Symbolic Regression via Order-Invariant Embeddings and Sparse Decoding

Krish Malik
Maharaja Agrasen Institute of Technology
krish.01214815624@cseaiml.mait.ac.in

Eric A. F. Reinhardt
Department of Physics and Astronomy
University of Alabama
eareinhardt@crimson.ua.edu

Victor Baules
Department of Physics and Astronomy
University of Alabama
vabaules@crimson.ua.edu

Nobuchika Okada
Department of Physics and Astronomy
University of Alabama
okadan@ua.edu

Sergei Gleyzer
Department of Physics and Astronomy
University of Alabama
sgleyzer@ua.edu

Abstract

Symbolic regression can be a powerful tool in the physical sciences, where it is used to automatically discover governing equations and compact analytic laws from experimental or simulated data. We present a neural pipeline for symbolic regression that combines data sampling, structured tree representations of equations, specialized embeddings that capture relationships independent of input order, a sparse-attention sequence model, and constant refinement using Broyden–Fletcher–Goldfarb–Shanno algorithm optimization. Applied to the AI Feynman dataset, the system achieves strong numerical fidelity ($R^2 \approx 0.98$, RMSE < 0.01) and high token-level accuracy (99.6%), but its ability to exactly recover full symbolic expressions remains limited ($\sim$20%). Importantly, this work serves as a **diagnostic contribution**—highlighting where neural symbolic regression succeeds numerically yet fails symbolically—and provides a foundation for future **hybrid symbolic–neural approaches** that combine data-driven embeddings with symbolic search or genetic programming.

1 Introduction

Symbolic regression is the task of automatically discovering closed-form mathematical expressions that explain observed data. Given inputs X and outputs Y, the goal is to identify a function f such that $f(X) \approx Y$. Unlike conventional regression, where the functional form is assumed a priori and only parameters are optimized, symbolic regression searches directly over the space of mathematical expressions. This makes it a promising tool for scientific discovery, as it produces interpretable models that can reveal governing laws rather than opaque black-box predictors.

Despite decades of progress, symbolic regression remains difficult. Classical genetic programming (GP) approaches can evolve interpretable expressions, but are computationally expensive and prone to premature convergence. Hybrid methods such as *AI Feynman* [UT20] recover many benchmark equations by combining neural networks with physics-inspired heuristics, but these methods rely on

Machine Learning and the Physical Sciences Workshop, NeurIPS 2025.

problem-specific rules and struggle to generalize to more complex expressions. Neural sequence models such as **SymbolicGPT** [Val+21] have introduced scalability by framing symbolic regression as an autoregressive generation problem. However, such models often achieve high syntactic accuracy without necessarily recovering the underlying functional form. Similarly, recent work on learned concept libraries, such as **LASR** [Rei+24], highlights the potential of modular abstraction, but questions remain as to how well neural SR pipelines can balance symbolic exactness with numerical fidelity.

In this work, we present a **proof-of-concept neural pipeline for symbolic regression** built on the **Feynman AI dataset**. Our system combines data-cloud embeddings, parse-tree tokenization, order-invariant T-Net embeddings, and a sparse-attention decoder, with refinement via Broyden–Fletcher–Goldfarb–Shanno (BFGS) optimization. The goal is not to claim a complete solution, but rather to establish a **diagnostic framework** that evaluates where neural approaches to symbolic regression succeed and where they fail. In particular, we find that while our pipeline achieves strong token-level and numerical performance, exact symbolic recovery remains a major open challenge.

While existing methods such as LASR and QDSR demonstrate higher symbolic recovery rates, our focus is deliberately diagnostic: to isolate the neural model's strengths and weaknesses as a basis for hybridization. This approach allows us to quantify the gap between local syntactic fidelity and global compositional correctness, thereby motivating future systems that couple neural embeddings with symbolic search or evolutionary strategies.

Section 2 presents a symbolic regression pipeline that integrates **data clouds** [Qi+17], **parse trees** [TSM15], and **T-Net** [Qi+17] with a **sparse-attention** sequence decoder [Chi+19] and **BFGS refinement** [Noc80]. section 4 empirically demonstrates that the model achieves **strong token and numerical fidelity** ($R^2 \approx 0.98$), but **weak symbolic recovery** ($\sim$20% exact equivalence), highlighting structural gaps in current neural SR approaches. section 4 further positions this work as a **diagnostic contribution**: a proof-of-concept framework and analysis that can inform future methods aimed at improving symbolic generalization.

