# SYMBOLICLM: TRAINING LANGUAGE MODELS AS SYMBOLIC REGRESSORS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Large Language Models (LLMs) have demonstrated strong capabilities in scientific reasoning, drawing growing attention to their potential in scientific discovery. However, scientific discovery fundamentally originates from deriving formal laws directly from observational data, known as Symbolic Regression (SR). This task poses a critical challenge to LLMs, stemming from an inherent tension between pre-trained LLMs' proficiency in approximate reasoning, rooted in probabilistic text generation, and the high-precision demands of SR tasks. While recent methods employ complex external scaffolds to mitigate this limitation, such an iterative agentic paradigm remains computationally inefficient and fundamentally separates the model's internal scientific knowledge from the symbolic regression process. To address this problem, we propose to directly equip LLMs with precise symbolic understanding and the ability to induce laws from raw observational data. In this paper, we introduce **PhysSymbArena**, a large-scale benchmark, comprising more than $160{,}000$ diverse equations and $1.8$B tokens of numerical symbolic data and physical description, to support post-training and extensive evaluation. Building on PhysSymbArena, we propose **SymbolicLM**, in which the symbolic regression capability of LLMs is systematically enhanced during post-training through **mathematical and physical supervision**. At inference time, we introduce a SymbolicSGA Refinement framework that uses quantitative fitting feedback to iteratively refine and recombine promising symbolic structures, enabling the model to correct structural errors and progressively recover more accurate governing equations. Experiments on multiple symbolic regression benchmarks demonstrate that SymbolicLM substantially outperforms representative symbolic regression methods in structural recovery while maintaining competitive numerical fitting performance. These results demonstrate that symbolic regression can be explicitly learned and strengthened as an intrinsic capability of LLMs.

## 1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities across a wide range of scientific tasks, including scientific question answering, literature understanding, mathematical reasoning, and scientific text generation (Taylor et al., 2022; Lewkowycz et al., 2022; Wang et al., 2023). Yet, genuine scientific discovery extends beyond reasoning over established knowledge and predefined problems; it fundamentally requires deriving exact, formal laws directly from observational data. A canonical formulation of this problem is symbolic regression (SR), which aims to discover governing equations that describe the data. Despite their reasoning power, general-purpose LLMs face distinct challenges in solving symbolic regression.

The limitations of LLMs in symbolic regression fundamentally stem from an inherent tension between their proficiency in approximate reasoning and the high-precision demands of the task. Specifically, these high-precision demands arise from the strict objective of the problem: given a set of numerical observations $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, the model must identify an analytical expression $f$ that consistently satisfies $y_i \approx f(\mathbf{x}_i)$ across all data points. While pre-trained LLMs excel at probabilistic next-token generation, where approximate plausibility is sufficient, evaluating a mathematical law admits no margin for structural error. Even a minor mistake in an operator, exponent, or variable dependency is amplified across the input domain, causing $f(\mathbf{x}_i)$ to diverge substantially from the



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

observed $y_i$. To mitigate this limitation, existing methods predominantly turn to external scaffolds, yet this reliance incurs prohibitive computational costs while still yielding suboptimal accuracy.

Existing scaffold-based methods typically rely on complex agentic workflows, where the LLM proposes candidate formulas and external modules evaluate numerical fitting (Ma et al., 2024; Grayeli et al., 2024; Shojaee et al., 2025a). However, this division of labor introduces a fundamental feedback bottleneck. In-context prompts cannot effectively translate scalar numerical residuals into precise structural algebraic edits; the model struggles to discern whether a poor fit stems from incorrect functional forms or unoptimized parameters. Consequently, the search easily falls into local optima or constructs ungrounded expressions, failing to recover the true mathematical skeleton. This blind, multi-round trial and error can take up to hours for a single equation, yet still yields structurally flawed results. To overcome this limitation, in this paper, we propose to directly instill symbolic regression capabilities into LLMs, enabling them to discover governing laws from observational data as an intrinsic capability.As illustrated in Figure 1.

To instill intrinsic symbolic regression capabilities into LLMs, we introduce a unified framework spanning physically grounded data synthesis, mathematical-physical tuning, and lightweight inference. First, we introduce **PhysSymbArena**, a large-scale benchmark featuring physically grounded synthetic data. By systematically coupling over $160{,}000$ equations ($1.8$B tokens) with physical quantities, dimensional constraints, and variable descriptions, it enables LLMs to exploit internal scientific priors rather than treating observations as ungrounded numbers. Second, building on this foundation, we develop **SymbolicLM** through a dedicated Mathematical-Physical Tuning (MPT), realized via a reinforcement fine-tuning framework. Going beyond token-level cross-entropy, this stage directly aligns policy through tailored rewards covering symbolic skeletons, physical constraints, and syntactic validity. This allows the model to produce valid, algebraically equivalent expressions without being confined to rigid token matching. Finally, at inference time, we design the SymbolicSGA refinement framework, a lightweight refinement loop. Empowered by the model's intrinsic structural awareness, SymbolicSGA eliminates the need for blind exploration, requiring only a few iterations of quantitative feedback to rectify residual errors and pinpoint exact governing equations.

