Evolutionary and Transformer based methods for Symbolic Regression

Samyak Jha
IIT Dhanbad, India
samyakjha71@gmail.com

Sergei Gleyzer
Department of Physics & Astronomy, University of Alabama,
Tuscaloosa, AL 35401, USA
sgleyzer@ua.edu

Eric A. F. Reinhardt
Department of Physics & Astronomy, University of Alabama,
Tuscaloosa, AL 35401, USA
mtoomey@mit.edu

Victor Baules
Department of Physics & Astronomy, University of Alabama,
Tuscaloosa, AL 35401, USA
vabaules@crimson.ua.edu

Nobuchika Okada
Department of Physics & Astronomy, University of Alabama,
Tuscaloosa, AL 35401, USA
okadan@ua.edu

Francois Charton
Meta AI Research
fcharton@meta.com

Abstract

Symbolic regression aims to uncover mathematical expressions that fit data, traditionally using evolutionary algorithms like genetic programming. However, these methods often struggle with noise, impacting robustness. We propose two hybrid approaches integrating genetic programming with transformer models to enhance performance. The first method, partially initialized genetic programming (PIGP), partially initializes solutions from a pre-trained transformer. The second approach employs symbolic Direct Preference Optimization (DPO), where a pre-trained transformer generates candidate solutions via beam search, which are then refined by genetic programming. Preference pairs from the top solutions fine-tune the transformer to improve R^2 scores. Our experiments show that these transformer-based methods significantly enhance robustness in noise and perform comparably or better than traditional genetic programming methods, such as e-lexicase selection, in noise-free conditions. These findings highlight the potential of transformer-enhanced symbolic regression for improved model robustness and accuracy.

1 Introduction

Symbolic regression aims to discover analytical expressions that describe relationships in data. Traditional approaches, such as genetic programming (GP)[1], are popular for this task due to their ability to evolve complex expressions without requiring explicit function forms. However, GP methods are sensitive to noise and can struggle with generalization across different datasets. Recently,

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

transformer-based models[2] have shown potential in symbolic domains[3, 4, 5, 6], enabling the combination of evolutionary methods with data-driven techniques to enhance symbolic regression.

In this work, we propose two approaches that integrate transformer models with genetic programming (GP) to tackle the challenges of noise and robustness in symbolic regression. We used a transformer architecture inspired by [4], using the P1000 method [7] to tokenize floating-point numbers. The equations are represented in prefix notation, and the transformer is pre-trained on batches of 1000 expressions. Our first approach, Partially Initialized Genetic Programming (PIGP), uses the pre-trained transformer to partially initialize the GP population. Solutions are generated by beam search, with the data set divided into chunks following insights from e-lexicase selection [8]. Post-processing ensures that the expressions adhere to symbolic grammar rules.The second approach, Symbolic Direct Preference Optimization (SymbolicDPO), iteratively refines the transformer-generated equations through GP and preference-based learning. Preference pairs are generated from top performing samples, and DPO [9] fine-tunes the transformer iteratively. Building upon the method in [10], which explores reinforcement learning combined with other techniques, it is well-established that reinforcement learning often requires careful parameter tuning to achieve optimal results in symbolic regression tasks, as noted in [11, 12, 13] and further supported by [14]. Our experiments demonstrate that these transformer-guided approaches are robust to noise while performing similarly to traditional GP methods in noise-free scenarios.

