Gabriel Kronberger, Michael Affenzeller, and Michael Kommenda. 2019. Parsimony measures in multi-objective genetic programming for symbolic regression. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 338–339.

Bogdan Burlacu, Gabriel Kronberger, and Michael Kommenda. 2020. Operon C++: An efficient genetic programming framework for symbolic regression. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1562–1570.

John W Cahn and John E Hilliard. 1958. Free energy of a nonuniform system. I. Interfacial free energy. *The Journal of Chemical Physics*, 28(2):258–267.

Nathaniel Chafee and Ettore F Infante. 1974. A bifurcation problem for a nonlinear partial differential equation of parabolic type. *Applicable Analysis*, 4(1):17–37.

Mark Chen, Jerry Tworek, Heewoo Jun, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Julian D Cole. 1951. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of Applied Mathematics*, 9(3):225–236.

Miles Cranmer. 2023. Interpretable machine learning for science with PySR and SymbolicRegression.jl. *arXiv preprint arXiv:2305.01582*.

Miles Cranmer, Alvaro Sanchez-Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. 2020. Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems*, volume 33, pages 17429–17442.

John R Dormand and Peter J Prince. 1980. A family of embedded Runge-Kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26.

Mengge Du, Yuntian Chen, and 1 others. 2024. LLM4ED: Large language models for automatic equation discovery. *arXiv preprint arXiv:2405.07761*.

Qiang Du, Lili Ju, Xiao Li, and Zhonghua Qiao. 2019. Maximum principle preserving exponential time differencing schemes for the nonlocal Allen–Cahn equation. *SIAM Journal on Numerical Analysis*, 57(2):875–898.

Urban Fasel, J Nathan Kutz, Bingni W Brunton, and Steven L Brunton. 2022. Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A*, 478(2260):20210904.

Chrisantha Fernando, Dylan Banarse, and 1 others. 2024. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.

Ronald Aylmer Fisher. 1937. The wave of advance of advantageous genes. *Annals of Eugenics*, 7(4):355–369.

Arya Grayeli, Atharva Shai, Solomon Fisch, and 1 others. 2024. Symbolic regression with a learned concept library. In *Advances in Neural Information Processing Systems*.

Qingyan Guo, Rui Wang, and 1 others. 2024. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.

Daniel Henry. 1981. Geometric theory of semilinear parabolic equations. *Lecture Notes in Mathematics*, 840.

Eberhard Hopf. 1950. The partial differential equation $u_t + uu_x = \mu u_{xx}$. *Communications on Pure and Applied Mathematics*, 3(3):201–230.

Donald R Jones, Matthias Schonlau, and William J Welch. 1998. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492.

Pierre-Alexandre Kamienny, Stéphane d'Ascoli, Guillaume Lample, and Francois Charton. 2022. End-to-end symbolic regression with transformers. In *Advances in Neural Information Processing Systems*, volume 35, pages 10269–10281.

Andrei Kolmogorov, Ivan Petrovskii, and Nikolai Piskunov. 1937. A study of the equation of diffusion with increase in the quantity of matter, and its application to a biological problem. *Moscow University Bulletin of Mathematics*, 1:1–25.

William La Cava, Patryk Orzechowski, and 1 others. 2021. Contemporary symbolic regression methods and their relative performance. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Mikel Landajuela, Chak Shing Lee, Jiachen Yang, and 1 others. 2022. A unified framework for deep symbolic regression. In *Advances in Neural Information Processing Systems*, volume 35, pages 33985–33998.

Da Long and 1 others. 2023. Equation discovery with Bayesian spike-and-slab priors and efficient kernels. *arXiv preprint arXiv:2310.05387*.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, and 1 others. 2024. LLM and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. In *International Conference on Machine Learning*, pages 33940–33962.

Matteo Merler, Katsiaryna Haitsiukevich, and 1 others. 2024. In-context symbolic regression: Leveraging large language models for function discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*.

Daniel A Messenger and David M Bortz. 2021. Weak SINDy for partial differential equations. *Journal of Computational Physics*, 443:110525.