Experiments demonstrate that explicitly training and aligning LLMs for symbolic regression substantially improves their ability to infer analytical structures from numerical observations. On PhysSymbArena, SymbolicLM significantly outperforms general-purpose LLM baselines. Moreover, on the external SRBench-Feynman benchmark, our method achieves strong performance against representative symbolic regression approaches, reaching an exact symbolic skeleton accuracy of $57.98\%$ while maintaining competitive numerical fitting accuracy. These results further demonstrate that the symbolic regression capability learned from PhysSymbArena can generalize beyond the training distribution to established symbolic regression benchmarks.

Our main contributions are summarized as follows:

- We introduce **PhysSymbArena**, a large-scale numerical–symbolic platform containing more than $160{,}000$ diverse equations and $1.8$ billion tokens, designed to support both the training and systematic evaluation of LLM symbolic regression capabilities.
- We propose **SymbolicLM**, which combines supervised fine-tuning with GRPO-based symbolic and physical feedback to directly align LLM generation with symbolic regression objectives, including syntactic validity, physical consistency, analytical structure, symbolic equivalence, and numerical accuracy.
- We introduce an SymbolicSGA refinement mechanism that enables the model to continuously revise its symbolic hypotheses according to quantitative experimental feedback. Extensive experiments on three symbolic regression benchmarks show that SymbolicSGA achieves strong performance in symbolic structure recovery and numerical fitting accuracy.

## 2 Related Work

**Symbolic regression.** Symbolic regression has traditionally been formulated as a search problem over the discrete space of mathematical expressions. Genetic programming and evolutionary approaches construct and modify expression trees through operations such as mutation, crossover, and

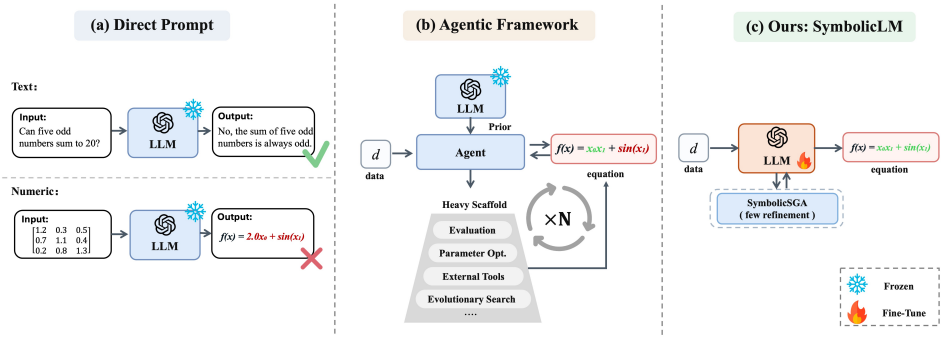


Figure 1: Comparison of different frameworks for symbolic equation discovery with LLMs.

selection, while reinforcement-learning methods such as Deep Symbolic Regression (DSR) learn policies for generating high-reward expressions (Petersen et al., 2019; Cranmer, 2023). Although effective, these approaches typically perform substantial task-specific search for each new problem. A complementary line of work seeks to amortize this search process by learning numerical-to-symbolic mappings from large collections of synthetically generated equations. NeSymReS trains a Transformer to infer symbolic expressions from numerical observations and combines neural prediction with subsequent symbolic search (Biggio et al., 2021), while E2E Symbolic Regression directly decodes complete analytical expressions from observed data (Kamienny et al., 2022). SNIP further improves numerical–symbolic representation learning through large-scale pretraining (Meidani et al., 2024).These methods demonstrate that numerical-to-symbolic mappings can be effectively learned from large-scale synthetic data. However, they are designed specifically for symbolic regression, whereas whether general-purpose LLMs can be explicitly trained to perform symbolic regression directly from numerical observations remains less explored.

**LLMs for symbolic regression.** The emergence of general-purpose LLMs has introduced a new paradigm for symbolic regression by leveraging the mathematical and scientific priors acquired during large-scale pretraining. Rather than training models specifically for numerical-to-symbolic prediction, recent approaches typically use LLMs to propose candidate equations or scientific hypotheses and combine them with external search, numerical optimization, or iterative refinement. ICSR adopts an iterative prompting strategy, guiding the LLM to propose and revise functional forms according to numerical fitting errors (Merler et al., 2024). LLM-SR represents candidate equations as programs and combines LLM-generated hypotheses with evolutionary search and numerical parameter optimization (Shojaee et al., 2025a). SGA similarly employs LLMs to reason over discrete scientific hypotheses while delegating continuous parameter optimization to differentiable procedures (Ma et al., 2024), whereas LaSR further improves hypothesis search by extracting reusable concepts from previously discovered high-performing candidates (Grayeli et al., 2024). These methods demonstrate that the scientific priors encoded in LLMs can substantially improve equation discovery. However, under this paradigm, LLMs primarily provide candidate structures or search priors, while a substantial portion of the symbolic regression process still relies on external search and optimization mechanisms. Consequently, the LLM itself is not explicitly trained to directly perform symbolic regression from numerical observations.