2 Background and Related Work

Symbolic regression has traditionally been dominated by GP, as introduced by Koza [15], but GP-based approaches often struggle with noisy datasets. The sensitivity of GP to noise can lead to overfitting, reducing the model's interpretability [16]. To mitigate this, e-lexicase selection has emerged as a noise-resistant selection method by evaluating solutions in diverse data subsets, improving the resilience of GP to noise. Transformer models have recently been applied to symbolic regression tasks.[4] showed that transformers can effectively generate symbolic expressions, outperforming traditional GP in noiseless datasets. The P1000 tokenization method [7], which divides floating-point numbers into signs, mantissa, and exponents, enhances the transformer's ability to represent numerical data. In our work, we adopt this method for numerical tokenization and prefix notation for equation representation, allowing transformers to better capture mathematical structures.Direct Preference Optimization (DPO) [9] has proven effective for improving neural-symbolic models by refining models using preference pairs. Furthermore, Mundhenk et al. [10] demonstrated that neural networks can guide the seeding of the GP population, providing a foundation for the integration of evolutionary methods with deep learning. Building on these concepts, we combine transformer-guided initialization with genetic programming , iterating between the two for improved performance and robustness, especially in noisy conditions.

3 Methodology

3.1 Transformer Pretraining

We pretrain a transformer model on symbolic equations following Kamienny et al. [4], using P1000 tokenization [7] to handle floating point numbers and prefix notation for clarity. The data set is segmented into chunks of 1000 expressions, inspired by the selection of e-lexicase [8], to reveal diverse patterns and build a robust foundation for the generation of transformer-based symbolic expressions.

3.2 Partially Initiallized Genetic Programming (PIGP)

PIGP integrates transformer-generated expressions into genetic programming (GP) to achieve a balance between exploitation and exploration, where exploitation leverages transformer-generated solutions to guide the search toward promising regions, and exploration utilizes GP to investigate diverse possibilities within the search space. We use beam search with the pre-trained transformer to generate candidate expressions and partially initialize the GP population, maintaining diversity with random initialization. We apply grammatical constraints to ensure valid expressions, enforcing rules such as having binary operators followed by two operands. This post-processing step helps to maintain the quality of the solution and compliance with the symbolic grammar.

3.3 Symbolic Direct Preference Optimization (SymbolicDPO)

SymbolicDPO refines transformer-generated expressions with GP, using beam search to propose candidates and continuously integrating these into GP for exploration. The best performing genetic programming (GP) solutions are used to construct preference pairs, which are then used to fine-tune the transformer using Direct Preference Optimization (DPO) [9]. When generating these preference pairs, we adopt the strategy of selecting the top m_1 solutions based on lower R^2 scores, designating them as preferred over the subsequent m_2 solutions ranked after m_1. Typically, m_2 is chosen to be smaller than m_1 to mitigate the risk that suboptimal expressions are mistakenly preferred. This process adjusts the transformer based on relative rankings, improving its expression generation over iterations. DPO is applied iteratively with an increasing beta parameter throughout each cycle, increasing the transformer sensitivity to performance differences and improving expression quality over time. Both PIGP and SymbolicDPO enhance robustness in the presence of noise and maintain search space diversity. PIGP's random initialization and SymbolicDPO's iterative refinement improve performance compared to traditional GP, particularly in noisy conditions, while achieving competitive results in noise-free settings.

Test Eqns	**Actual Equation**	**GP**	**PIGP**	**SymbolicDPO**
I.6.2a	exp(-theta^2/2)/sqrt(2*pi)	0.999995	**0.999999**	0.999993
I.12.5	q^2*Ef	**1.0**	**1.0**	**1.0**
I.18.14	m*r*v*sin(theta)	0.999984	0.999982	**1.0**
I.39.1	3/2*pr*V	0.999999	0.999999	**1.0**
I.43.16	mu_drift*q*Volt/d	0.999988	0.999997	**1.0**
I.43.31	mob*kb*T	0.999992	0.999995	**1.0**
II.4.23	q/(4*pi*epsilon*r)	**0.999955**	0.999967	0.999954
II.21.32	q/(4*pi*epsilon*r*(1-v/c))	**0.999983**	0.999949	0.999945
II.35.21	n_rho*mom*tanh(mom*B/(kb*T))	**0.999994**	0.999986	0.999986
II.38.3	Y*A*x/d	0.999997	0.999969	**1.0**

Table 1: Table 1: Comparison of GP, PIGP, and SymbolicDPO performance (without noise).

