Decompose, Adapt and Evolve: Towards Efficient Scientific Equation Discovery with Large Language Models

Pouya Behzadifar[1*] **Parshin Shojaee**[2*] **Sanchit Kabra**[2*]
Kazem Meidani[3] **Chandan K. Reddy**[2]

[1]Sharif University of Technology [2]Virginia Tech [3]Capital One

Abstract

Finding mathematical relations underlying natural phenomena and scientific systems has been one of the fundamental tasks in the history of scientific discovery. Recent advancements in evolutionary search with Large Language Models (LLMs), with their embedded scientific knowledge, have shown great promise for this task. However, discovering such mathematical models governing scientific observations still remains significantly challenging, as it requires navigating vast combinatorial hypothesis spaces with an explosion of possible relations. Existing LLM-based approaches overlook the impact of data on the structure of mathematical relations, and treat LLMs as a static hypothesis generator unaware of the observed scientific system. This leads to inefficient exploration of the hypothesis space with over-reliance on LLMs' internal priors. To bridge this gap, we introduce *Decompose, Adapt, and Evolve* (**DecAEvolve**), a framework that leverages granular feedback from symbolic term decomposition and LLM refinement through reinforcement learning (RL) fine-tuning to enhance both robustness and efficiency of evolutionary discovery frameworks. Our experiments across diverse datasets demonstrate that DecAEvolve significantly improves the accuracy of discovered equations and the efficiency of the discovery process compared to the state-of-the-art baseline.

1 Introduction

The emergence of Large Language Models (LLMs) has fundamentally transformed automated problem-solving across diverse domains. Beyond their well-established capabilities in natural language understanding and programming [1, 2], LLMs have recently demonstrated remarkable reasoning abilities that enable them to tackle complex optimization and discovery tasks. Their capacity to leverage embedded domain knowledge, interpolate between them, generate structured hypotheses and engage in iterative refinement, positions LLMs as powerful engines for systematic exploration of complex solution spaces towards discovery goals [3–5]. This potential extends naturally to scientific discovery tasks, where the combination of domain expertise and systematic search/exploration in the hypothesis space can unlock new approaches to longstanding challenges of scientific inquiry [6].

Scientific equation discovery—the process of uncovering compact and interpretable mathematical models that govern natural phenomena—represents one of the fundamental tasks in automated scientific discovery, with applications across many fields of science such as physics, biology, and material science [7]. Traditional approaches in Symbolic Regression (SR) rely on genetic programming and evolutionary strategies [8, 9]; however, these approaches often struggle with scalability limitations and inefficient exploration of the vast combinatorial hypothesis space [10]. More recent advances

*Equal contribution.

Published in the AI for Science workshop (NeurIPS 2025).