## 3 PhysSymbArena: Dataset Construction

We introduce **PhysSymbArena**, a large-scale symbolic regression dataset comprising 160,000 analytical expressions with physical descriptions and numerical observations, totaling 1.8B tokens. Each sample is formalized as a tuple $(\mathcal{D}, \mathcal{C}, f)$, where $f : \mathbb{R}^D \to \mathbb{R}$ denotes the target expression $(y = f(\mathbf{x}))$, $\mathcal{C}$ encapsulates physical semantics and dimensional metadata, and $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^{N}$ represents empirical observations. As illustrated in Figure 3, dataset construction proceeds across two steps: (1) *numerical equation generation and observation sampling*, and (2) *physical grounding and dimensional validation*. Comprehensive details could be seen in Appendix.

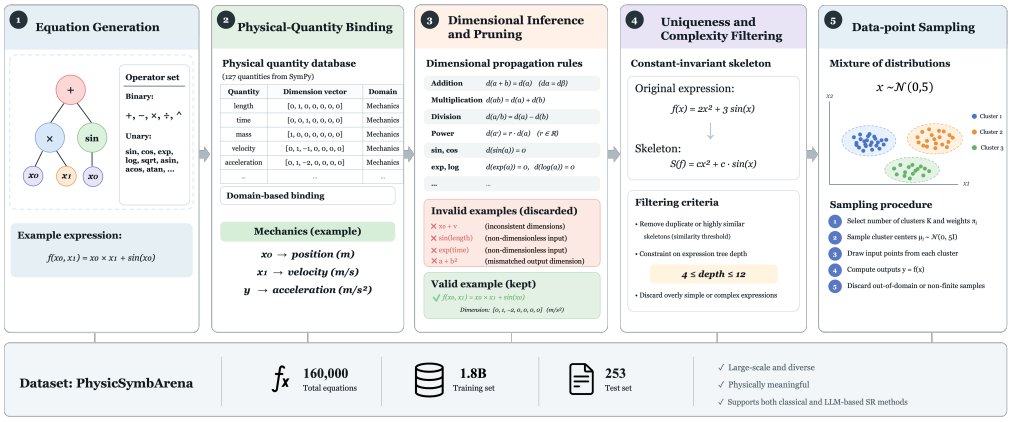


Figure 2: Overview of our proposed method.

**Numerical Equation Generation and Observation Sampling.** We synthesize mathematical expressions and their corresponding numerical observations through a randomized tree-based simulation pipeline (Kamienny et al., 2022; Meidani et al., 2024). Specifically, candidate expressions $f$ are constructed as trees whose depths and mathematical operators are randomly drawn from predefined libraries to introduce diverse structural complexities. For each generated expression, we sample input vectors across the domain and evaluate $f$ to obtain the corresponding responses, formulating the numerical observation set $\mathcal{D}$.

**Physical Grounding and Dimensional Validation.** To endow abstract expressions with real-world physical semantics, we curate a dictionary of 127 physical quantities across domains such as mechanics, electromagnetism, and thermodynamics from SymPy. For each expression, candidate physical quantities from a shared domain are randomly assigned to the input variables $\mathbf{x}$ and the target output $y$. We then perform bottom-up dimensional propagation along the expression tree under standard SI rules: addition and subtraction require identical dimensions, transcendental functions require dimensionless arguments, and the propagated root dimension must match that of $y$. Assignments that violate dimensional consistency are discarded and resampled, ensuring that the resulting physical metadata $\mathcal{C}$ strictly complies with physical laws.

Table 1: Comparison of symbolic regression benchmarks. SymbArena distinguishes itself through its massive scale, the inclusion of a train/test split, and its support for both traditional and LLM-based methods.

| Benchmarks | Numbers of Equations | Train & Test | Supported Methods |
|---|---|---|---|
| Nguyen | 12 | ✘ | Traditional only |
| Rational | 3 | ✘ | Traditional only |
| SRbench | 252 | ✘ | Traditional only |
| LLM-SRbench | 239 | ✘ | LLM-based only |
| SymbArena | 160,000 | ✔ | Both |

**Dataset Statistics** Table 1 compares PhysSymbArena with several widely adopted symbolic regression benchmarks, including Nguyen Uy et al. (2011), Rational Krawiec & Pawlak (2013), SRBench La Cava et al. (2021), and LLM-SRBench Shojaee et al. (2025a). PhysSymbArena contains 160,000 equations, substantially exceeding the scale of existing benchmark datasets. Unlike benchmarks designed solely for evaluation, PhysSymbArena supports both model training and standardized testing. Moreover, PhysSymbArena supports both conventional symbolic regression algorithms and LLM-based approaches, providing a unified platform for training and evaluating different SR paradigms.