Test Eqns	**GP (Noise = 0.1*std)**	**PIGP (Noise = 0.1*std)**	**SymbolicDPO (Noise = 0.1*std)**
I.6.2a	**0.999998**	0.999997	0.999994
I.12.5	**1.0**	**1.0**	**1.0**
I.18.14	0.999966	0.999993	**1.0**
I.39.1	**0.999998**	0.999997	0.999996
I.43.16	0.999996	0.999989	**1.0**
I.43.31	0.999992	**1.0**	**1.0**
II.4.23	0.999923	0.999946	**0.999954**
II.21.32	0.999982	**0.999984**	0.999955
II.35.21	0.999974	0.999986	**0.999990**
II.38.3	0.999990	0.999987	**1.0**

Table 2: Table 2: Performance comparison under noise (0.1*std) across GP, PIGP, and SymbolicDPO.

4 Experiments and Dataset

4.1 Dataset

The AI Feynman dataset [17] is used for evaluating the proposed methods. This dataset contains 100 equations representing various physical phenomena. We use 80 equations for pretraining the transformer model, 10 for validation, and 10 for testing. The dataset provides a challenging and diverse set of equations, ideal for testing symbolic regression techniques.During pre-training, equations are tokenized using the P1000 method [7], which breaks down floating-point numbers into sign, mantissa, and exponent components. Each equation is divided into chunks of 1000 points for tokenization and representation in prefix notation. This ensures consistency and improves the efficiency of training.

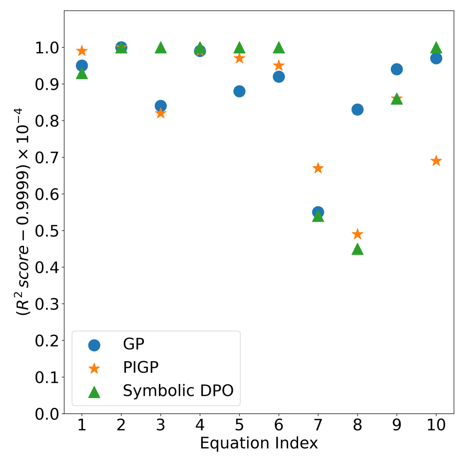

Figure 1: Comparison between the three approaches in terms of R^2score (scaled) without noise.

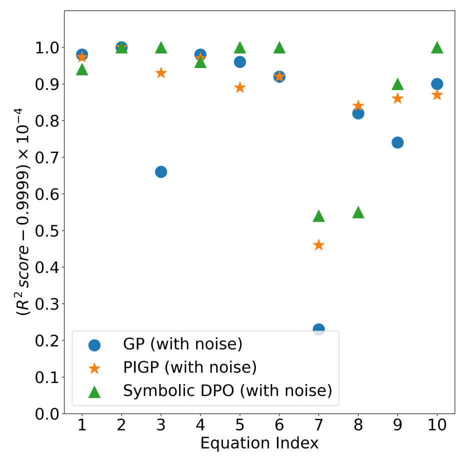

Figure 2: Comparison between the three approaches in terms of R^2 score (scaled) with target noise = 0.1*std of target.

4.2 Experimental Setup

We conduct two main experiments to evaluate our proposed methods, Partially Initialized Genetic Programming (PIGP) and Symbolic Direct Preference Optimization (SymbolicDPO). Both methods are evaluated on their ability to recover original equations, with additional experiments on noisy data to assess robustness. For all experiments, we use the R^2 score to evaluate the accuracy of the generated symbolic expressions. In the noise-robustness experiments, Gaussian noise with a standard deviation of 0.1 is added to the target variables, allowing us to compare how the methods handle noisy data against traditional genetic programming (GP) with e-lexicase selection [8].

5 Results and Discussion

5.1 Results

The results of the experiments are presented in Table 1 (noise-free) and Table 2 (noisy data).

