Journal of Artificial Intelligence Research
E-ISSN (3109-0508)
https://jouair.com/index.php/Joair/index

Physics-Constrained Symbolic Regression via Reinforcement Learning: A Rigorous Framework for Discovering Closed-Form Solutions to Differential Equations

Esmatullah Abed[1], Mallang Ahmadi[2]
Universitas Bamyan, Afghanistan[1]
Universitas Panjshir, Afghanistan[2]
Corresponding Author: esmatullah.abed1368@gmail.com[1], kamal2@gmail.com[2]

Doi: https://doi.org/10.59261/jouair.v2i1.29

Accepted: 10-04-2026 Approved: 20-05-2026 Published: 23-06-2026

Abstract

Discovering exact, interpretable, closed-form solutions to differential equations (DEs) remains one of the most intellectually demanding challenges at the intersection of applied mathematics and computational intelligence. Classical analytical techniques, while rigorous, are largely restricted to well-structured linear systems and fail to generalise to the nonlinear, high-dimensional configurations that dominate contemporary science and engineering. The present study introduces and theoretically and empirically validates a physics-constrained symbolic regression framework, designated PCSRL (Physics-Constrained Symbolic Regression via Reinforcement Learning), that unifies policy-gradient reinforcement learning with an exact physical constraint evaluator to recover closed-form symbolic solutions for a broad class of ordinary and partial differential equations. Rigorous evaluation across six canonical benchmark problems including linear and nonlinear Poisson, heat, and wave equations in both two- and three-dimensional spatial domains demonstrates that PCSRL achieves complete symbolic recovery (recovery rate = 100%) and physical-constraint residuals multiple orders of magnitude below those of competing methods, including genetic programming-based symbolic regression, Kolmogorov–Arnold Networks, and PINN-assisted symbolic regression pipelines. These results establish a principled, reproducible methodology for the machine discovery of physically exact symbolic solutions, with direct implications for mathematical physics, computational fluid dynamics, and scientific machine learning. Practically, the framework enables engineers and scientists to obtain transparent, analytically tractable models for complex physical systems, facilitating rapid design optimisation, stability analysis, and knowledge discovery in domains where black-box numerical approximations are insufficient.

Keywords: symbolic regression; reinforcement learning; differential equations; closed-form solutions; physics-constrained learning

INTRODUCTION

Differential equations constitute the mathematical bedrock of the physical, biological, and engineering sciences. From the Navier–Stokes equations that govern viscous fluid dynamics to the reaction–diffusion systems that model morphogenetic patterning, the capacity to obtain exact solutions has profound epistemic and practical consequences (Evans, 2022; Blechschmidt & Ernst, 2021). An exact, closed-form expression is not merely computationally convenient; it is analytically transparent in ways that numerical approximations fundamentally are not.

It permits rigorous stability analysis, infinite extrapolation beyond the computational domain, identification of singularity structures, and unambiguous communication of physical law. For these reasons, the discovery of closed-form solutions even when they are available only for simplified or idealised configurations has remained a core objective of applied mathematics since the formulation of calculus (Lagaris et al., 1998).

The economic and scientific impact of differential equation solving is substantial and quantifiable. According to recent analyses in scientific computing, numerical simulation of partial differential equations (PDEs) consumes approximately 70–80% of high-performance computing cycles worldwide, with annual expenditures exceeding $20 billion across government laboratories, industrial R&D, and academic research facilities (Karniadakis et al., 2021). Despite this massive investment, the overwhelming majority of simulations produce black-box numerical outputs that cannot be directly analysed with algebraic or analytical tools.

The inability to obtain closed-form solutions imposes severe constraints on scientific understanding: engineers cannot perform symbolic sensitivity analysis, physicists cannot identify conservation laws from numerical tables, and mathematicians cannot verify structural properties of solutions that are only approximately known. This gap between numerical capability and analytical insight represents one of the most persistent bottlenecks in computational science.

The landscape of computational approaches to differential equations has undergone a dramatic transformation in the past decade, driven primarily by the emergence of deep learning. Physics-Informed Neural Networks (PINNs), introduced by Raissi et al. (2019) and subsequently extended in numerous directions (Cuomo et al., 2022; Karniadakis et al., 2021), embed physical constraints directly into neural network training via the equation residual, enabling unsupervised approximation of PDE solutions without labelled solution data.

Neural operator architectures including DeepONet (Lu et al., 2021), the Fourier Neural Operator (Li et al., 2023), and their variants go further by learning solution maps between function spaces, facilitating rapid evaluation of new problem instances once trained. Despite impressive empirical results, all of these methods share a fundamental limitation: they produce parametric approximations that lack symbolic form and hence are not analytically interpretable.

The field of symbolic regression offers a complementary pathway that explicitly targets interpretable mathematical expressions. Classical genetic programming (GP) searches expression trees via evolutionary operators and has been applied to DE solving with varying degrees of success (Oh et al., 2023; Cao et al., 2023). More recently, reinforcement learning (RL) has emerged as a powerful alternative search engine for symbolic regression, capable of learning structured policies over combinatorial expression spaces (Petersen et al., 2020; Landajuela et al., 2021).

However, the application of RL-based symbolic regression specifically to the recovery of physics-compliant, closed-form DE solutions remains nascent, and several critical gaps persist: (i) computationally efficient handling of constant optimisation under physical constraints; (ii) scalable search strategies for high-dimensional PDEs; and (iii) rigorous benchmark protocols that isolate symbolic recovery quality from mere numerical approximation quality.

The present paper addresses these gaps through three principal contributions. First, we present PCSRL, a complete, self-contained methodological framework for physics-constrained symbolic regression of DE solutions, providing derivations and justifications that complement and extend prior work in this area. Second, we introduce a theoretically motivated risk-seeking constant optimisation protocol, demonstrating formally that applying hard boundaries and initial conditions as a pre-filter before full residual optimisation yields substantially faster convergence without sacrificing accuracy. Third, we develop a dimension-recursive decomposition algorithm that reduces the search complexity of high-dimensional PDEs from exponential to manageable polynomial scaling, enabling consistent symbolic recovery where all prior methods fail.

The successful development of PCSRL carries significant benefits for both scientific research and practical engineering applications. From a theoretical perspective, the framework validates the hypothesis that risk-seeking policy gradients can efficiently navigate combinatorial symbolic spaces under physical constraints, extending the theoretical foundations established by Petersen et al. (2020) from data-driven regression to physics-constrained

differential equation solving. This advancement contributes to the broader scientific machine learning community by establishing a principled methodology for integrating physical knowledge with automated symbolic discovery.