Moran Mizrahi, Guy Kaplan, Dan Malber, and 1 others. 2024. State of what art? a call for multi-prompt LLM evaluation. *arXiv preprint arXiv:2401.00595*.

T Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, and 1 others. 2021. Symbolic regression via neural-guided genetic programming population seeding. In *Advances in Neural Information Processing Systems*, volume 34, pages 24912–24923.

James D Murray. 2002. Mathematical biology i: An introduction. *Interdisciplinary Applied Mathematics*, 17.

Brenden K Petersen, Mikel Landajuela, and 1 others. 2021. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations*.

Joaquin Quiñonero-Candela and Carl Edward Rasmussen. 2005. A unifying view of sparse approximate Gaussian process regression. In *Journal of Machine Learning Research*, volume 6, pages 1939–1959.

Maithra Raghu and Eric Schmidt. 2020. A survey of deep learning for scientific discovery. *arXiv preprint arXiv:2003.11755*.

Carl Edward Rasmussen and Christopher KI Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Bernardino Romera-Paredes, Mohammadamin Barekatain, and 1 others. 2024. Mathematical discoveries from program search with large language models. *Nature*, 625(7995):468–475.

Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. 2017. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614.

Antonio Sabbatella, Andrea Ponti, and 1 others. 2024. A Bayesian approach for prompt optimization in pre-trained language models. *arXiv preprint arXiv:2312.00471*.

Michael Schmidt and Hod Lipson. 2009. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design. In *International Conference on Learning Representations*.

Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando De Freitas. 2016. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175.

Parshin Shojaee, Kazem Meidani, and 1 others. 2025. LLM-SR: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25.

Zhaoxuan Sun, Ziniu Zhuang, and 1 others. 2024. Prompt optimization with EASE? efficient ordering-aware automated selection of exemplars. In *Advances in Neural Information Processing Systems*.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B*, 58(1):267–288.

Lloyd N Trefethen. 2000. *Spectral Methods in MATLAB*. SIAM.

Silviu-Marian Udrescu, Andrew Tan, Jiaqi Feng, and 1 others. 2020. AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in Neural Information Processing Systems*, volume 33, pages 4860–4871.

Silviu-Marian Udrescu and Max Tegmark. 2020. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631.

Martin Vastl, Jonáᵴvs Kulhánek, and 1 others. 2022. SymFormer: End-to-end symbolic regression using transformer-based architecture. *arXiv preprint arXiv:2205.15764*.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, and 1 others. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272.

Hanchen Wang, Tianfan Fu, Yuanqi Du, and 1 others. 2023. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60.

Jason Wei, Xuezhi Wang, Dale Schuurmans, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837.

Gerald Beresford Whitham. 2011. *Linear and Nonlinear Waves*. John Wiley & Sons.

Chengrun Yang, Xuezhi Wang, and 1 others. 2024. Large language models as optimizers. *arXiv preprint arXiv:2309.03409*.

Di Zhang, Wei Liu, Qian Tan, and 1 others. 2024a. ChemLLM: A chemical large language model. *arXiv preprint arXiv:2402.06852*.

Jiachen Zhang, Chengyang Guo, and 1 others. 2024b. Physics-informed genetic programming for discovery of partial differential equations from scarce and noisy data. *Journal of Computational Physics*, 514:113262.

Yongchao Zhou, Andrei Ioan Muresanu, and 1 others. 2023. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.

A Algorithm Overview

Algorithm 1 NEUROSYM-BO: Dynamic Instruction Tuning for PDE Discovery

Require: Dataset $\mathcal{D} = \{(x_i, t_i, u_i)\}_{i=1}^{N}$, Strategy bank $\mathcal{B} = \{s_1, \ldots, s_K\}$, Max iterations T, Top-N history size
Ensure: Best discovered equation $\hat{u}^*$