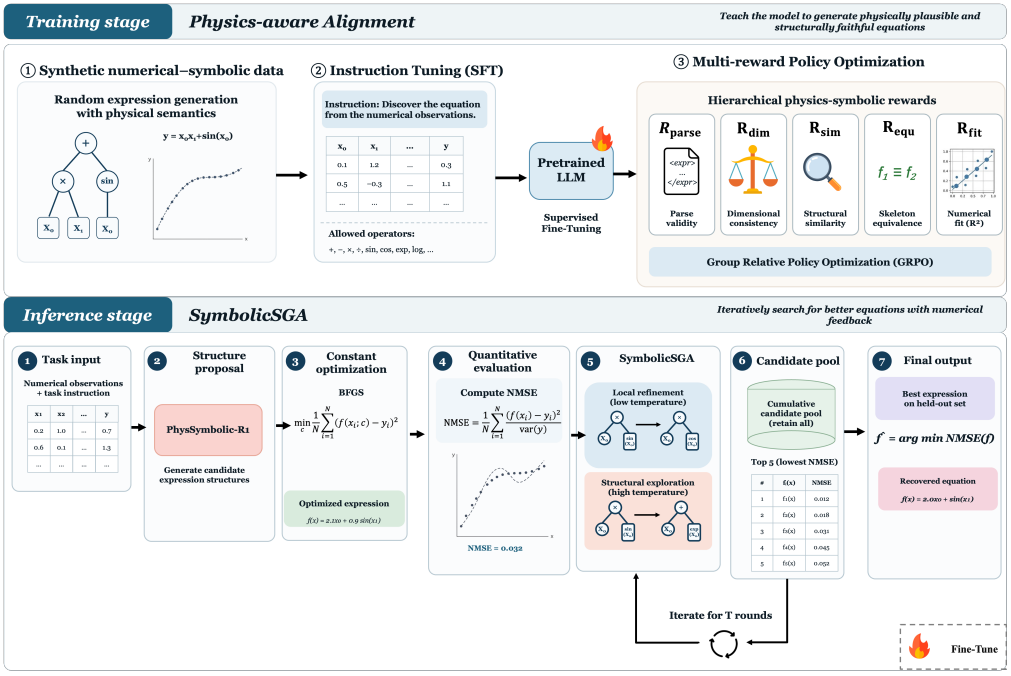


Figure 3: Overview of our proposed method.

# 4 SymbolicLM

## 4.1 Overview

Building on PhysSymbArena, we propose **SymbolicLM**, a framework that explicitly trains general-purpose LLMs as symbolic regressors. The framework consists of three stages. First, instruction tuning establishes the mapping from numerical observations to symbolic expressions. Second, reinforcement fine-tuning further aligns the model with symbolic regression objectives through physics-symbolic rewards, encouraging the generation of syntactically valid, physically consistent, and structurally accurate expressions. Finally, during inference, **SymbolicSGA** further improves equation recovery by iteratively refining candidate expressions based on numerical evaluation and symbolic feedback. The overall framework is illustrated in Figure 3.

## 4.2 Instruction Tuning

Building on PhysSymbArena, we first perform supervised instruction tuning to establish the mapping from numerical observations to symbolic expressions. Each training sample consists of a set of numerical observations $\mathcal{D}_i$, associated physical descriptions $C_i$, and the corresponding symbolic expression $f_i^*$: $\mathcal{D}_i = \{(\mathbf{x}_{i,k}, y_{i,k})\}_{k=1}^{N_i}, y_{i,k} = f_i^*(\mathbf{x}_{i,k})$. For each sample, the input prompt is constructed as: $P_i = [I; V_i]$, where $I$ specifies the symbolic regression instruction and associated physical descriptions, while $V_i$ contains the numerical observations in a structured format:`x_0 = ..., x_1 = ..., f(x)=...`. Following previous symbolic regression approaches (Kamienny et al., 2022), we represent the target expression $f_i^*$ using prefix notation to facilitate sequence generation. For example,$f(x) = 2\sin(x) + 3$ is represented as `[add mul 2 sin x 3]`. Let $z_i^* = (z_{i,1}^*, \ldots, z_{i,T_i}^*)$ denote the target token sequence. We optimize the model using the standard token-level cross-entropy loss:

$$\mathcal{L}_{\text{SFT}} = -\frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} \sum_{t=1}^{T_i} \log \pi_\theta \left( z_{i,t}^* \mid P_i, z_{i,<t}^* \right). \tag{1}$$

### 4.3 PHYSICS-SYMBOLIC REWARDS

While instruction tuning establishes the mapping from numerical observations to symbolic expressions, token-level supervision alone cannot fully optimize symbolic regression objectives. We therefore further apply reinforcement fine-tuning with multiple rewards covering syntactic validity, physical consistency, symbolic structure, and numerical accuracy. Specifically, we introduce five complementary rewards: parse validity, dimensional consistency, structural similarity, symbolic equivalence, and numerical fitting. These rewards guide the model to generate valid, physically consistent, and accurate symbolic expressions.