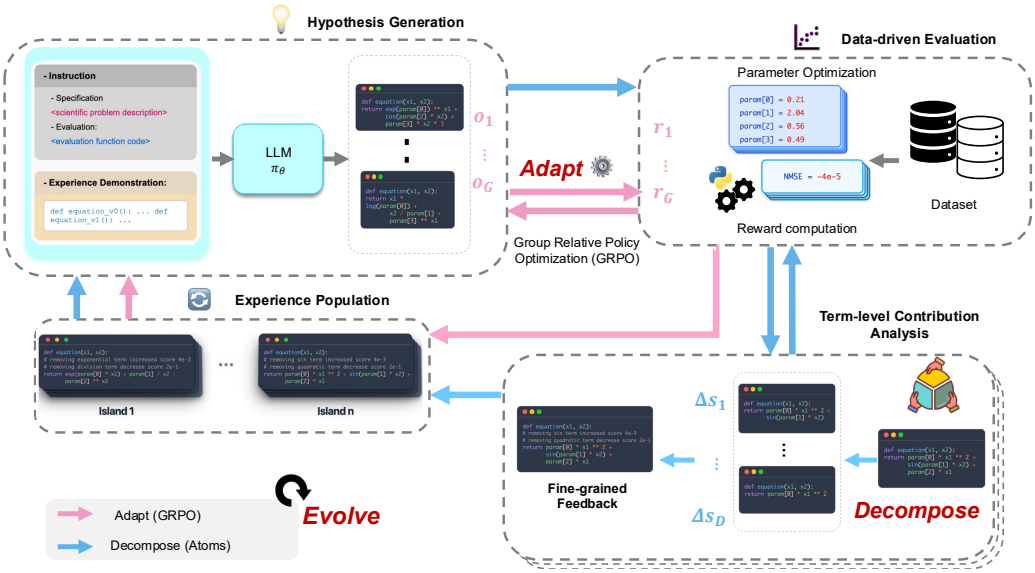

Figure 1: **Overview of the DecAEvolve framework.** The framework integrates *Adaptation* (LLM fine-tuning via reinforcement learning using Group Relative Policy Optimization with data-driven rewards) and *Decomposition* (granular-level feedback through symbolic atomic term analysis) within an *Evolutionary* search process. The adaptation aligns the LLM to the target scientific system beyond its internal priors, while decomposition provides fine-grained guidance for hypothesis refinement. Iterating these three key components enables effective and efficient exploration of the combinatorial hypothesis space in equation discovery.

have introduced neural-guided approaches, where deep learning architectures are trained to generate or refine symbolic expressions [11, 12], and transformer-based methods that are pre-trained with large-scale synthetic data to directly model symbolic sequences as language generation tasks [13–15]. These developments have demonstrated promising capabilities in data-driven learning, yet are limited in balancing learning and search components and in incorporating scientific prior knowledge into the process of discovery.

Several works have recently introduced promising frameworks to integrate LLMs for scientific equation discovery, leveraging their scientific priors and reasoning capabilities to navigate the complex landscape of mathematical expressions more effectively. Notably, LLM-SR [6] combines LLMs' scientific knowledge with multi-island evolutionary search, generating equation hypotheses as Python function skeletons guided by data feedback. LaSR [16] introduces a concept learning approach that extracts abstract textual concepts from successful equation hypotheses, using these concepts to guide both evolutionary search (with PySR [17]) and LLM-based hypothesis generation, and SGA [18] employs a bilevel optimization framework that iteratively combines LLMs for discrete hypothesis generation with physical simulations for continuous parameter optimization. These methods demonstrate this potential by combining LLMs' domain expertise with systematic search strategies, treating equation discovery as a program synthesis problem guided by scientific knowledge [19, 20].

However, current LLM-based discovery methods exhibit fundamental limitations that constrain their effectiveness. First, they treat LLMs as static hypothesis generators, where the model's parameters remain fixed regardless of the problem domain, nuances of the specific observed system or, insights gained during the search process. This prevents LLMs from adapting their generation strategies based on the specific problem, the data, and the domain-specific requirements. Second, existing approaches mainly provide coarse-grained feedback about solution quality, typically limited to scalar reward signals (MSE) from execution of whole hypothesis that indicate which hypotheses perform well respectively, without revealing why specific mathematical components or patterns drive success. This limited feedback mechanism prevents LLMs from understanding the underlying symbolic structure of successful solutions and refining their search strategies accordingly.

To address these limitations, we introduce **DecAEvolve** (Decompose, Adapt, and Evolve), a novel framework that enhances the effectiveness and efficiency of LLM-based equation discovery through several synergistic contributions:

- We develop a systematic methodology for providing LLMs with interpretable directional feedback about which components of their generated hypothesis prove effective. Through structured hypothesis decomposition and evaluations, the contributions of individual terms and their pairwise interactions are quantified and provided as feedback. This enables LLMs to understand not just which hypotheses succeed, but *why* specific mathematical building blocks are effective, transforming blind generation into informed iterative refinement.
- We employ reinforcement learning with Group-Relative Policy Optimization (GRPO) to implicitly encode the data distribution into the model's parameters for better hypothesis generation process. This test-time adaptation/training approach allows the LLM to learn from successful equation discoveries without directly observing raw data, progressively aligning its hypothesis generation with the underlying symbolic relationships through reward-weighted gradient updates.
- We demonstrate that these synergistic contributions dramatically improve search efficiency, requiring significantly fewer iterations to discover accurate symbolic expressions. Our comprehensive evaluation across multiple benchmarks shows superior performance compared to LLM-SR and other baselines in both in-domain and out-of-domain settings, validating the effectiveness of our guided discovery approach.