1: Initialize GP surrogate $\mathcal{GP}$, history buffer $\mathcal{H} \leftarrow \emptyset$, observation set $\mathcal{O} \leftarrow \emptyset$
2: $y^* \leftarrow -\infty$ ▷ Best fitness so far
3: **for** $t = 1, \ldots, T$ **do**
4: *// Strategy Selection via Bayesian Optimization*
5: **if** $t \leq K_{\text{init}}$ **then** ▷ Initial exploration phase
6: $k_t \leftarrow \text{RANDOMSELECT}(\mathcal{B})$
7: **else**
8: Fit $\mathcal{GP}$ to observation set $\mathcal{O}$
9: $k_t \leftarrow \arg\max_k \text{EI}(k; \mathcal{GP}, y^*)$ ▷ Eq. 2
10: **end if**
11: *// Dynamic Prompt Construction*
12: $\mathcal{H}_{\text{top}} \leftarrow \text{TOPN}(\mathcal{H}, N)$ ▷ Best N equations with scores
13: $P_t \leftarrow I_{\text{task}} \oplus \mathcal{H}_{\text{top}} \oplus I_{\text{strategy}}^{(k_t)}$ ▷ Eq. 1
14: *// LLM Generation & Numerical Evaluation*
15: $\{\hat{u}_1, \ldots, \hat{u}_M\} \leftarrow \text{LLM}(P_t)$ ▷ Generate M candidates
16: **for** each candidate $\hat{u}_j$ **do**
17: $\hat{u}_j \leftarrow \text{STRIDGE}(\hat{u}_j, \mathcal{D})$ ▷ Fit coefficients
18: $S_j \leftarrow \frac{1-\lambda \cdot \text{complexity}(\hat{u}_j)}{1+\text{NRMSE}(\hat{u}_j, \mathcal{D})}$ ▷ Eq. 3
19: **end for**
20: *// Feedback & Model Update*
21: $S_t \leftarrow \max_j S_j$; $\hat{u}_t \leftarrow \arg\max_j S_j$
22: $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\hat{u}_t, S_t)\}$
23: $\mathcal{O} \leftarrow \mathcal{O} \cup \{(k_t, S_t)\}$ ▷ Update BO observations
24: **if** $S_t > y^*$ **then**
25: $y^* \leftarrow S_t$; $\hat{u}^* \leftarrow \hat{u}_t$
26: **end if**
27: **end for**
28: **return** $\hat{u}^*$

Algorithm 1 summarizes the NEUROSYM-BO procedure. The algorithm operates in three phases per iteration. First, the **strategy selection** phase (lines 4–10) uses Bayesian Optimization to choose the next instruction strategy k_t from the bank $\mathcal{B}$. During an initial exploration phase ($t \leq K_{\text{init}}$, set to 10 in our experiments), strategies are sampled randomly to build an initial surrogate model. Subsequently, the GP surrogate is fitted to accumulated (strategy, fitness) observations, and the strategy maximizing Expected Improvement is selected.

Second, the **prompt construction** phase (lines 12–13) assembles the dynamic prompt P_t by concatenating the static task description I_{task}, the top-N best-performing equations from history $\mathcal{H}_{\text{top}}$, and the BO-selected instruction $I_{\text{strategy}}^{(k_t)}$. This provides the LLM with both problem context and implicit feedback about which structural patterns have succeeded.

Third, the **generation and evaluation** phase (lines 15–20) queries the LLM to produce M candidate equations (we use $M = 5$), fits their coefficients via sparse regression, and computes fitness scores balancing accuracy against parsimony. The best candidate updates both the history buffer (for in-context learning) and the BO observation set (for surrogate refinement). This closed-loop design enables the system to progressively refine both *what* equations to propose and *how* to instruct the LLM.

B Dataset and Equation Details

To ensure rigorous and reproducible evaluation, we utilize a combination of established benchmark datasets and self-generated simulations that span a diverse range of physical phenomena. For the Burgers, Fisher, Chafee-Infante, and PDE_divide equations, we employ standard datasets from the **LLM4ED** benchmark suite (Du et al., 2024), which has been widely adopted for evaluating symbolic equation discovery methods. Additionally, we generate a synthetic dataset for the Allen-Cahn equation to test our framework's capability on phase-separation dynamics characterized by sharp interfaces and bistable nonlinearities. Together, these five PDEs represent a comprehensive testbed covering fluid dynamics, population biology, pattern formation, and materials science.