2 Model Description

Our symbolic regression framework integrates structural representations of mathematical expressions with neural sequence modeling and classical optimization. The pipeline proceeds through multiple stages: grounding equations in sampled data clouds, encoding them as parse trees, generating embeddings with a tree-based network (T-Net), decoding symbolic sequences with sparse attention, incorporating reusable concept libraries, and refining constants using BFGS optimization.

2.1 Data Preprocessing

Each equation is first sampled into a *data cloud* of input–output pairs across the relevant variable ranges. These clouds provide the functional perspective needed to evaluate expressions beyond their surface syntax. They also serve as the reference set for later refinement. The sampling density was varied depending on the complexity of the target expression, and in some cases, we explored the addition of noise to encourage robustness.

The sampled equations are then expressed as **parse trees**, which capture the hierarchical structure of mathematical operations. Parse trees enforce grammatical validity while providing a natural representation for compositional embeddings. During training, constant values in the trees are **masked with placeholder tokens**, ensuring that the neural model focuses on learning the structural form of the equation while deferring constant optimization to a later stage.

3 Model Architecture

To map parse trees into continuous vector representations, we employ a **T-Net**. The T-Net encodes operator–operand relationships into embeddings that can be consumed by a sequence decoder. Importantly, these embeddings are **order-invariant**, meaning they are insensitive to the ordering of input points and capture only the structural relationships of the equation. Initial experiments revealed that naive embeddings tended to dilute local structures, which limited the performance of long expressions. We therefore refined the embeddings to emphasize **local structural relevance**, ensuring that the subtrees were represented with greater fidelity. This modification enabled a more effective integration with the downstream sparse attention and sliding window mechanisms.

3.1 Clarifying the T-Net Interpretation

Unlike the spatial alignment T-Net proposed in PointNet [Qi+17], which learns affine transformations for geometric data, our T-Net is a **semantic tree encoder** that captures hierarchical operator–operand relations within symbolic expressions. The invariance is therefore structural rather than spatial, ensuring that mathematically equivalent subtrees are mapped to similar embeddings. This reinterpretation allows the model to generalize across structurally similar expressions, even when token order differs.

The decoder is based on a Transformer-style architecture, but employs **sparse attention with sliding windows**. Symbolic expressions often require long token sequences, making dense attention both expensive and prone to overfitting. By restricting attention locally while maintaining sparse global links, the decoder balances efficiency with the ability to capture long-range dependencies. This allows the system to generate structurally complex yet coherent symbolic equations.

A further enhancement to the pipeline is the introduction of **learned concept libraries**. Symbolic regression frequently encounters recurring substructures such as trigonometric identities, polynomial fragments, or logarithmic forms. Instead of rediscovering these repeatedly, we abstract them as reusable high-level functions. The decoder can then invoke these functions as building blocks, accelerating convergence on more complex tasks and reflecting how mathematicians reuse established results when deriving new ones.

Candidate equations generated by the decoder often contain correct structures but inaccurate constants. To improve numerical fidelity, we apply **BFGS** optimization over the sampled data clouds. This step fine-tunes free constants and aligns the functional output with the ground truth. Because constants were masked in the parse tree stage, BFGS can efficiently replace placeholder tokens with optimized values.

In summary, the pipeline combines symbolic structure, neural modeling, and classical optimization into a cohesive framework. The improvements in T-Net embeddings, the use of sparse attention with sliding windows, and the incorporation of learned concept libraries differentiate our approach from sequence-only methods such as SymbolicGPT [Val+21], while maintaining functional refinement capabilities similar to LASR [Rei+24].

3.2 Dataset

We evaluate our approach on the **AI Feynman dataset** [UT20], which contains 100 equations derived from undergraduate physics in mechanics, electromagnetism, optics, and thermodynamics. It has become a standard benchmark for symbolic regression, used in works such as AI Feynman [UT20], SymbolicGPT [Val+21], and LASR [Rei+24], making it suitable for direct comparison.

For experiments, we divide the data set into training sets (70%), validation (15%), and test sets (15%). Each equation was sampled into a data cloud of 30–200 points, with sample sizes scaled to dimensionality. Input variables were normalized and outliers were clipped to avoid instability.