From a practical standpoint, the ability to automatically discover closed-form solutions to PDEs has immediate implications for engineering design and analysis. In computational fluid dynamics, closed-form solutions enable rapid parameter sweeps and optimisation loops that would be computationally prohibitive with numerical solvers. In mathematical physics, exact symbolic expressions facilitate the identification of conserved quantities, symmetry properties, and asymptotic behaviours that are obscured by numerical approximations.

In scientific machine learning, the methodology provides a transparent alternative to black-box neural networks, supporting the growing demand for interpretable and explainable AI in safety-critical applications. Furthermore, the dimension-recursive decomposition strategy offers a generalisable approach to mitigating the curse of dimensionality in symbolic search, with potential applications beyond differential equations to fields such as automated theorem proving, materials science, and systems biology.

The remainder of the article is organised as follows. Section 2 reviews relevant literature across the four domains most pertinent to this work: numerical DE solvers, neural operator learning, genetic programming for DEs, and RL-based symbolic regression. Section 3 presents the mathematical foundations and formal problem statement. Section 4 describes the PCSRL methodology in full detail, including research type, design, instruments, data collection, and analysis techniques. Section 5 reports experimental settings and results. Section 6 presents an ablation study and complexity analysis. Section 7 discusses broader implications and limitations. Section 8 concludes.

The publication of Physics-Informed Neural Networks by Raissi et al. (2019) initiated a paradigm shift in the computational solution of PDEs. By incorporating the differential equation residual into the loss function, PINNs enable the training of fully connected networks to approximate solutions without requiring pre-computed labelled training data. Subsequent work has identified several persistent challenges: gradient pathologies in stiff problems (Wang et al., 2021), difficulty satisfying boundary conditions with soft constraints (Lu et al., 2021), and poor convergence for high-frequency solution components (Rao et al., 2023). Karniadakis et al. (2021) provided a comprehensive review and outlook for the field, articulating the need for hybrid methods that combine the expressive power of neural networks with stronger physical inductive biases.

Neural operator learning takes a different perspective, treating the PDE solution process as an operator mapping between function spaces (Lu et al., 2021a). The Fourier Neural Operator (FNO) by Li et al. (2023) uses spectral convolutions to capture global solution structure efficiently, demonstrating strong performance on turbulence and climate modelling benchmarks. Transformer-based operators (Li et al., 2023) have further extended this paradigm to irregular geometries and multi-scale problems.

A fundamental limitation shared by all these approaches is the requirement for large labelled datasets, which in scientific contexts typically entail expensive numerical simulation or experimental measurement. More critically for the present work, neither PINNs nor neural operators produce symbolic expressions; they yield black-box approximations that cannot be meaningfully analysed with algebraic or analytical tools (Blechschmidt & Ernst, 2021).

Genetic programming (GP), introduced by Koza (1994) as a branch of evolutionary computation, naturally accommodates the symbolic expression structure required for closed-form solution discovery. The fitness function in DE-solving GP is the physical constraint residual rather than a data-fitting criterion. Tsoulos and Lagaris (2006) provided early demonstrations of this approach, while Oh et al. (2023) significantly advanced the state of the art through the Bingo framework, which exploits automatic differentiation to evaluate DE residuals with high accuracy during GP evolution.

Cao et al. (2023, 2024) introduced pruning techniques and transfer learning to improve scalability, particularly for higher-dimensional PDEs. Despite these advances, GP-based methods suffer from two structural problems that are difficult to resolve within the evolutionary paradigm: (i) the expression trees they produce tend to grow in structural complexity over generations, often yielding solutions that are mathematically equivalent to simpler forms but syntactically dense; and (ii) the population-based search mechanism is inherently stochastic in a way that makes convergence guarantees elusive in high-dimensional spaces (Randall et al., 2022; Jiang & Xue, 2023).

Population-based metaheuristics, including ant colony programming (Kamali et al., 2015) and artificial bee colony programming (Boudouaoui et al., 2020), draw inspiration from collective biological foraging behaviours to guide symbolic search. These methods introduce diversity through population dynamics but inherit the same scalability limitations as GP when confronted with high-dimensional problems. The core difficulty exponential growth of the symbolic search space with both expression depth and the number of variables

remains unresolved by evolutionary population dynamics alone, necessitating fundamentally different algorithmic approaches.

The application of RL to symbolic regression was substantively advanced by Petersen et al. (2020), whose Deep Symbolic Regression (DSR) framework trains an RNN to generate symbolic expression trees using risk-seeking policy gradients. By focusing the learning signal on the top-performing episodes rather than the average, DSR efficiently explores the combinatorial space of mathematical expressions. Mundhenk et al. (2021) extended this with enhanced exploration mechanisms, while Landajuela et al. (2021) demonstrated that hierarchical entropy regularisation substantially improves diversity and prevents premature convergence. Sun et al. (2023) further advanced the field through Monte Carlo Tree Search (MCTS) integration, enabling explicit tree-structured planning over expression spaces. Li et al. (2024) introduced neural-guided symbolic networks that constrain the generation process based on learned mathematical structure.

In the DE-solving context specifically, Liang and Yang (2022) proposed the Finite Expression Method (FEX), which formulates PDE solving as a combinatorial optimisation over binary expression trees with continuous parameters. While FEX demonstrated feasibility for high-dimensional PDEs, its continuous parameterisation of discrete symbolic structure blurs the boundary between neural and symbolic representations, limiting the interpretability of recovered solutions.

Majumdar et al. (2023) combined PINN-generated numerical solutions with DSR, but the accumulated approximation errors from the numerical phase propagate into the symbolic regression phase, compromising physical accuracy. Kolmogorov–Arnold Networks (KANs), proposed by Liu et al. (2025), enable automatic symbolic regression through network symbolisation, but the conversion of learned network activations into closed-form expressions introduces significant accuracy degradation, particularly for multi-variable functions. The present work addresses all three of these limitations simultaneously within a unified, physics-first framework.

Supervised learning approaches to symbolic regression, which train large models on synthetic expression datasets and then decode expressions for new problems in a single forward pass, have achieved remarkable efficiency gains (Biggio et al., 2021; Kamienny et al., 2022).

The Transformer-based Symbolic Regression (TPSR) model of Shojaee et al. (2024) combines supervised pre-training with MCTS-based refinement, achieving state-of-the-art performance on standard symbolic regression benchmarks. However, the effectiveness of supervised methods depends critically on the alignment between the training distribution and the target

problem: when physical systems generate solution structures not well-represented in synthetic training data, performance degrades substantially (Li et al., 2023a). This distribution shift problem is particularly acute in the DE-solving context, where solution structures are determined by the differential operator rather than by any natural data distribution.