**Parse-Validity Reward.** We define a parse-validity reward $R_{\text{parse}}$ to evaluate whether generated expressions satisfy the required syntax and can be successfully parsed. Specifically, we introduce a validity checking function $\text{isValid}$ and use SymPy to parse each generated expression. An expression that fails parsing is considered invalid and receives zero reward, while valid expressions receive a reward of one:

$$R_{\text{parse}}(z) = \begin{cases} 1, & \text{isValid}(z) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

**Dimensional-Consistency Reward.** To encourage the model to generate physically consistent expressions, we introduce a dimensional-consistency reward $R_{\text{dim}}$. This reward leverages the physical information provided in PhysSymbArena and evaluates the dimensional consistency of generated expressions according to the validation rules introduced in Section 3. Expressions that violate dimensional constraints are considered invalid and receive a reward of zero, whereas expressions satisfying the constraints receive a reward of one:

$$R_{\text{dim}}(\hat{f}) = \begin{cases} 1, & \text{isDimValid}(\hat{f}, y) = 1, \\ 0, & \text{otherwise,} \end{cases}$$

**Symbolic-Structure Similarity Reward.** To provide dense structural supervision, we compare the expression skeletons of the predicted and ground-truth expressions. Let $\mathcal{S}(\cdot)$ replace numerical constants except exponents with a shared placeholder $c$: $s_{\hat{f}} = \mathcal{S}(\hat{f}), s^* = \mathcal{S}(f^*)$. Each skeleton is characterized by $\Phi(s) = (\mathcal{O}(s), \mathcal{F}(s), \mathcal{V}(s), N_c(s), P(s), C(s))$ , where the components describe operators, nonlinear functions, variables, constants, structural sequence, and expression complexity. The structural similarity reward is

$$R_{\text{sim}}(\hat{f}, f^*) = \sum_{m=1}^{6} w_m \, \text{Sim}_m \left( \Phi_m(s_{\hat{f}}), \Phi_m(s^*) \right), \qquad \sum_{m=1}^{6} w_m = 1.$$

Set-valued features are compared using Jaccard similarity, while count- and sequence-based features use normalized similarity measures.Since each component similarity is normalized to $[0, 1]$. Therefore, even when a candidate expression recovers only a subset of the correct variables, operators, functions, or local structures, it receives a continuous reward proportional to its structural agreement with the ground truth. This feedback encourages the model to progressively approach the target analytical structure.

**Numerical-fit reward.** We use the numerical reward $R_{\text{num}}$ to quantify how well an equation fits the input-output data. This reward is only utilized for equations passed the check of the valid decision function. We use the $R^2$ score to achieve the numerical reward. However, since the raw $R^2$ score is unbounded from below, its direct use can lead to training instability. We therefore apply a truncation strategy, defining the final reward as follows:

$$R_{\text{num}}(\hat{f}) = \begin{cases} \max\left(0, R^2(\hat{f}, f^*)\right), & \text{isValid}(\hat{f}) = 1, \\ 0, & \text{otherwise.} \end{cases}$$

This ensures that only syntactically correct equations are eligible for a numerical reward, while invalid ones are penalized with a score of zero.

**Symbolic-Equivalence Reward.** We define an equivalence reward $R_{\text{equ}}$ to encourage the model to recover expressions with symbolic structures equivalent to the ground-truth expressions. Since exact skeleton recovery is crucial yet challenging in symbolic regression, this reward provides positive feedback when the predicted expression and the ground-truth expression share the same skeleton structure. Specifically, it is defined as:

$$R_{\text{equ}}(\hat{f}_i, f_i) = \begin{cases} 1, & s_{\hat{f}_i} = s_{f_i}, \\ 0, & otherwise. \end{cases}$$

where $\hat{e}_i$ and $e_i$ denote the skeleton structures extracted from the predicted expression $\hat{f}_i$ and the ground-truth expression $f_i$, respectively. Specifically, numerical coefficients are replaced with shared placeholders to eliminate differences caused by constant values and focus on the symbolic structure.