Time Complexity. Let T be the total number of iterations, K the strategy bank size, M the number of LLM-generated candidates per iteration, N the data points in $\mathcal{D}$, and d the number of candidate terms in the operator library.

- *Strategy Selection (lines 4–10):* Fitting the GP surrogate to t observations requires $\mathcal{O}(t^3)$ for exact inference due to kernel matrix inversion. Computing EI across K strategies costs $\mathcal{O}(K)$. Per iteration at step t: $\mathcal{O}(t^3 + K)$.

- *Prompt Construction (lines 12–13):* Selecting top-N equations from history $\mathcal{H}$ costs $\mathcal{O}(|\mathcal{H}|)$ with a heap, or $\mathcal{O}(1)$ if maintained incrementally. String concatenation is $\mathcal{O}(L_P)$ where L_P is the prompt length.

- *LLM Generation (line 15):* Each forward pass through the LLM costs $\mathcal{O}(L_P \cdot d_{\text{model}})$ for context encoding, where d_{model} is the model dimension. Generating M candidates: $\mathcal{O}(M \cdot L_P \cdot d_{\text{model}})$.

- *Coefficient Fitting (line 17):* STRidge sparse regression for each candidate costs $\mathcal{O}(N \cdot d^2)$ per iteration of the thresholding loop. With M candidates: $\mathcal{O}(M \cdot N \cdot d^2)$.

- *Fitness Evaluation (line 18):* Computing NRMSE requires $\mathcal{O}(N)$ operations; complexity counting is $\mathcal{O}(d)$. Total: $\mathcal{O}(M \cdot N)$.

The total time complexity over T iterations is $\mathcal{O}\left(\sum_{t=1}^{T} t^3 + T \cdot \left(K + M \cdot (C_{\text{LLM}} + N \cdot d^2)\right)\right) = \mathcal{O}(T^4 + T \cdot M \cdot C_{\text{LLM}})$ where C_{LLM} denotes the LLM inference cost. In practice, the GP cubic term can be mitigated using sparse GP approximations (Quiñonero-Candela and Rasmussen, 2005), reducing it to $\mathcal{O}(T \cdot m^2)$ with $m \ll T$ inducing points. The dominant cost is typically LLM inference.

Space Complexity. The algorithm maintains: (1) the history buffer $\mathcal{H}$ storing $\mathcal{O}(T)$ equations, (2) the GP observation set $\mathcal{O}$ of size $\mathcal{O}(T)$, and (3) the kernel matrix of size $\mathcal{O}(T^2)$. Total space complexity is $\mathcal{O}(T^2 + T \cdot L_{\text{eq}})$, where L_{eq} is the average equation length.

Practical Considerations. In our experiments with $T = 300$, $K = 100$, $M = 5$, and Llama-3.2-3B, each iteration completes in approximately 15–30 seconds on a single A100 GPU, with LLM inference accounting for ∼80% of the runtime. The GP overhead remains negligible (<1%) due to the moderate number of iterations.

B.1 Benchmarks from LLM4ED

The following equations are adopted from the LLM4ED repository (Du et al., 2024). We provide their canonical forms, physical interpretations, and simulation configurations below.

Burgers' Equation. Burgers' equation is a fundamental partial differential equation in fluid mechanics that serves as a simplified model for shock wave formation, turbulence, and nonlinear acoustics (Burgers, 1948; Whitham, 2011). First introduced by Bateman (Bateman, 1915) and later extensively studied by Burgers (Burgers, 1948), it combines nonlinear convection with diffusive dissipation, making it an ideal testbed for symbolic discovery methods. The viscous form is given by:

$$u_t + uu_x = 0.1u_{xx} \tag{4}$$

where the left-hand side represents nonlinear advection (the term uu_x causes wave steepening) and the right-hand side represents viscous diffusion with coefficient $\nu = 0.1$. The interplay between these terms leads to the formation of shock-like structures that eventually smooth out due to diffusion. The celebrated Cole-Hopf transformation (Cole, 1951; Hopf, 1950) provides an analytical framework for understanding these solutions.