METHOD

Research Type and Design

This study employs a computational research methodology, specifically an algorithmic design and empirical validation approach. The research type is classified as computational experimental research, wherein novel algorithms are designed, implemented, and systematically evaluated against established benchmarks and competing methods under controlled, reproducible conditions. The research design follows a structured four-phase protocol: (i) framework design and theoretical justification; (ii) algorithmic implementation and software engineering; (iii) empirical evaluation against benchmark problems and baseline methods; and (iv) ablation analysis to isolate the contribution of individual components. This design ensures that each methodological innovation is independently validated and that the overall framework performance is assessed against state-of-the-art alternatives.

Instruments

The computational experiments were conducted on a high-performance workstation equipped with an Intel Core i9-13900K processor (24 cores, 3.0 GHz base clock, 5.8 GHz turbo), 64 GB DDR5 RAM (5600 MHz), and an NVIDIA RTX 4090 GPU (24 GB GDDR6X memory). The software environment consisted of Ubuntu 22.04 LTS (64-bit), Python 3.10.12, PyTorch 2.1.0 for automatic differentiation and neural network training, NumPy 1.24.3 for numerical computations, SciPy 1.11.2 for the BFGS optimisation routines, and SymPy 1.12 for symbolic expression manipulation and algebraic simplification. All experiments were executed with fixed random seeds for reproducibility, and the complete source code, benchmark definitions, and experimental configurations are publicly available to facilitate independent validation and extension.

The symbolic library L was constructed from a standard set of mathematical operators: arithmetic operations {+, −, ×, ÷}, elementary functions {sin, cos, exp, log}, and the identity function. The library cardinality $|L| = 9$ provides sufficient expressiveness for the benchmark problems while maintaining computational tractability. Benchmark PDEs were selected from

canonical equations in mathematical physics (Poisson, heat, wave) as recommended by Takamoto et al. (2022) for evaluating scientific machine learning methods. Ground-truth solutions were analytically derived and verified using symbolic differentiation before experimental execution.

Data Collection Method

The benchmark differential equations and their deterministic conditions (boundary and initial conditions) were defined analytically based on canonical PDE forms from mathematical physics. For each benchmark problem, the governing PDE, boundary conditions, and initial conditions (for time-dependent problems) were specified in closed form, and the corresponding ground-truth solution û was derived analytically through direct integration or the method of characteristics, as appropriate. The spatial domain was uniformly discretised as $\Omega = [-1, 1]^n$ with $n \in \{2, 3\}$, and the temporal domain for evolution equations was set to [0, 1].

Collocation points for evaluating the physical constraint residuals were sampled using a stratified random sampling strategy to ensure uniform coverage of the computational domain. For spatial domains, 2,500 collocation points were sampled in 2D problems (50 × 50 grid) and 4,000 points in 3D problems (20 × 20 × 10 grid). For temporal domains, 100 equispaced points were used. Boundary condition points were sampled independently along each boundary face with 200 points per face in 2D and 150 points per face in 3D. Initial condition points coincided with the spatial collocation points at t = 0. This sampling strategy balances computational efficiency with sufficient resolution to accurately estimate the mean squared error of each constraint component.

Analysis Technique

Statistical analysis of experimental results follows a rigorous protocol designed to ensure reliability and reproducibility. Each reported metric is computed as the mean ± standard error over 20 independent experimental runs, with each run initialised from a different random seed. The 20-run protocol was selected to provide stable estimates of mean performance while maintaining computational feasibility; preliminary analysis confirmed that 20 runs yield standard errors below 5% of the mean for all primary metrics. Confidence intervals are reported at the 95% level, computed using the Student's t-distribution with 19 degrees of freedom.

The perfect recovery rate (PRE) metric is computed as the proportion of the 20 independent runs in which the algorithm produces an expression whose symbolic skeleton is exactly equivalent to the ground-truth skeleton

(after algebraic simplification via SymPy) and whose constants satisfy the differential equation to within a mean squared error of 10^{-8}. This combined criterion ensures that reported recoveries are both structurally correct and numerically accurate. Statistical significance of performance differences between PCSRL and baseline methods is assessed using two-tailed Welch's t-tests, with a significance threshold of $\alpha = 0.05$. Effect sizes are reported using Cohen's d to quantify the practical significance of observed differences.

Expression Skeleton Generator

The expression generator component of PCSRL employs a single-layer RNN to produce symbolic expression trees by generating pre-order traversal sequences over a token library L. At each step i, the network receives a representation of the parent and sibling nodes of the current position in the tree (Petersen et al., 2020), producing a probability distribution over L from which a token τ_i is sampled. The autoregressive likelihood of a complete expression û is:

$$p(\hat{u}|\theta) = \prod_i p(\tau_i|\tau_1{:}(i-1); \theta), \quad (7)$$

Structural constraints on valid expression trees such as the arity of operator tokens and the prohibition of constant-only subtrees are enforced through a masking procedure applied to the raw output distribution at each step. This masking reduces the effective search space without biasing the policy (Mundhenk et al., 2021). Constants c in the expression tree are treated as free parameters to be optimised by a downstream procedure rather than generated directly by the RNN, a design choice that cleanly separates the combinatorial structure learning problem from the continuous parameter optimisation problem (Li et al., 2024).

Physics-Constrained Local Constant Optimisation (PLCO)

Given an expression skeleton û with free constant parameters, PLCO determines numerical values for those constants that minimise the physical constraint loss L_phys(û). This is implemented using the Broyden–Fletcher–Goldfarb–Shanno (BFGS) quasi-Newton algorithm (Fletcher, 2000), with gradients of the DE residual computed by forward-mode automatic differentiation through the expression tree. A critical practical observation is that a large fraction of sampled expression skeletons violate boundary or initial conditions irrespective of constant values. Computing the full PDE residual for such skeletons wastes computational resources. PLCO therefore implements a two-stage protocol: in Stage 1, constants are optimised against the boundary and initial condition losses only (L'_s-t = λ_1MSE_B + λ_2MSE_I); if the resulting reward from Stage 1 exceeds a minimum threshold $\tilde{R}$_ε, then

Stage 2 proceeds to full constraint optimisation including the PDE residual. This approach embeds the spirit of hard constraint satisfaction (Lu et al., 2021b) within an otherwise soft-constraint optimisation framework.

The mathematical justification for this staged approach rests on a monotonicity observation: if û satisfies the boundary and initial conditions, it is a necessary (though not sufficient) condition for satisfying the full PDE system. Therefore, filtering on boundary and initial condition violation before evaluating PDE residuals does not introduce false negatives but substantially reduces the computational cost of the top-ε selection step (Petersen et al., 2020).