**Reinforcement Fine-tuning.** The overall reward adopts a gated design. Parse validity and dimensional consistency serve as fundamental gates: if the generated expression fails either syntactic parsing or dimensional validation, the overall reward is set to zero. Otherwise, additional rewards for symbolic structure, symbolic equivalence, and numerical fitting are considered. Formally, the reward is defined as:

$$R = R_{\text{parse}} R_{\text{dim}} (w_{\text{sim}} R_{\text{sim}} + w_{\text{equ}} R_{\text{equ}} + w_{\text{num}} R_{\text{num}}).$$

we employ the Group Relative Policy Optimization (GRPO) algorithm (Shao et al., 2024). Specifically, for each input prompt, we sample a group of $G = 8$ candidate expressions from the current policy. Each generated expression is evaluated using the proposed reward function, and the obtained rewards are normalized within the sampled group to compute relative advantages. These advantages are then used to update the policy, encouraging the model to generate symbolic expressions that satisfy syntactic validity, physical consistency, structural correctness, and numerical accuracy.

### 4.4 SymbolicSGA

During inference, we introduce **SymbolicSGA**, a lightweight external scaffold to maximize the model's symbolic regression capabilities. Our approach is inspired by SGA (Ma et al., 2024), a ReAct-based framework for scientific discovery that incorporates two distinct branches: exploitation and exploration. These two branches regulate search behaviors through generation temperatures, using low temperatures for exploitation to achieve local refinement and high temperatures for exploration to foster diversity. SGA then selects the top-5 candidates based on verification results to feed back into the model for the next round of refinement.

We adapt SGA to symbolic regression by modifying both the verification metric and the branching mechanism. For each equation, we evaluate numerical fitness on the data as the verification metric and retain the top-5 candidates for iterative refinement. Rather than adjusting sampling temperatures, we employ task-specific prompts to guide the search. Specifically, the exploitation branch prompts the model to modify variable names or substitute unary and binary terms from the current best equation. Conversely, in the exploration branch, the model is prompted to generate new candidates distinct from the current worst equation while encouraging the reuse of sub-expressions from high-fitness candidates. By leveraging symbolic-aware structural priors, this prompt-guided refinement effectively balances local optimization with global functional exploration, enabling the model to converge to exact analytical equations within few iterations.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** We evaluate SymbolicLM on three symbolic regression benchmarks: PhysSymbArena, SRBench-Feynman (La Cava et al., 2021), and LLM-SRBench-Transform (Shojaee et al., 2025b). PhysSymbArena is our proposed large-scale numerical–symbolic benchmark for training and evaluating LLM-based symbolic regression. SRBench-Feynman contains 119 physics equations from the Feynman equation dataset and is used to evaluate equation discovery on established physical problems. LLM-SRBench-Transform provides transformed symbolic regression tasks for evaluating the generalization ability of LLM-based methods beyond direct equation matching.

Table 2: Results on PhysSymbArena.

| Method | $R^2 \uparrow$ | NMSE $\downarrow$ | $\mathrm{Acc}_\tau \uparrow$ |
|---|---|---|---|
| **Specialized SR Methods** | | | |
| DSR | 0.3239 | 0.6761 | 0.0237 |
| AFP-FE | 0.4078 | 0.5922 | 0.0830 |
| AFP | 0.3931 | 0.6069 | 0.0751 |
| E2E | 0.6297 | 0.3703 | 0.2727 |
| GP-GOMEA | 0.5468 | 0.4532 | 0.2925 |
| GPLearn | 0.2579 | 0.7421 | 0.0632 |
| SNIP | 0.8707 | 0.1293 | 0.3676 |
| TPSR | 0.7871 | 0.2129 | 0.3241 |
| PySR | 0.8652 | 0.1348 | 0.5059 |
| **General-purpose LLMs** | | | |
| Qwen3-8B | 0.0179 | 0.9821 | 0.0119 |
| DeepSeek-V4-flash | 0.4218 | 0.5782 | 0.2946 |
| Gemini-3.8-flash | 0.4538 | 0.5462 | 0.3294 |
| GPT-5.5 | 0.6150 | 0.3889 | 0.3359 |
| **LLM-based SR Methods** | | | |
| SGA (Qwen3-8B) | 0.4710 | 0.5290 | 0.1265 |
| LLM-SR (Qwen3-8B) | 0.6771 | 0.3229 | 0.2727 |
| **Ours** | | | |
| SymbolicLM(SFT) | 0.7581 | 0.2496 | 0.4375 |
| SymbolicLM(SFT+RL) | 0.7626 | 0.2451 | 0.4648 |
| SymbolicLM + SymbolicSGA | **0.9056** | **0.1061** | **0.5508** |

**Baselines.** We compare SymbolicLM with representative methods spanning conventional symbolic regression, neural symbolic regression, and LLM-based approaches. Conventional symbolic regression methods include AFP, AFP-FE, GP-GOMEA, GPLearn, and PySR, which perform explicit search over symbolic expression spaces. Neural symbolic regression methods, including DSR, E2E, SNIP, and TPSR, are included to compare with models specifically trained for numerical-to-symbolic mapping. We further include recent LLM-based symbolic regression methods, such as LLM-SR and SGA. In addition, general-purpose LLMs are evaluated to measure the symbolic regression capability of LLMs without task-specific adaptation. More details are provided in the Appendix.