2 Preliminaries

In equation discovery, the goal is to find a compact mathematical expression $f(x)$ that approximates an unknown target function $f_{\text{real}} : \mathbb{R}^d \to \mathbb{R}$, using a dataset of input-output pairs $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. The objective is to discover functional relationships such that $f(x_i) \approx y_i$ for all i, producing expressions that are both interpretable and capable of generalizing to unseen data. Performance is typically evaluated using fitness to data with metrics such as mean squared error: $\text{MSE}(f, \mathcal{D}) = -\frac{1}{n}\sum_{i=1}^n (f(x_i) - y_i)^2$.

3 Method

We propose **DecAEvolve (Decompose, Adapt, and Evolve)**, shown in Figure 1, a framework that shifts the evolutionary search of equation discovery towards guided discovery, achieved through granular and directional feedback as well as test-time adaptation with reinforcement learning fine-tuning of the backbone LLM to the observed scientific system. We employ LLMs to generate equation program skeletons via their parametric knowledge and adapt the model weights and the equation discovery optimization process with the observations of a scientific system.

The core premise of our approach is that effective symbolic discovery requires the generator to learn not only *what* works, but also *why* it works and *how* to search. We implement this via two main components: (i) fine-grained attribution that quantifies marginal and pairwise term contributions and returns structured feedback to the generator; (ii) test-time adaptation with Group-Relative Policy Optimization (GRPO) [21], which shifts the proposal distribution toward low-error structures. Together, these components define a feedback-driven optimization loop: term-level attributions provide credit assignment, GRPO applies the policy update, The integrated framework guides the generator's search policy, increases expected improvement per iteration, and accelerates convergence to more accurate symbolic models (Figure 1).

3.1 Directional Feedback with Term-Level Contribution

At the core of our framework is an iterative discovery process where the LLM generates candidate symbolic equations guided by structured, interpretable feedback. Unlike prior approaches that rely solely on coarse performance metrics, we introduce a fine-grained contribution analysis mechanism that quantifies the importance of individual terms and their interactions within discovered equations.

Each iteration begins with a carefully structured prompt containing: (i) the discovery task specification, (ii) input variable descriptions, (iii) a curated buffer of high-performing equations from previous iterations annotated with term-level contributions, and (iv) a Python function template. This

programmatic interface—where the LLM completes executable Python program rather than plain text equations—ensures syntactic validity and seamlessly integrates with our optimization pipeline.

Given a generated skeleton $f(x;\theta) = \sum_{i=1}^{n} w_i \cdot \phi_i(x)$, where $\phi_i(x)$ are basis functions proposed by the LLM and w_i are learnable weights, we first optimize the parameters θ using BFGS on the training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$. The fitted model is then evaluated using negative mean squared error as our primary performance metric.

In the contribution analysis, we parse the generated Python equation program skeleton into an Abstract Syntax Tree (AST) and decompose it into constituent terms. Addition and subtraction operations serve as natural term boundaries, while multiplicative structures, powers, and unary function calls (e.g., $\sin(x)$) are preserved as atomic units. This parsing respects the LLM's intended mathematical structure—we inline simple variable assignments and handle unary operations appropriately to maintain semantic integrity.

Following decomposition, we conduct ablated contribution analysis by systematically removing individual terms or term pairs and measuring the resulting performance discrepancy. These ablated scores reveal each component's contribution to the model's predictive power. These contribution annotations are saved and passed as comments in the Python program of hypothesis that gets stored in the experience buffer. When the LLM encounters these annotated equations with directional feedback from term decomposition in subsequent iterations, it can immediately see which terms and interactions drive performance, presented in a natural, readable format within the code context and build upon them in the hypothesis generation.