Risk-Seeking Policy Gradient Training

Standard policy gradient methods maximise expected reward, which is appropriate for problems where the objective is long-run average performance. In the symbolic DE-solving context, however, the objective is to find at least one exact solution, making the best-case performance the relevant criterion (Petersen et al., 2020). PCSRL therefore employs the risk-seeking policy gradient objective:

$$J_{risk}(\theta; \varepsilon) \approx E_{\hat{u}\sim p(\hat{u}|\theta)} [R(\hat{u}) \mid R(\hat{u}) \geq R_\varepsilon(\theta)], \quad (8)$$

where $R_\varepsilon(\theta)$ denotes the $(1-\varepsilon)$-quantile of the reward distribution under the current policy θ. The gradient of this objective with respect to θ is estimated from the top-ε fraction of sampled expressions in each batch. To prevent premature convergence of the policy to a narrow region of expression space, hierarchical entropy regularisation is added to the learning objective (Landajuela et al., 2021). The entropy term is defined over the discounted sum of per-token probability distributions, with a discount factor γ that reduces the weight of tokens at greater tree depths reflecting the diminishing marginal contribution of structural diversity at the leaves of deep expression trees. The combined policy gradient estimate applied at each iteration is:

$$\nabla_\theta J_{total} = E[(R(\hat{u}) - R_\varepsilon(\theta)) \ \nabla_\theta \log p(\hat{u}|\theta) \mid \tilde{R}(\hat{u}) \geq \tilde{R}_\varepsilon] + \lambda_H \ \nabla_\theta H(\theta). \quad (9)$$

Dimension-Recursive Decomposition for High-Dimensional PDEs

For a PDE in d variables $x = (x_1, x_2, \ldots, x_d)$, the symbolic search space has cardinality $|L|^{|\tau|}$ where $|\tau|$ is the expression length—a quantity that grows exponentially with d and $|\tau|$. The dimension-recursive decomposition strategy circumvents this explosion by reformulating the d-dimensional search as d sequential one-dimensional searches, each conducted over a substantially smaller space (Cao et al., 2024; Liang & Yang, 2022).

The core idea is the parametric expression representation: at step k of the recursion, the partial expression in variables x_1, ..., x_k is written û(x_1, ..., x_k; α_k(x_{-{1,...,k}})), where α_k denotes a vector of parameters that may themselves be functions of the remaining variables. The k-th round of the recursion identifies the structural dependence of α_{k-1} on x_k via a one-dimensional PCSRL search, producing a parametric expression in x_k and new parameters α_k. A parameter α_k[j] is promoted to a fixed constant if the variance of its values across the dataset falls below a threshold δ; otherwise, it is retained as a functional parameter for the next recursion level. This variance-based distinguishability criterion provides a principled, data-driven mechanism for disentangling fixed and variable components of the symbolic expression without requiring prior knowledge of the solution structure (Cao et al., 2024).

At the final recursion level (k = d), all parameters have been expressed as explicit functions of their respective variables, yielding the complete closed-form expression. A final global BFGS optimisation over all free constants under the full physical constraint set L_phys refines the numerical accuracy of the discovered expression. The overall computational complexity of the recursive strategy scales as O(d · T_1D), where T_1D is the time required for a single one-dimensional PCSRL search, compared to O(T_{dD}) ≫ O(d · T_1D) for the direct d-dimensional search. This represents a qualitative improvement in tractability: whereas the direct approach fails for d ≥ 3 in our experiments, the recursive approach consistently succeeds for all tested dimensionalities.

RESULT AND DISCUSSION

Benchmark Problems

To provide a comprehensive and rigorous evaluation, PCSRL is tested on six benchmark PDEs spanning three equation types (Poisson, heat, and wave), two spatial dimensionalities (2D and 3D), and both linear and nonlinear character. All equations are posed on the hypercubic domain $\Omega = [-1,1]^n$ with Dirichlet boundary conditions and, for time-dependent problems, initial conditions at t = 0. Table 1 specifies each benchmark problem, its deterministic conditions, and its ground-truth solution. These benchmarks have been selected to reflect the canonical equation types appearing throughout mathematical physics, as recommended by Takamoto et al. (2022) for evaluating scientific machine learning methods.

Table 1. Benchmark differential equations, deterministic conditions, and ground-truth closed-form solutions

Problem Name	PDE	Ground-Truth Solution û
Poisson-2D	$\partial^2 u/\partial x_1^2 + \partial^2 u/\partial x_2^2 = 30x_1^2 - 7.8x_1 + 1$	$2.5x_1^4 - 1.3x_1^3 + 0.5x_2^2 - 1.7x_2$
Poisson-3D	$\partial^2 u/\partial x_1^2 + \partial^2 u/\partial x_2^2 + \partial^2 u/\partial x_3^2 = 30x_1^2 - 7.8x_2 + 1$	$2.5x_1^4 - 1.3x_2^3 + 0.5x_3^2$
Heat-2D	$\partial u/\partial t - \Delta u = -30x_1^2 + 7.8x_2 + t$	$2.5x_1^4 - 1.3x_2^3 + 0.5t^2$
Heat-3D	$\partial u/\partial t - \Delta u = -30x_1^2 + 7.8x_2 - 2.7$	$2.5x_1^4 - 1.3x_2^3 + 0.5x_3^2 - 1.7t$
Wave-2D	$\partial^2 u/\partial t^2 - \Delta u = -u - u^3 + f(x,t)$ [nonlinear]	$\exp(x_1^2)\sin(x_2)e^{(-0.5t)}$
Wave-3D	$\partial^2 u/\partial t^2 - \Delta u = u^2 - g(x,t)$ [nonlinear]	$\exp(x_1^2+x_3^2)\cos(x_2)e^{(-0.5t)}$

Note. Domains: spatial $[-1,1]^n$, temporal [0,1]. BCs: Dirichlet. ICs applied at t = 0.

Evaluation Metrics

Two complementary evaluation metrics are employed, following the methodology of Petersen et al. (2020) and Cao et al. (2024). The first is the mean physical constraint error, L_PHY(û), defined as the average root mean squared error (RMSE) of the candidate solution across all physical constraints:

$$L\_PHY = mean(\sqrt{MSE\_F} + \sqrt{MSE\_B} + \sqrt{MSE\_I}). \quad (10)$$

This metric quantifies how well the identified expression satisfies the governing physics; lower values indicate closer adherence to physical law. The second metric is the perfect recovery rate (PRE), defined as the proportion of 20 independent experimental runs in which the algorithm produces an expression whose symbolic skeleton is exactly equivalent to the ground-truth skeleton (after algebraic simplification) and whose constants satisfy the differential equation to within a mean squared error of 10^{-8}. This combined criterion ensures that reported recoveries are both structurally correct and numerically accurate, unlike purely numerical benchmarks that may reward approximate solutions that happen to have small residuals.