Setup: The simulation is performed on a spatial domain $x \in [-8, 8]$ over a time interval $t \in [0, 10]$. The data is discretized on a uniform grid of size 256×201 (spatial $\times$ temporal points). The initial condition consists of a smooth profile that evolves into a traveling shock wave, providing rich dynamics for equation discovery.

Fisher's Equation (Fisher-KPP). Fisher's equation, also known as the Fisher-Kolmogorov-Petrovsky-Piskunov (Fisher-KPP) equation, is a classical reaction-diffusion model originally proposed by Fisher (Fisher, 1937) and independently by Kolmogorov, Petrovskii, and Piskunov (Kolmogorov et al., 1937) to describe the spatial spread of advantageous genes in a population. It has since found applications in ecology, epidemiology, and combustion theory (Murray, 2002). The equation takes the form:

$$u_t = u_{xx} + u(1 - u) \tag{5}$$

where u_{xx} represents spatial diffusion and $u(1-u)$ is a logistic growth term that drives the population toward carrying capacity. This equation admits traveling wave solutions (Ablowitz and Zeppetella, 1979) that propagate at a minimum speed determined by the linearization at the unstable equilibrium $u = 0$.

Setup: The simulation domain is $x \in [-1, 1]$ with time interval $t \in [0, 1]$. The data is discretized on a 200×100 grid. The initial condition is chosen to exhibit front propagation behavior, testing the method's ability to recover both diffusive and reactive terms simultaneously.

Chafee-Infante Equation. The Chafee-Infante equation is a reaction-diffusion PDE that arises in the study of phase transitions and pattern formation (Chafee and Infante, 1974). It is closely related to the Allen-Cahn equation and exhibits bistable dynamics with two stable equilibria at $u = \pm 1$. The geometric theory of such semilinear parabolic equations has been extensively developed (Henry, 1981). The equation is given by:

$$u_t - u_{xx} = u - u^3 \tag{6}$$

The cubic nonlinearity $u - u^3$ creates a double-well potential structure, leading to the formation of domain walls (interfaces) separating regions of different phases. This equation has been extensively studied in the context of chaotic attractors and infinite-dimensional dynamical systems.

Setup: The computational domain is $x \in [0, 3]$ with time interval $t \in [0, 0.5]$. The data is discretized on a 301×200 grid. The relatively short time interval captures the initial transient dynamics and interface formation, providing a challenging test case for symbolic recovery of the cubic reaction term.

PDE_divide (Synthetic Division Test). This synthetic benchmark is specifically designed to evaluate the capability of symbolic discovery methods to recover rational terms involving division operators (e.g., $1/x$ or u/x). Such terms pose significant challenges for traditional genetic programming approaches, which often struggle with protected division operations and singularity handling. The equation is:

$$u_t = 0.25 u_{xx} - \frac{u_x}{x} \tag{7}$$

This PDE can be interpreted as a diffusion equation with a spatially-varying advection term that becomes singular at $x = 0$. The coefficient $-1/x$ in front of u_x represents a radially-dependent drift in cylindrical or spherical coordinates.

Setup: The domain is $x \in [1, 2]$ with time interval $t \in [0, 1]$. Importantly, the spatial domain is chosen to exclude $x = 0$, thereby avoiding the singularity while still requiring the discovery method to identify the $1/x$ dependence. The data is discretized on a 100×251 grid. This benchmark specifically tests whether NEUROSYM-BO can discover non-polynomial functional forms that are difficult for standard symbolic regression methods.