3.2 Test-Time Adaptation with GRPO

To further enhance the LLM's hypothesis generation capabilities, we incorporate a test-time training or adaptation approach using reinforcement learning fine-tuning with Group-Relative Policy Optimization (GRPO). This allows us to adapt the model to the specific symbolic regression task by learning from the distribution of successful equation discoveries.

After each iteration of hypothesis generation, we collect a dataset of prompts paired with candidate equations and their corresponding rewards. Each equation is evaluated using negative MSE on the training data, which we transform to a bounded reward between 0 and 1 via $r = \exp(-\text{MSE})$ to ensure gradient stability. Failed or invalid completions receive a floor reward of 0.01.

For each prompt x with k candidate equations $\{y_i\}_{i=1}^{k}$, we compute group-relative advantages $A_i = r_i - b(x)$ where $b(x) = \frac{1}{k}\sum_i r_i$ serves as the baseline. This formulation provides variance reduction without requiring a learned value function. The training objective balances reward maximization with a KL regularization term to prevent the model from drifting too far from its initial policy:

$$\mathcal{L}(\theta) = -\mathbb{E}_{x,\{y_i\}}\left[\sum_{i=1}^{k} A_i \log \pi_\theta(y_i|x)\right] + \beta \cdot \text{KL}(\pi_\theta || \pi_{\text{ref}})$$

We implement fine-tuning using LoRA adapters, enabling efficient parameter updates while maintaining the base model as a reference anchor. The KL coefficient β ensures the fine-tuned model retains its general reasoning capabilities while effectively adapting to the observed scientific system with the help of data-driven reward. This GRPO training serves as an implicit mechanism for incorporating the underlying data distribution into the model's hypothesis generation process to go beyond its internal priors. While the LLM never directly observes the raw data, it learns which functional forms and basis functions best capture the data's structure through the reward signal. The model effectively internalizes the dataset's latent patterns by optimizing for equations that minimize prediction error, creating a form of indirect supervision where the data guides the search through reward-weighted gradient updates rather than explicit input-output examples. This creates a virtuous cycle: as the model generates better hypotheses informed by the data distribution, these successful equations become part of the training corpus, reinforcing effective structural patterns and basis functions. The iterative refinement process thus combines the LLM's prior knowledge of mathematical functions with empirical evidence from the specific dataset, yielding a search procedure that becomes progressively more aligned with the true underlying symbolic relationship of the observed data.

4 Experiments

We evaluate DecAEvolve on benchmark datasets from [6], covering physics, biology, and materials science:

Nonlinear Oscillator: Simulates two nonlinear damped oscillators (*Oscillator1*, *Oscillator2*) governed by second-order differential equations in displacement and velocity. Both systems are designed with complex but solvable nonlinear structures that differ from standard oscillator models to challenge LLMs towards discovery through data-driven reasoning.

Bacterial Growth: Models E. coli growth under varying conditions of density, substrate, temperature, and pH. Novel nonlinear terms designed for temperature and pH introduce complexities that require exploration and discovery and are hard to recover from LLM recall.

Stress-Strain Behavior: Captures tensile response of aluminum alloy across temperatures. This dataset uses experimental measurements, providing a more realistic setting with experimental data that challenge LLM-based models beyond synthetic formulations.

We compare DecAEvolve with the state-of-the-art baseline LLM-SR [6] under same configurations: $3{,}000$ LLM calls per problem with sampling temperature $\tau = 0.8$. Equation parameters are optimized with the BFGS solver from SciPy library and a 30s timeout used for the execution of each hypothesis. In the GRPO adaptation phase, we use batch size of 16 per device, gradient accumulation 4, learning rate 10^{-6}, and KL coefficient $\beta = 0.05$. For fine-tuning, we use LoRA adapters with $r = 16$. Decomposition analysis is also limited to 7 terms and their pairwise combinations per equation program hypothesis. We conduct experiments on two open-source Qwen model variants (Qwen2.5-0.5B and Qwen2.5-1.5B) to evaluate scaling behaviors across different model capacities within our computational constraints for fine-tuning.