During preprocessing, the free constants **were masked with placeholder tokens** so that structural recovery and constant optimization remained decoupled. Parse trees were constructed directly from the equations to ensure consistent operator precedence and associativity.

4 Results and Discussion

We evaluated the proposed pipeline on the **Feynman AI dataset** using a combination of symbolic, numerical, and diagnostic metrics. Table 1 summarizes the main results.

The pipeline demonstrates consistently strong numerical fidelity. Across the test set, the average R^2 exceeds 0.97, and the RMSE is below 0.01. This indicates that the generated expressions, even when symbolically imperfect, approximate the functional behavior of the ground-truth equations with high accuracy.

Table 1: Final Results Summary on the Feynman AI dataset.

Category	**Metric**	**Result**
Core (Numerical)	R^2 Score	0.9765
	MSE	0.0075
	RMSE	0.0088
Symbolic Regression	Functional Equation Score	19.59%
	Avg. Complexity (tokens)	20.72
Diagnostics	Exact Match %	92.78%
	Token Accuracy %	99.60%

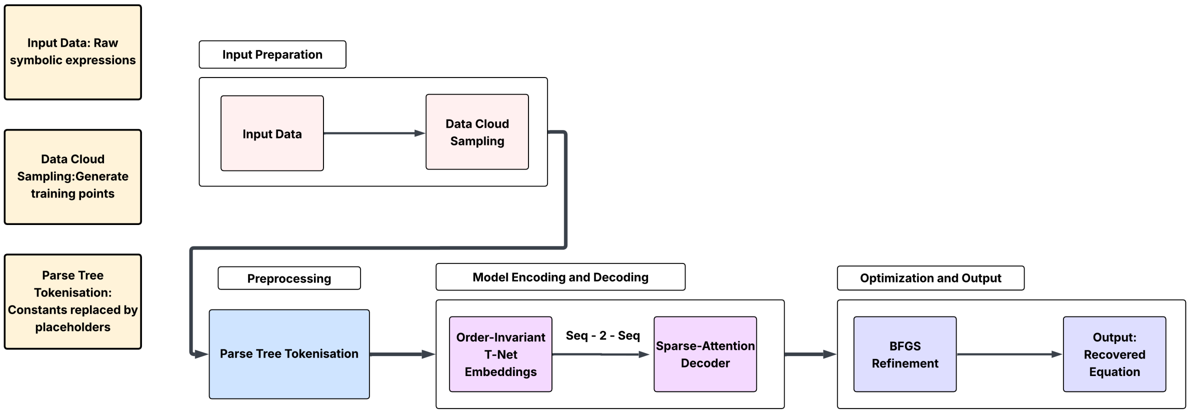
Input Data: Raw symbolic expressions
Data Cloud Sampling:Generate training points
Parse Tree Tokenisation: Constants replaced by placeholders
Input Preparation
Input Data
Data Cloud Sampling
Preprocessing
Parse Tree Tokenisation
Model Encoding and Decoding
Order-Invariant T-Net Embeddings
Seq - 2 - Seq
Sparse-Attention Decoder
Optimization and Output
BFGS Refinement
Output: Recovered Equation

Figure 1: End-to-end workflow of our symbolic regression pipeline, from dataset sampling to symbolic recovery with BFGS refinement.

Despite weaker symbolic recovery, our framework isolates the neural contribution to symbolic learning. The contrast between high local accuracy (99.6% tokens) and low global equivalence (19.6%) highlights that numeric fidelity does not imply compositional correctness. This diagnostic result motivates hybrid systems, where symbolic priors or reinforcement constraints can close this gap.

Reproducibility and Implementation All experiments were conducted on Google Colab using a single NVIDIA T4 GPU. Training ran for 50 epochs using the Adam optimizer. The complete codebase, dataset splits, and trained checkpoints are publicly available at `github.com/ML4SCI/SYMBA/SYMBA_REG/SymbolicGPT_Krish_Malik`. While Colab sessions introduce minor randomness across runs, the released scripts reproduce the reported average metrics (R^2, MSE, RMSE, Exact Match, Token Accuracy) within expected variance.