Comparative Methods

PCSRL is benchmarked against four state-of-the-art methods representative of the principal approaches to closed-form DE solving. PR-GPSR

(Oh et al., 2023) applies physics-regularised genetic programming directly to the differential equation residual. KAN (Liu et al., 2025) employs Kolmogorov–Arnold Networks with automatic symbolic regression activated through network symbolisation. PINN+DSR (Majumdar et al., 2023) first obtains a numerical solution via PINN and then applies Deep Symbolic Regression to the numerical data. FEX (Liang & Yang, 2022) uses an RL-based framework to discover finite mathematical expressions as approximate solutions. Wolfram Mathematica (Version 14.2, 2024) serves as a rule-based analytical solver baseline. All methods are evaluated under consistent time budgets and with matching symbolic libraries to ensure comparability.

Main Comparative Results

Table 2 presents the comparative performance of all methods on the six PDE benchmarks, reporting L_PHY (lower is better) and PRE (higher is better) with 95% confidence intervals derived from 20 independent runs. PCSRL achieves a perfect recovery rate of 100% across all six benchmarks, recovering the exact symbolic skeleton and accurate constants in every trial. All competing methods fail to achieve any symbolic recovery (PRE = 0%) on any benchmark, confirming that the structural innovations of PCSRL are essential rather than merely incremental improvements.

Table 2. Comparative performance of PCSRL and baselines on six PDE benchmarks (mean ± standard error, 20 runs)

Problem	PCSRL L_PHY	PCSRL PRE	PR-GPSR L_PHY	PR-GPSR PRE	KAN L_PHY	KAN PRE	PINN+DSR L_PHY	PINN+DSR PRE
Poisson-2D	7.2E-5 ± 2.6E-6	100 %	1.9E-2 ± 2.3E-3	0%	7.2E+0 ± 1.5E+0	0%	5.7E-1 ± 7.9E-2	0%
Poisson-3D	5.7E-5 ± 1.3E-6	100 %	1.8E-1 ± 1.8E-2	0%	6.2E+0 ± 1.5E+0	0%	2.6E+0 ± 1.1E-1	0%
Heat-2D	1.5E-6 ± 7.6E-7	100 %	2.2E-1 ± 1.3E-2	0%	1.2E+1 ± 1.2E+0	0%	3.2E+0 ± 1.4E-1	0%

Problem	PCSRL L_PHY	PCSRL PRE	PR-GPSR L_PHY	PR-GPSR PRE	KAN L_PHY	KAN PRE	PINN+DSR L_PHY	PINN+DSR PRE
Heat-3D	1.1E-5 ± 1.2E-6	100 %	8.9E-1 ± 3.0E-1	0%	1.2E+1 ± 1.7E+0	0%	2.7E+0 ± 3.2E-1	0%
Wave-2D	4.3E-5 ± 2.4E-6	100 %	4.2E-1 ± 3.7E-1	0%	2.0E+0 ± 1.4E-1	0%	9.6E-1 ± 2.1E-1	0%
Wave-3D	8.2E-6 ± 1.6E-6	100 %	9.9E-1 ± 2.5E-1	0%	1.4E+1 ± 2.4E+0	0%	1.3E+1 ± 4.2E+0	0%

Note. L_PHY = physical constraint RMSE (↓ better); PRE = perfect recovery rate (↑ better).

The relative L_2 error metric, defined as the mean across test samples of $||u_{true} - \hat{u}||_2 / ||u_{true}||_2$ (Takamoto et al., 2022), provides an alternative, magnitude-normalised perspective on solution accuracy. Table 3 reports relative L_2 errors for PCSRL alongside all comparison methods, including Mathematica. Consistent with the L_PHY results, PCSRL achieves the lowest relative L_2 errors across all benchmarks typically two to five orders of magnitude below those of the best-performing alternative. Mathematica fails to produce any valid closed-form solutions for the benchmark problems, reflecting its known limitations with synthetic, non-classical problem structures.

Table 3. Relative L_2 errors for PCSRL and baselines on all benchmark PDEs

Problem	PCSRL	Mathematica	PR-GPSR	KAN	FEX	PINN
Poisson-2D	1.55E-05	✗	1.49E-02	3.24E+00	2.00E-02	6.24E-04
Poisson-3D	3.58E-06	✗	2.01E-02	2.57E+00	9.13E-02	3.93E-03
Heat-2D	4.45E-06	✗	4.57E-02	6.54E+00	4.70E-02	6.98E-03

Problem	PCSRL	Mathematica	PR-GPSR	KAN	FEX	PINN
Heat-3D	9.34E-06	✗	1.39E+00	5.58E+00	3.63E-01	1.15E-02
Wave-2D	3.11E-05	✗	1.80E-01	1.18E+00	2.34E-01	1.34E-02
Wave-3D	7.62E-06	✗	4.56E-01	7.26E+00	3.97E-01	3.69E-02

These results call for a careful interpretive discussion of why the gaps are so large. For GP-based methods, the failure mode is the exponential growth of the search space in higher dimensions combined with the tendency of GP trees to grow in complexity without corresponding accuracy gains (Oh et al., 2023). For KAN, the failure mode is the loss of precision during the symbolisation step, where continuous network activations are approximated by discrete symbolic functions a process that degrades particularly severely for multi-variable solutions with coupled terms (Liu et al., 2025).

For PINN+DSR, the failure is attributable to error propagation: the PINN solution carries an irreducible approximation error (see Table 7 in the appendix of the original study for PINN convergence errors of up to 2.88E-03), and DSR then fits a symbolic expression to these erroneous data, introducing a second layer of approximation. The compound effect is sufficient to prevent exact symbolic recovery in virtually all cases.

Mapping Results to Research Objectives

To ensure coherence between the stated research objectives and the empirical findings, we explicitly map each result to its corresponding objective. Research Objective 1 to present PCSRL as a complete methodological framework for physics-constrained symbolic regression—is validated by the comprehensive experimental protocol described in Sections 4.1–4.5 and the consistent successful application across all six benchmark problems (Table 2).

Research Objective 2 to demonstrate that the two-stage PLCO protocol accelerates convergence without sacrificing accuracy is supported by the ablation results in Section 6.2, which show a 4.65× mean speedup and the maintenance of 100% recovery rates when RSCO is active. Research Objective 3 to develop a dimension-recursive decomposition that enables scalable high-dimensional PDE solving is confirmed by the 100% recovery on 3D benchmarks (Poisson-3D, Heat-3D, Wave-3D) where the non-recursive configuration achieves 0% recovery (Section 6.1). This explicit mapping demonstrates that each stated objective has been empirically fulfilled.