For the analysis, we use the normalized mean squared error (NMSE) as in [19]: NMSE $= \frac{\sum_{i=1}^{N_{\text{test}}}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N_{\text{test}}}(y_i - \bar{y})^2}$ on both in-domain (ID) and out-of-domain (OOD) test settings, where N_{test} is the test size and $\bar{y}$ the mean target value. NMSE normalizes errors by scale of dataset variance, enabling comparison across datasets.

4.1 Results

Figure 2 presents the discovery trajectories showing best-achieved NMSE scores across search iterations for DecAEvolve and its ablated variants compared to the state-of-the-art LLM-SR baseline. Each curve shows the average across five runs. The results demonstrate that both core components contribute meaningfully to performance: Adaptation (+GRPO) and Decomposition (+Decomp) consistently achieve lower discovery errors and converge faster than the LLM-SR baseline across all benchmark datasets. Notably, the full DecAEvolve framework, which integrates both components, delivers best performance in terms of both final accuracy (lower terminal NMSE) and search efficiency (faster convergence), establishing new state-of-the-art results across all scientific discovery tasks.

To evaluate the generalizability of discovered equations—a fundamental prerequisite for scientific equations and laws—we assess all methods on out-of-distribution (OOD) test data from [6] beyond their training domains. Figure 3 compares in-domain (ID) versus out-of-domain (OOD) NMSE performance across all model variants and benchmark datasets. While all methods exhibit expected performance degradation on OOD data, DecAEvolve consistently achieves the lowest NMSE in both settings. DecAEvolve's strong OOD performance indicates that our framework discovers equations with better inherent generalizability rather than merely fitting to training distributions, a critical distinction for scientific discovery applications where extrapolation beyond observed data is essential.

Lastly, Figure 4 shows consistent reward improvement during GRPO adaptation across both model scales and all datasets, validating our reinforcement learning fine-tuning approach as test-time adaptation for equation discovery. Notably, we observe domain-dependent scaling behaviors: the smaller model (Qwen2.5-0.5B) converges faster on oscillator datasets, while the larger model (Qwen2.5-1.5B) shows better progression on bacterial growth and stress-strain tasks. We think that this pattern correlates with problem complexity for LLM, as evidenced by higher initial rewards on oscillator problems (0.6-0.7) compared to the more challenging biological and mechanical systems (0.2-0.3). Interestingly, the smaller model eventually matches larger model performance even on complex datasets,