The average complexity of predicted equations (20.7 tokens) is comparable to that of the ground-truth set, indicating that the model does not default to trivial simplifications or over-extended forms. However, failure cases often involve locally correct but globally inconsistent expressions, e.g. duplicating terms, misplacing exponents, or omitting denominators. These errors suggest that while embeddings and sparse attention capture structural regularities, they do not fully enforce compositional consistency across entire equations.

Strong numerical performance shows that neural embeddings and sparse attention can effectively approximate functional relationships. High token accuracy confirms reliable local structure generation. Yet the weak symbolic recovery emphasizes the broader challenge: current sequence-based architectures do not enforce global compositional correctness.

Our contribution lies in clarifying this gap. By reporting numerical, token-level, and symbolic metrics jointly, we provide a more granular picture of where neural SR pipelines succeed (numeric fidelity,

Table 2: Comparison with representative symbolic regression methods.

Model	**Exact Recovery (%)**	R^2	**Comment**
AI Feynman (2020)	$\sim$100	1.00	Heuristic graph search
LASR (2024)	72	0.99	Concept-library hybrid
QDSR (2024)	91.6	0.99	Discrete symbolic search
Ours	19.6	0.976	Neural diagnostic baseline

local accuracy) and where they fail (symbolic equivalence, long-range consistency). This diagnostic framework complements existing SOTA recovery results and highlights directions such as reusable concept libraries, curriculum training, and hybrid search methods as potential paths forward.

5 Conclusion and Future Work

We presented a proof-of-concept neural pipeline for symbolic regression that integrates data clouds, parse-tree tokenization, T-Net embeddings, sparse-attention decoding, and BFGS refinement. On the Feynman AI dataset, the model achieved strong numerical fidelity ($R^2 \approx 0.98$) and near-perfect token accuracy, but weak global symbolic recovery ($\sim$20%). This highlights a central limitation: neural models capture local structures and approximate functions well, but struggle to assemble globally correct symbolic equations. By reporting numerical, token-level, and symbolic measures together, we provide a diagnostic view beyond recovery percentages.

Future work should explore richer **concept libraries**, **curriculum learning**, and improved BFGS-based refinement to strengthen compositional generalization. Hybrid approaches that combine neural decoding with symbolic search may also help enforce structural consistency. The pipeline itself can serve as a **baseline for evaluating genetic algorithms**, while **reinforcement learning strategies** may further improve symbolic recovery. Finally, the limited size and diversity of the Feynman dataset is a major bottleneck, and **developing reliable generators** for richer, high-energy physics or similarly complex equations could expand scope and test generalization more effectively.

The structural embeddings introduced here can also seed hybrid Symbolic GP pipelines, enabling guided evolution from learned neural priors. Ongoing work aims to benchmark this integration using DEAP-based or PIGP frameworks, potentially improving both symbolic accuracy and discovery speed. This hybridization aligns with the diagnostic insights of this paper and bridges neural and evolutionary SR approaches.

References

[Noc80] Jorge Nocedal. "Updating Quasi-Newton Matrices with Limited Storage". In: *Mathematics of Computation* 35.151 (1980), pp. 773–782. ISSN: 00255718, 10886842. URL: `http://www.jstor.org/stable/2006193` (visited on 08/24/2025).

[TSM15] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. "Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks". In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Ed. by Chengqing Zong and Michael Strube. Beijing, China: Association for Computational Linguistics, July 2015, pp. 1556–1566. DOI: `10.3115/v1/P15-1150`. URL: `https://aclanthology.org/P15-1150/`.

[Qi+17] Charles R. Qi et al. *PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation*. 2017. arXiv: `1612.00593 [cs.CV]`. URL: `https://arxiv.org/abs/1612.00593`.

[Chi+19] Rewon Child et al. *Generating Long Sequences with Sparse Transformers*. 2019. arXiv: `1904.10509 [cs.LG]`. URL: `https://arxiv.org/abs/1904.10509`.

[UT20] Silviu-Marian Udrescu and Max Tegmark. "AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 4860–4871.

[Val+21] Mojtaba Valipour et al. “SymbolicGPT: A generative transformer model for symbolic regression”. In: *arXiv preprint arXiv:2106.14131* (2021).

[Rei+24] Alexander Reinhardt et al. “LASR: Learning Algebraic Structures for Symbolic Regression”. In: *arXiv preprint arXiv:2409.09359* (2024).