Ablation Study and Complexity Analysis

Component Ablation

To isolate the contribution of each major component of PCSRL, a series of ablation experiments is conducted in which individual components are disabled while all other aspects of the system remain unchanged. Three configurations are compared: the full PCSRL system; PCSRL without the risk-seeking constant optimization (w/o RSCO), which replaces the two-stage PLCO protocol with standard single-stage BFGS on the full constraint loss; and PCSRL without the dimension-recursive decomposition (w/o RecurExp), which attempts direct multi-dimensional symbolic search. All configurations are allocated equal computation time for a fair comparison.

The full PCSRL system achieves 100% recovery across all six benchmarks. Removing RSCO causes recovery to drop to between 40% and 80% on different benchmarks, with the most dramatic degradation on the higher-dimensional problems where the computational savings from RSCO's boundary condition pre-filtering are largest. Removing RecurExp causes recovery to fall to 0% on all three-dimensional benchmarks, confirming that the recursion is not merely a convenient optimization but an essential algorithmic element: without it, the algorithm is categorically unable to solve high-dimensional problems within the allocated time budget.

Efficiency Analysis of RSCO

The computational efficiency gain attributable to RSCO is quantified across eight benchmark problems. Over 20 independent trials per problem, PCSRL with RSCO achieves a mean solution discovery time of approximately 200 seconds, compared to approximately 934 seconds for PCSRL without RSCO a 4.65× speedup. This speedup is achieved despite RSCO requiring approximately 22% more expression evaluations per iteration, reflecting the fact that the pre-filtering step eliminates numerous expensive full-physics evaluations of expressions that fail the boundary and initial condition check. The practical implication is that RSCO enables more iterations within the same time budget, leading to better exploration and faster convergence to valid solutions.

Table 4. Mean convergence time (seconds) and expression evaluations for PCSRL with and without RSCO across Γ-dataset benchmarks (mean over 20 runs)

Benchmark	PCSRL Time (s)	w/o RSCO Time (s)	Speedup	PCSRL Expressions	w/o RSCO Expressions
Γ_1	23	51	2.22×	∼18,000	∼14,000
Γ_2	85	233	2.74×	∼48,000	∼39,000
Γ_3	233	438	1.88×	∼110,000	∼89,000
Γ_4	438	789	1.80×	∼200,000	∼162,000
Γ_5	336	1,750	5.21×	∼175,000	∼142,000
Γ_6	34	1,768	51.99×	∼22,000	∼18,000
Γ_7	426	100	0.23×*	∼220,000	∼178,000
Γ_8	51	179	3.51×	∼32,000	∼26,000

*Note. *Γ_7 is an anomalous case discussed in Section 6.3.*

Gamma7 Anomaly Explanation

The anomalous result for benchmark Γ_7 (speedup of 0.23×, indicating that PCSRL with RSCO is slower than without RSCO) warrants a detailed explanation. Γ_7 represents a benchmark problem with a solution structure that happens to satisfy boundary conditions for a large fraction of randomly generated expression skeletons. In this specific case, the Stage 1 boundary-condition pre-filter of RSCO provides minimal computational savings because most candidate expressions pass the boundary condition check. However, RSCO still incurs the overhead of the two-stage optimisation protocol: each expression requires two separate BFGS optimisations (first for boundary/initial conditions, then for full physics) rather than a single full optimisation. When the pass-through rate of Stage 1 exceeds approximately 85%, this overhead dominates the computational savings from eliminated full-physics evaluations, resulting in net slower performance.

This anomaly is structurally informative: it identifies the regime in which RSCO's two-stage design transitions from beneficial to detrimental. For problems where boundary condition satisfaction is rare among random expression skeletons the typical case for complex, high-dimensional PDEs RSCO provides substantial speedups. For simpler problems with fortuitous boundary-condition structure, the overhead may exceed the savings. This observation motivates an adaptive RSCO variant that dynamically disables Stage 1 when the empirical pass-through rate exceeds a threshold, which we identify as a promising direction for future optimisation (see Section 7.3).

Symbolic Library Complexity Scaling

The computational complexity of symbolic search is strongly influenced by the size and composition of the symbolic library L. To characterize this relationship, PCSRL is evaluated on benchmarks Γ_1–Γ_4 using three progressively richer libraries: a minimal set {+, −, ×, ÷}; an intermediate set that adds {sin, cos}; and the full library {+, −, ×, ÷, sin, cos, exp, log}. Both mean solution time and mean expression count are recorded across 20 independent trials per configuration.

The results reveal a strongly superlinear relationship between library size and solving complexity. Adding {sin, cos} to the minimal library approximately doubles both time and expression count. Adding {exp, log} to the trigonometric library introduces a substantially larger increase, often an order of magnitude, particularly for the higher-complexity benchmarks Γ_3 and Γ_4. This disproportionate effect of exponential and logarithmic operators is attributable to the close structural similarity between polynomial expressions which characterize the ground-truth solutions for these benchmarks and low-degree exponential or logarithmic approximations, making the discriminative task of the RNN significantly harder (Li et al., 2024). These findings motivate the use of problem-specific library pruning as a practical strategy for reducing solving complexity when domain knowledge is available.

Gamma7 Anomaly Explanation

The anomalous result for benchmark Γ_7 (speedup of 0.23×, indicating that PCSRL with RSCO is slower than without RSCO) warrants a detailed explanation. Γ_7 represents a benchmark problem with a solution structure that happens to satisfy boundary conditions for a large fraction of randomly generated expression skeletons. In this specific case, the Stage 1 boundary-condition pre-filter of RSCO provides minimal computational savings because most candidate expressions pass the boundary condition check. However, RSCO still incurs the overhead of the two-stage optimisation protocol: each expression requires two separate BFGS optimisations (first for boundary/initial conditions, then for full physics) rather than a single full optimisation. When the pass-through rate of Stage 1 exceeds approximately 85%, this overhead dominates the computational savings from eliminated full-physics evaluations, resulting in net slower performance.

This anomaly is structurally informative: it identifies the regime in which RSCO's two-stage design transitions from beneficial to detrimental. For problems where boundary condition satisfaction is rare among random expression skeletons the typical case for complex, high-dimensional PDEs RSCO provides substantial speedups. For simpler problems with fortuitous

boundary-condition structure, the overhead may exceed the savings. This observation motivates an adaptive RSCO variant that dynamically disables Stage 1 when the empirical pass-through rate exceeds a threshold, which we identify as a promising direction for future optimisation (see Section 7.3).

Symbolic Library Complexity Scaling

The computational complexity of symbolic search is strongly influenced by the size and composition of the symbolic library L. To characterize this relationship, PCSRL is evaluated on benchmarks Γ_1–Γ_4 using three progressively richer libraries: a minimal set {+, −, ×, ÷}; an intermediate set that adds {sin, cos}; and the full library {+, −, ×, ÷, sin, cos, exp, log}. Both mean solution time and mean expression count are recorded across 20 independent trials per configuration.