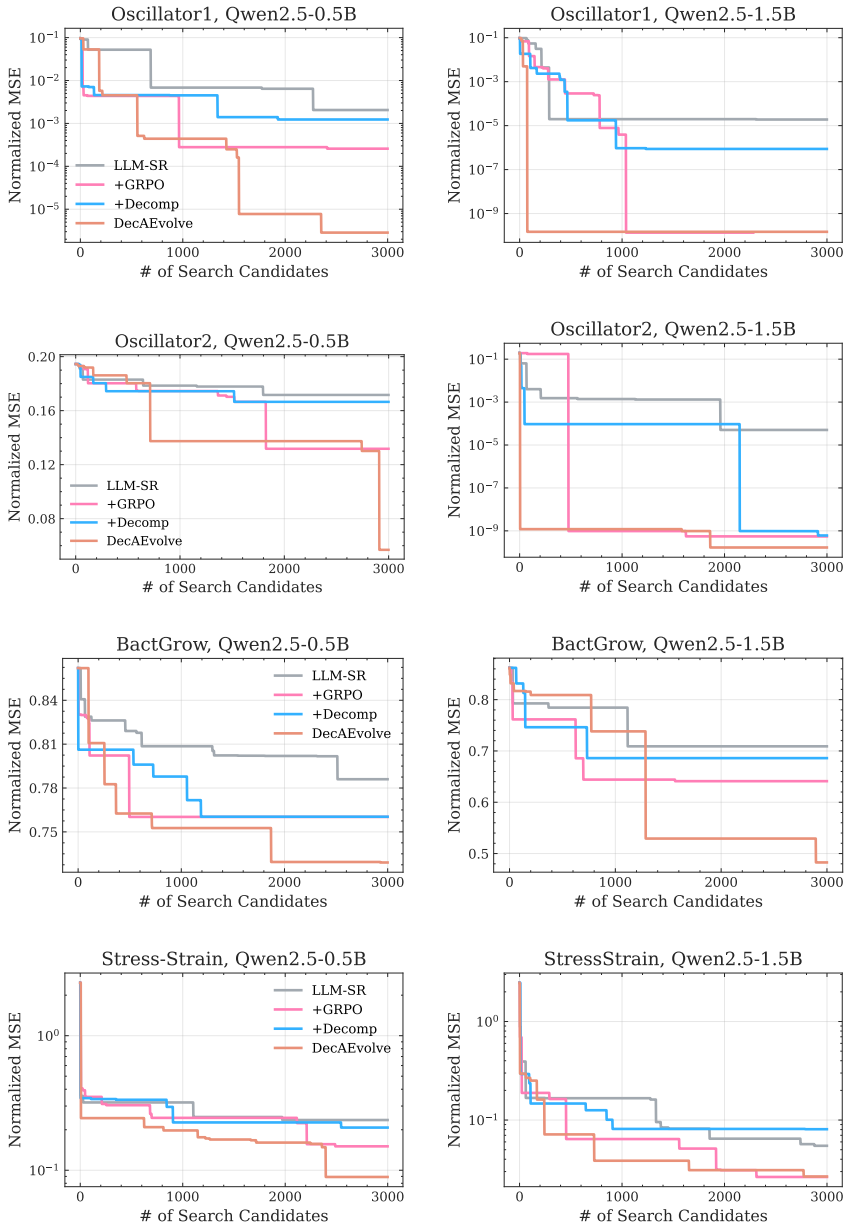

Figure 2: Best-score trajectories of DecAEvolve and its variants against the LLM-SR baseline across benchmark problems. Adaptation (+GRPO) and decomposition (+Decomp) each enhance discovery effectiveness and efficiency, yielding more accurate final equations with fewer search candidates. Their integration in DecAEvolve achieves the best result across all datasets (lower is better). Each curve shows the average across five runs.

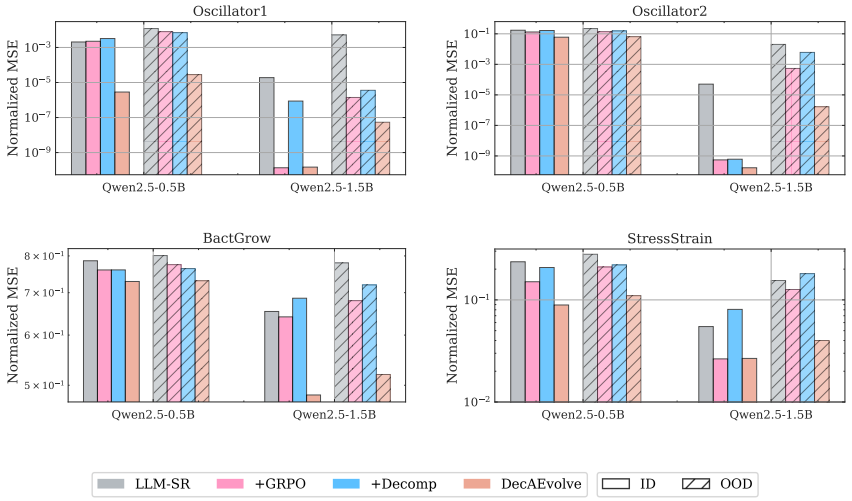

Figure 3: In-domain (ID) and out-of-domain (OOD) performance of DecAEvolve and its variants compared to LLM-SR, reported as normalized MSE across benchmark datasets and LLM backbones (lower is better).

suggesting that targeted adaptation through GRPO can help to effectively bridge the capability gap between model scales for scientific discovery tasks.