B.2 Allen-Cahn Equation

To further evaluate our framework on physically meaningful problems beyond the LLM4ED suite, we generate a synthetic dataset for the Allen-Cahn equation. This equation is a fundamental model in materials science and mathematical physics, describing phase separation phenomena in binary alloys, order-disorder transitions, and interface motion (Allen and Cahn, 1979). The governing equation is:

$$u_t = 0.1 u_{xx} + 5.0(u - u^3) \tag{8}$$

Here, the diffusion coefficient $D = 0.1$ controls the interface width, while the reaction rate $R = 5.0$ determines the strength of the bistable nonlinearity. The term $(u - u^3)$ derives from the derivative of a double-well free energy potential $F(u) = \frac{1}{4}(1 - u^2)^2$, originally introduced by Cahn and Hilliard (Cahn and Hilliard, 1958) in their seminal work on phase separation, with stable equilibria at $u = \pm 1$ representing two distinct phases.

Physical Significance. The Allen-Cahn equation exhibits rich dynamics including phase coarsening (where smaller domains shrink and larger ones grow), interface annihilation, and curvature-driven motion (Bates and Fife, 1990). The sharp interfaces between phases make this equation particularly challenging for numerical methods (Du et al., 2019) and, consequently, for data-driven discovery approaches that must accurately capture both the smooth bulk dynamics and the steep gradients at interfaces.

Numerical Generation. We generate ground truth data using high-accuracy numerical methods to ensure reliable training and evaluation. Specifically, we employ a pseudo-spectral method (Trefethen, 2000; Boyd, 2001) for spatial discretization, which provides exponential convergence for smooth solutions and accurately resolves the steep gradients at phase boundaries. Time integration is performed using an adaptive Runge-Kutta scheme (RK45) (Dormand and Prince, 1980) that automatically adjusts step sizes to maintain accuracy.

- **Parameters:** The diffusion coefficient is set to $D = 0.1$ and the reaction rate to $R = 5.0$. These values are chosen to produce well-separated timescales between diffusion and reaction, resulting in sharp but resolvable interfaces.

- **Domain & Grid:** The simulation is performed on a one-dimensional spatial domain $x \in [-10, 10]$ (total length $L = 20$) over a time interval $t \in [0, 10]$. We discretize the system on a uniform grid of size $N_x \times N_t = 256 \times 201$, providing sufficient resolution to capture interface dynamics while maintaining computational efficiency.

- **Initial Condition:** To generate complex, physically realistic dynamics with multiple interacting interfaces, we initialize the system with a composite trigonometric function perturbed by random noise:

$$u(x, 0) = \sin\left(\frac{2\pi x}{L}\right) + 0.5 \cos\left(\frac{4\pi x}{L}\right) + \epsilon \tag{9}$$

  where $L = 20$ is the domain length and $\epsilon \sim \mathcal{U}(0, 0.2)$ represents uniformly distributed initialization noise. This initial condition creates multiple zero-crossings that evolve into sharp interfaces, testing the discovery method's ability to handle multi-scale dynamics.

- **Boundary Conditions:** We impose periodic boundary conditions, which are naturally handled by the spectral method and ensure that no artificial boundary effects contaminate the interior dynamics.

- **Solver Details:** Spatial derivatives (u_{xx}) are computed in the frequency domain using the Fast Fourier Transform (FFT), which provides spectral accuracy for periodic problems (Trefethen, 2000). The semi-discrete system of ODEs is integrated using `scipy.integrate.solve_ivp`[3] (Virtanen et al., 2020) with the RK45 method. Both relative and absolute tolerances are set to 10^{-8} to ensure high-fidelity data generation. The solver adaptively refines the time step during periods of rapid interface motion.

- **Data Extraction:** After simulation, we uniformly sample the solution at 201 time points and store the full spatial field at each time, resulting in a data tensor of shape $(256, 201)$. We further split this data into training (80%) and testing (20%) sets along the temporal axis for evaluation.

[3] https://docs.scipy.org/doc/scipy/reference/integrate.html

Rationale for Inclusion. The Allen-Cahn equation complements the LLM4ED benchmarks by introducing several additional challenges: (1) stronger nonlinearity with the coefficient $R = 5.0$ compared to the Chafee-Infante equation, (2) longer time evolution allowing observation of phase coarsening dynamics, and (3) a different balance between diffusion and reaction terms. Successfully discovering this equation demonstrates that NEUROSYM-BO generalizes beyond the specific parameter regimes present in existing benchmarks.