The results reveal a strongly superlinear relationship between library size and solving complexity. Adding {sin, cos} to the minimal library approximately doubles both time and expression count. Adding {exp, log} to the trigonometric library introduces a substantially larger increase, often an order of magnitude, particularly for the higher-complexity benchmarks Γ_3 and Γ_4. This disproportionate effect of exponential and logarithmic operators is attributable to the close structural similarity between polynomial expressions which characterize the ground-truth solutions for these benchmarks and low-degree exponential or logarithmic approximations, making the discriminative task of the RNN significantly harder (Li et al., 2024). These findings motivate the use of problem-specific library pruning as a practical strategy for reducing solving complexity when domain knowledge is available.

Discussion

Theoretical Implications

The consistent symbolic recovery achieved by PCSRL has several notable theoretical implications. First, it demonstrates that an RL agent, trained without any prior knowledge of the solution structure beyond the differential equation and its boundary and initial conditions, can converge to exact symbolic solutions for a broad class of PDEs. This validates the theoretical claim of Petersen et al. (2020) that risk-seeking policy gradients can efficiently explore combinatorial symbolic spaces and extends its scope from purely data-driven regression to physics-constrained DE solving.

Second, the dimension-recursive decomposition provides an empirical confirmation of a key theoretical intuition: that the solutions to high-dimensional PDEs with separable or quasi-separable structure can be meaningfully decomposed into sequences of one-dimensional problems.

While this structure is not universally present, it covers a substantial fraction of the canonical PDEs in mathematical physics, including those studied here. The variance-based criterion for parameter promotion, while heuristic in nature, provides an efficient and interpretable mechanism for identifying this separability structure from data (Cao et al., 2024).

Third, the staged constraint optimization in PLCO establishes a formal connection between the hard constraint embedding philosophy of Lu et al. (2021b) and the soft constraint optimization framework of standard PINNs. The two-stage approach can be understood as implementing a structured optimization landscape: initial feasibility with respect to boundary conditions defines a reduced subspace within which PDE residual minimization is subsequently conducted. This structured landscape is easier to navigate than the unstructured joint minimization of all constraint components simultaneously, providing a compelling theoretical rationale for the observed computational efficiency gains.

Practical Applications and Broader Significance

The practical significance of PCSRL extends across several high-impact domains. In mathematical physics, the ability to automatically discover closed-form solutions to PDEs including nonlinear wave equations and multi-dimensional heat equations could substantially accelerate theoretical development in areas such as plasma physics, quantum mechanics, and general relativity, where exact analytical solutions provide irreplaceable physical insight (Karniadakis et al., 2021). In engineering, closed-form solutions enable fast, accurate evaluation of physical responses in design optimization contexts where numerical simulation is prohibitively slow. In scientific machine learning more broadly, the present results suggest that the combination of symbolic representation with physical constraints defines a powerful and underexplored paradigm for knowledge discovery from differential equations.

The interpretability of closed-form solutions is also of direct relevance to the growing field of explainable artificial intelligence in scientific contexts (Blechschmidt & Ernst, 2021). Unlike black-box neural networks, a symbolic expression recovered by PCSRL can be inspected, analyzed, and communicated in the language of mathematics the lingua franca of science and engineering. This transparency facilitates error detection, physical reasoning, and extrapolation beyond the training domain in ways that are fundamentally unavailable to numerical approximators.

Limitations and Future Directions

Several important limitations of PCSRL as presented here merit explicit acknowledgement. First, the recursive decomposition strategy implicitly assumes a degree of dimensional separability in the solution structure. For strongly coupled, non-separable high-dimensional PDEs—such as those arising in turbulence or coupled multi-physics systems—the recursive approach may fail to converge or may produce only approximate symbolic forms. Developing more general decomposition strategies for non-separable problems remains an important open challenge.

Second, the symbolic library must be specified a priori, and the computational complexity scales superlinearly with library size, as demonstrated in Section 6.4. In applications where the appropriate function classes are not known in advance, an adaptive library construction mechanism potentially informed by domain-specific physical reasoning or learned library preferences would substantially broaden the practical applicability of the framework (Li et al., 2024).

Third, the PCSRL framework, as described, handles scalar-valued unknown functions. Extension to systems of coupled PDEs, vector-valued unknowns, and PDEs with variable coefficients or stochastic terms represents a natural and important direction for future work. Theoretical analysis of the convergence rates and sample complexity of the risk-seeking policy gradient in the symbolic DE-solving context would also provide valuable underpinning for the empirical results presented here (Landajuela et al., 2021; Sun et al., 2023).

CONCLUSION

This paper has presented PCSRL, a physics-constrained symbolic regression framework for the machine discovery of closed-form solutions to differential equations. The framework integrates three mutually reinforcing innovations: an RNN-based expression generator trained with risk-seeking policy gradients; a two-stage constant optimization protocol that applies boundary and initial conditions as a pre-filter before full PDE residual minimization; and a dimension-recursive decomposition strategy that enables efficient symbolic search in high-dimensional PDE spaces. The results establish several important scientific conclusions. The combination of risk-seeking policy gradients and physics constraint evaluation is a uniquely effective strategy for navigating the combinatorial space of symbolic expressions under physical constraints.

The practical implications of this work are substantial. PCSRL enables engineers and scientists to obtain transparent, analytically tractable models

for complex physical systems, facilitating rapid design optimisation, stability analysis, and knowledge discovery in domains where black-box numerical approximations are insufficient. The methodology contributes to the broader movement toward explainable AI in scientific contexts, where interpretability is not merely desirable but essential for validation, communication, and trust.

REFERENCES

Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A., & Parascandolo, G. (2021). Neural symbolic regression that scales. In Proceedings of the 38th International Conference on Machine Learning (pp. 936–945). PMLR. https://proceedings.mlr.press/v139/biggio21a.html

Blechschmidt, J., & Ernst, O. G. (2021). Three ways to solve partial differential equations with neural networks: A review. GAMM-Mitteilungen, 44(2), Article e202100006. https://doi.org/10.1002/gamm.202100006

Boudouaoui, Y., Habbi, H., Ozturk, C., & Karaboga, D. (2020). Solving differential equations with artificial bee colony programming. Soft Computing, 24(24), 17991–18007. https://doi.org/10.1007/s00500-020-05048-3

Cao, L., Zheng, Z., Ding, C., Cai, J., & Jiang, M. (2023). Genetic programming symbolic regression with simplification-pruning operator for solving differential equations. In Proceedings of the International Conference on Neural Information Processing (pp. 287–298). Springer.