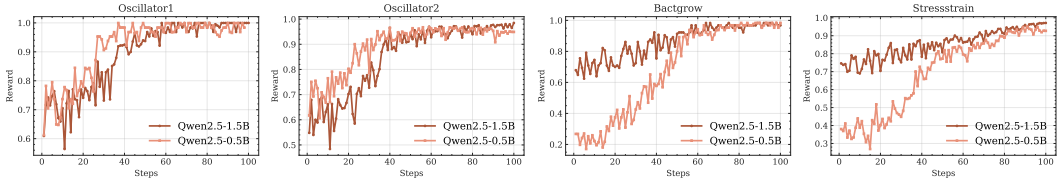

Figure 4: Reward improvement over steps during GRPO adaptation across datasets.

5 Conclusion

We introduce DecAEvolve, a framework that enhances LLM-based equation discovery through granular term-level feedbacks, test-time adaptation via GRPO and, evolutionary search with LLMs. Our approach transforms static hypothesis generation into adaptive learning, enabling LLMs to progressively align with nuances of underlying observed scientific systems through reinforcement learning model adaptation and interpretable feedback mechanisms. Experimental results across diverse benchmark datasets demonstrate that DecAEvolve consistently outperforms state-of-the-art baselines in both discovery accuracy and search efficiency, while maintaining strong out-of-domain generalization. The success of smaller models through targeted test-time adaptation suggests promising directions for democratizing scientific discovery tools without requiring large, resource-intensive models. Future work could extend our simple decomposition mechanisms to more complex structures and explore better optimization strategies for the evolutionary process. The term-level feedback approach developed here may also prove valuable for broader program synthesis tasks requiring iterative refinement in the symbolic space of programs based on component-level understanding.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023.

[3] Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli, and Alhussein Fawzi. Mathematical discoveries from program search with large language models. *Nat.*, 625(7995):468–475, January 2024.

[4] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025.

[5] Anja Surina, Amin Mansouri, Lars Quaedvlieg, Amal Seddas, Maryna Viazovska, Emmanuel Abbe, and Caglar Gulcehre. Algorithm discovery with llms: Evolutionary search meets reinforcement learning. *arXiv preprint arXiv:2504.05108*, 2025.

[6] Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K Reddy. Llm-sr: Scientific equation discovery via programming with large language models, 2025.

[7] Nour Makke and Sanjay Chawla. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1):2, 2024.

[8] John R. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and Computing*, 4(2):87–112, 1994.

[9] William La Cava, Patryk Orzechowski, Bogdan Burlacu, Fabrício Olivetti de França, Marco Virgolin, Ying Jin, Michael Kommenda, and Jason H. Moore. Contemporary symbolic regression methods and their relative performance, 2021.

[10] Marco Virgolin and Solon P. Pissis. Symbolic regression is np-hard, 2022.

[11] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: a physics-inspired method for symbolic regression, 2020.

[12] J.-P. Bruneton. Enhancing symbolic regression with quality-diversity and physics-inspired constraints (qdsr). *arXiv preprint*, 2025.

[13] Author Kamienny et al. End-to-end transformer-based equation generation for symbolic regression. In *NeurIPS*, 2022.

[14] Parshin Shojaee, Kazem Meidani, Amir Barati Farimani, and Chandan Reddy. Transformer-based planning for symbolic regression. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 45907–45919. Curran Associates, Inc., 2023.

[15] Kazem Meidani, Parshin Shojaee, Chandan K. Reddy, and Amir Barati Farimani. SNIP: Bridging mathematical symbolic and numeric realms with unified pre-training. In *The Twelfth International Conference on Learning Representations*, 2024.

[16] Arya Grayeli, Atharva Sehgal, Omar Costilla-Reyes, Miles Cranmer, and Swarat Chaudhuri. Symbolic regression with a learned concept library. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

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

[18] 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. In *Forty-first International Conference on Machine Learning*, 2024.

[19] 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. In *Forty-second International Conference on Machine Learning*, 2025.

[20] Chandan K Reddy and Parshin Shojaee. Towards scientific discovery with generative ai: Progress, opportunities, and challenges. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 28601–28609, 2025.

[21] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.