**Metrics.** Following the evaluation protocols proposed in prior studies (Kamienny et al., 2022; Biggio et al., 2021; La Cava et al., 2021), we adopt two widely used metrics to assess numerical accuracy: the coefficient of determination ($R^2$) and tolerance-based accuracy ($\mathrm{Acc}_\tau$), which reports whether the worst-case relative error is within tolerance $\tau$, given by:

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{\text{test}}} \left(f(x_i) - \hat{f}(x_i)\right)^2}{\sum_{i=1}^{N_{\text{test}}} \left(f(x_i) - \overline{f(x_i)}\right)^2}, \quad \mathrm{Acc}_\tau = 1\left(\max_{1 \le i \le N_{\text{test}}} \left|\frac{\hat{f}(x_i) - f(x_i)}{f(x_i)}\right| \le \tau\right), \tag{2}$$

where $\hat{f}(x_i)$ and $f(x_i)$ are the dependent outputs generated by feeding the same data into the predicted and ground-truth equations.

### 5.2 Results on PhysSymbArena

Table 2 investigates whether symbolic regression can be acquired by general-purpose LLMs through dedicated numerical-symbolic training. Without task-specific adaptation, the pretrained Qwen3-8B model shows limited ability to recover analytical expressions, achieving only $R^2 = 0.0179$ and $\mathrm{Acc}_\tau = 0.0119$. After supervised fine-tuning on PhysSymbArena, SymbolicLM(SFT) substantially improves equation recovery, reaching $R^2 = 0.7581$ and $\mathrm{Acc}_\tau = 0.4375$, outperforming general-purpose LLM baselines and existing LLM-based symbolic regression approaches. These results indicate that numerical-symbolic supervision can effectively transform a general-purpose LLM into a model capable of performing symbolic regression.

Table 3: Results on SRBench-Feynman and LLM-SRBench-Transform.

| Method | SRBench-Feynman | | | LLM-SRBench-Transform | | |
|---|---|---|---|---|---|---|
| | $R^2 \uparrow$ | NMSE $\downarrow$ | $\mathrm{Acc}_\tau \uparrow$ | $R^2 \uparrow$ | NMSE $\downarrow$ | $\mathrm{Acc}_\tau \uparrow$ |
| AFP-FE | 0.6149 | 0.3850 | 0.1344 | 0.4763 | 0.5237 | 0.0270 |
| AFP | 0.6006 | 0.3993 | 0.0924 | 0.4661 | 0.5339 | 0.0090 |
| DSR | 0.6539 | 0.3460 | 0.0672 | 0.5931 | 0.4069 | 0.0000 |
| E2E | 0.9111 | 0.0888 | 0.4117 | 0.7248 | 0.2752 | 0.1622 |
| GP-GOMEA | 0.6018 | 0.3981 | 0.4033 | 0.6308 | 0.3692 | 0.3333 |
| GPLearn | 0.5821 | 0.4178 | 0.2184 | 0.3279 | 0.6720 | 0.0270 |
| SNIP | **0.9683** | **0.0316** | 0.3308 | 0.8925 | 0.1075 | 0.1622 |
| TPSR | 0.9060 | 0.0939 | 0.5042 | 0.8941 | 0.1059 | 0.2432 |
| PySR | 0.9548 | 0.0451 | 0.6386 | 0.9390 | 0.0610 | 0.3423 |
| LLM-SR (Qwen-8B) | - | - | - | 0.7919 | 0.2081 | 0.3333 |
| SGA(Qwen-8B) | - | - | - | 0.3200 | 0.6800 | 0.1441 |
| SymbolicLM + SymbolicSGA | 0.9569 | 0.0430 | **0.6470** | **0.9481** | **0.0519** | **0.4144** |

We further investigate the effect of reinforcement learning alignment. Compared with SymbolicLM(SFT), incorporating GRPO with physic-symbolic rewards further improves structural recovery performance. Specifically, SymbolicLM(SFT+RL) increases exact accuracy from $0.4375$ to $0.4648$, while maintaining comparable numerical fitting performance. This improvement indicates that reward alignment provides additional supervision beyond token-level imitation, encouraging SymbolicLM to generate expressions that better satisfy symbolic structures and physical constraints. Finally, we evaluate the effect of inference-time refinement. Starting from SymbolicLM(SFT+RL), SymbolicSGA iteratively refines candidate equations using numerical feedback. After incorporating SymbolicSGA, SymbolicLM achieves$\mathrm{Acc}_\tau = 0.5508$. Compared with SymbolicLM(SFT), the complete framework improves exact symbolic recovery accuracy by 11.33 %. This demonstrates that quantitative feedback can effectively correct structurally inaccurate hypotheses and further improve equation discovery.