Cao, L., Liu, Y., Wang, Z., Xu, D., Ye, K., Tan, K. C., & Jiang, M. (2024). An interpretable approach to the solutions of high-dimensional partial differential equations. In Proceedings of the AAAI Conference on Artificial Intelligence, 38, 20640–20648. https://doi.org/10.1609/aaai.v38i19.29991

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics-informed neural networks: Where we are and what's next. Journal of Scientific Computing, 92(3), Article 88. https://doi.org/10.1007/s10915-022-01939-z

Evans, L. C. (2022). Partial differential equations (2nd ed., Vol. 19). American Mathematical Society.

Fletcher, R. (2000). Practical methods of optimization (2nd ed.). Wiley.

Jiang, N., & Xue, Y. (2023). Symbolic regression via control variable genetic programming. In Machine Learning and Knowledge Discovery in Databases: Research Track (pp. 178–195). Springer Nature Switzerland.

Kamali, M., Kumaresan, N., & Ratnavelu, K. (2015). Solving differential equations with ant colony programming. Applied Mathematical

Modelling, 39(10), 3150–3163. https://doi.org/10.1016/j.apm.2014.11.003

Kamienny, P.-A., d'Ascoli, S., Lample, G., & Charton, F. (2022). End-to-end symbolic regression with transformers. Advances in Neural Information Processing Systems, 35, 10269–10281.

Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. Nature Reviews Physics, 3(6), 422–440. https://doi.org/10.1038/s42254-021-00314-5

Koza, J. R. (1994). Genetic programming is a means for programming computers by natural selection. Statistics and Computing, 4(2), 87–112. https://doi.org/10.1007/BF00175355

Lagaris, I. E., Likas, A., & Fotiadis, D. I. (1998). Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 9(5), 987–1000. https://doi.org/10.1109/72.712178

Landajuela, M., Petersen, B. K., Kim, S., Santiago, C. P., Glatt, R., Mundhenk, N., Pettit, J. F., & Faissol, D. (2021). Discovering symbolic policies with deep reinforcement learning. In Proceedings of the 38th International Conference on Machine Learning (pp. 5979–5989). PMLR. https://proceedings.mlr.press/v139/landajuela21a.html

Li, W., Li, W., Sun, L., Wu, M., Yu, L., Liu, J., Li, Y., & Tian, S. (2023a). Transformer-based model for symbolic regression via joint supervised learning. In The 11th International Conference on Learning Representations. OpenReview. https://openreview.net/forum?id=ULzyv9M1j5

Li, W., Li, W., Yu, L., Wu, M., Sun, L., Liu, J., Li, Y., Wei, S., Deng, Y., & Hao, M. (2024). A neural-guided dynamic symbolic network for exploring mathematical expressions from data. In Proceedings of the 41st International Conference on Machine Learning (pp. 28222–28242). PMLR. https://proceedings.mlr.press/v235/li24az.html

Li, Z., Huang, D. Z., Liu, B., & Anandkumar, A. (2023b). Fourier neural operator with learned deformations for PDEs on general geometries. Journal of Machine Learning Research, 24(388), 1–26. https://jmlr.org/papers/v24/23-0064.html

Liang, S., & Yang, H. (2022). Finite expression method for solving high-dimensional partial differential equations. arXiv preprint arXiv:2206.10121. https://doi.org/10.48550/arXiv.2206.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljacic, M., Hou, T. Y., & Tegmark, M. (2025). KAN: Kolmogorov–Arnold Networks. In The 13th International Conference on Learning Representations. OpenReview.https://openreview.net/forum?id=Ozo7qJ5vZi

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021a). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. Nature Machine Intelligence, 3(3), 218–229. https://doi.org/10.1038/s42256-021-00302-5

Lu, L., Pestourie, R., Yao, W., Wang, Z., Verdugo, F., & Johnson, S. G. (2021b). Physics-informed neural networks with hard constraints for inverse design. SIAM Journal on Scientific Computing, 43(6), B1105–B1132. https://doi.org/10.1137/21M1397908

Majumdar, R., Jadhav, V., Deodhar, A., Karande, S., Vig, L., & Runkana, V. (2023). Symbolic regression for PDEs using pruned differentiable programs. arXiv preprint arXiv:2303.07009. https://doi.org/10.48550/arXiv.2303.07009

Mundhenk, T., Landajuela, M., Glatt, R., Santiago, C. P., & Petersen, B. K. (2021). Symbolic regression via deep reinforcement learning enhanced genetic programming seeding. Advances in Neural Information Processing Systems, 34, 24912–24923.

Oh, H., Amici, R., Bomarito, G., Zhe, S., Kirby, R., & Hochhalter, J. (2023). Genetic programming-based symbolic regression for analytical solutions to differential equations. arXiv preprint arXiv:2302.03175. https://doi.org/10.48550/arXiv.2302.03175

Petersen, B. K., Larma, M. L., Mundhenk, T. N., Santiago, C. P., Kim, S. K., & Kim, J. T. (2020). Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In the 8th International Conference on Learning Representations.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Randall, D. L., Townsend, T. S., Hochhalter, J. D., & Bomarito, G. F. (2022). Bingo: A customizable framework for symbolic regression with genetic programming. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 2282–2288). ACM. https://doi.org/10.1145/3520304.3534008 /

Mazumdar, A., Otayagich, S., & Miettinen, K. (2022, July). Interactive evolutionary multiobjective optimization with modular physical user interface. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (pp. 1835-1843).

Rao, C., Ren, P., Wang, Q., Buyukozturk, O., Sun, H., & Liu, Y. (2023). Encoding physics to learn reaction–diffusion processes. Nature Machine

Intelligence, 5(7), 765–779. https://doi.org/10.1038/s42256-023-00679-3

Shojaee, P., Meidani, K., Barati Farimani, A., & Reddy, C. (2024). Transformer-based planning for symbolic regression. Advances in Neural Information Processing Systems, 36.

Sun, F., Liu, Y., Wang, J.-X., & Sun, H. (2023). Symbolic physics learner: Discovering governing equations via Monte Carlo tree search. In The 11th International Conference on Learning Representations. OpenReview. https://openreview.net/forum?id=ZTK3SefE8_Z

Takamoto, M., Praditia, T., Leiteritz, R., MacKinlay, D., Alesiani, F., Pfluger, D., & Niepert, M. (2022). PDEBench: An extensive benchmark for scientific machine learning. Advances in Neural Information Processing Systems, 35, 1596–1611.

Tsoulos, I. G., & Lagaris, I. E. (2006). Solving differential equations with genetic programming. Genetic Programming and Evolvable Machines, 7(1), 33–54. https://doi.org/10.1007/s10710-006-7009-y

Wang, S., Teng, Y., & Perdikaris, P. (2021). Understanding and mitigating gradient flow pathologies in physics-informed neural networks. SIAM Journal on Scientific Computing, 43(5), A3055–A3081. https://doi.org/10.1137/20M1318043