### 5.3 Results on External Symbolic Regression Benchmarks

Table 3 summarizes the performance of SymbolicLM on two external symbolic regression benchmarks, SRBench-Feynman and LLM-SRBench-Transform. These benchmarks evaluate whether the symbolic regression capability learned from PhysSymbArena can generalize beyond the training distribution to independently constructed equation discovery tasks.

On SRBench-Feynman, which contains physical equations from the Feynman dataset, SymbolicLM achieves the highest symbolic recovery accuracy among all compared methods, indicating its ability to recover underlying analytical structures rather than merely optimize numerical fitting. Meanwhile, it maintains competitive numerical accuracy compared with specialized symbolic regression approaches.

On LLM-SRBench-Transform, which evaluates symbolic regression under transformed equation representations, SymbolicLM also achieves strong performance. The results demonstrate that the model can infer underlying functional relationships from numerical observations beyond direct equation matching. Together, these results show that SymbolicLM effectively transfers the symbolic regression capability acquired from PhysSymbArena to diverse equation discovery scenarios.

## 6 Conclusion

In this work, we introduce **PhysSymbArena** and **SymbolicLM** to equip LLMs with symbolic regression capabilities through numerical–symbolic and physical supervision. Experiments across multiple benchmarks demonstrate that SymbolicLM can effectively recover analytical structures from numerical observations. These results indicate that carefully designed numerical, symbolic, and physical supervision can effectively improve the symbolic regression capabilities of LLMs and provide a promising direction toward using language models for scientific equation discovery.

### AI USE STATEMENT

In this work, we used generative AI tools for assistance with writing refinement, language editing, and improving the clarity and organization of the manuscript. We have not used generative AI tools for generating research results, designing experiments, conducting scientific analysis, or making methodological decisions. All AI-assisted outputs were reviewed and verified by the authors. The authors manually checked the correctness of the code, experimental configurations, mathematical formulations, and all reported results. The authors also reviewed the related literature to ensure the originality and proper attribution of the ideas discussed in this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

The authors acknowledge their responsibility to adhere to the ICLR Code of Ethics.

### REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our results, we provide comprehensive details of our methodology, experiments, and implementation. A full description of our model architectures, algorithms, and experimental setup is provided in Appendix. We believe this provides sufficient information for the research community to reproduce and build upon our findings.

## REFERENCES

Luca Biggio, Tommaso Bendinelli, Alexander Neitz, Aurelien Lucchi, and Giambattista Parascandolo. Neural symbolic regression that scales. In *International conference on machine learning*, pp. 936–945. Pmlr, 2021.

Miles Cranmer. Interpretable machine learning for science with pysr and symbolicregression. jl. *arXiv preprint arXiv:2305.01582*, 2023.

Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. *Advances in Neural Information Processing Systems*, 37:44678–44709, 2024.

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

Krzysztof Krawiec and Tomasz Pawlak. Approximating geometric crossover by semantic backpropagation. In *Proceedings of the 15th annual conference on Genetic and evolutionary computation*, pp. 941–948, 2013.

William La Cava, Bogdan Burlacu, Marco Virgolin, Michael Kommenda, Patryk Orzechowski, Fabrício Olivetti de França, Ying Jin, and Jason H Moore. Contemporary symbolic regression methods and their relative performance. *Advances in neural information processing systems*, 2021(DB1):1, 2021.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

Pingchuan Ma, Tsun-Hsuan Wang, Minghao Guo, Zhiqing Sun, Joshua B Tenenbaum, Daniela Rus, Chuang Gan, and Wojciech Matusik. Llm and simulation as bilevel optimizers: A new paradigm to advance physical scientific discovery. *arXiv preprint arXiv:2405.09783*, 2024.

Kazem Meidani, Parshin Shojaee, Chandan Reddy, and Amir Barati Farimani. Snip: Bridging mathematical symbolic and numeric realms with unified pre-training. In *International Conference on Learning Representations*, volume 2024, pp. 42359–42372, 2024.

Matteo Merler, Katsiaryna Haitsiukevich, Nicola Dainese, and Pekka Marttinen. In-context symbolic regression: Leveraging large language models for function discovery. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 427–444, 2024.

Brenden K Petersen, Mikel Landajuela, T Nathan Mundhenk, Claudio P Santiago, Soo K Kim, and Joanne T Kim. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv preprint arXiv:1912.04871*, 2019.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan Reddy. Llm-sr: Scientific equation discovery via programming with large language models. In *International Conference on Learning Representations*, volume 2025, pp. 16054–16085, 2025a.

Parshin Shojaee, Ngoc-Hieu Nguyen, Kazem Meidani, Amir Barati Farimani, Khoa D Doan, and Chandan K Reddy. Llm-srbench: A new benchmark for scientific equation discovery with large language models. *arXiv preprint arXiv:2504.10415*, 2025b.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*, 2022.

Nguyen Quang Uy, Nguyen Xuan Hoai, Michael O'Neill, Robert I McKay, and Edgar Galván-López. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines*, 12(2):91–119, 2011.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.