In noise-free conditions, both PIGP and SymbolicDPO perform similarly to traditional GP, with R^2 scores close to 1.0, indicating accurate symbolic expression generation and recovery of original equations. In the presence of Gaussian noise, transformer-based methods outperform traditional GP, with PIGP and SymbolicDPO maintaining higher accuracy. SymbolicDPO, in particular, exhibits the greatest resilience to noise due to its iterative fine-tuning and preference-guided optimization, demonstrating superior robustness compared to traditional GP based methods.

5.2 Discussion

The experiments underscore the key advantages of integrating transformers with genetic programming for symbolic regression. In noise-free conditions, both PIGP and SymbolicDPO perform comparable to traditional GP, maintaining accuracy without degradation. However, in the presence of noise, the transformer-guided approaches demonstrate notable improvements in robustness. The iterative refinement process in SymbolicDPO, driven by Direct Preference Optimization (DPO), enables it to adapt more effectively to noisy data, progressively enhancing predictions with each iteration. PIGP's partial initialization strikes a balance between the transformer's guidance and GP's exploratory nature, resulting in better generalization across diverse datasets. Although the R^2 scores are uniformly high, it is important to note that even small variations in these scores can correspond to significant differences in the generated equations. In fact, if we were to examine the mean squared error (MSE), we would observe a similar trend, as MSE is inversely proportional to R^2. However, the large variation in MSE values across different equations makes direct comparison difficult.Overall, these findings suggest that transformer-enhanced methods are not only robust but also highly adaptable, offering a promising approach for symbolic regression, particularly in challenging real-world scenarios where noise is prevalent.

6 Acknowledgments

This work was supported by the Google Summer of Code (GSoC) 2024 program through the Machine Learning for Science (ML4SCI) organization and the U.S. Department of Energy (DOE) under Grant No. DE-SC0023713. The research utilized resources from the National Energy Research Scientific Computing Center (NERSC), a DOE Office of Science User Facility at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231, through NERSC Award HEP-ERCAP0025759.

References

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

[2] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[3] Abdulhakim Alnuqaydan, Sergei Gleyzer, Harrison B Prosper, Eric AF Reinhardt, Neeraj Anand, and Francois Charton. Symbolic machine learning for high energy physics calculations. *ML4PS workshop, Neurips*, 2024.

[4] 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.

[5] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. *arXiv preprint arXiv:1912.01412*, 2019.

[6] Felipe Leno da Silva, Andre Goncalves, Sam Nguyen, Denis Vashchenko, Ruben Glatt, Thomas Desautels, Mikel Landajuela, Brenden Petersen, and Daniel Faissol. Leveraging language models to efficiently learn symbolic optimization solutions. In *Adaptive and Learning Agents (ALA) Workshop at AAMAS*, 2022.

[7] François Charton. Linear algebra with transformers. *arXiv preprint arXiv:2112.01898*, 2021.

[8] William La Cava, Lee Spector, and Kourosh Danai. Epsilon-lexicase selection for regression. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 741–748, 2016.

[9] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[10] T Nathan Mundhenk, Mikel Landajuela, Ruben Glatt, Claudio P Santiago, Daniel M Faissol, and Brenden K Petersen. Symbolic regression via neural-guided genetic programming population seeding. *arXiv preprint arXiv:2111.00053*, 2021.

[11] 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.

[12] Hengzhe Zhang and Aimin Zhou. Rl-gep: symbolic regression via gene expression programming and reinforcement learning. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021.

[13] Mikel Landajuela, Brenden K Petersen, Sookyung Kim, Claudio P Santiago, Ruben Glatt, Nathan Mundhenk, Jacob F Pettit, and Daniel Faissol. Discovering symbolic policies with deep reinforcement learning. In *International Conference on Machine Learning*, pages 5979–5989. PMLR, 2021.

[14] Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[15] John R Koza. Genetic programming as a means for programming computers by natural selection. *Statistics and computing*, 4:87–112, 1994.

[16] Ekaterina J Vladislavleva, Guido F Smits, and Dick Den Hertog. Order of nonlinearity as a complexity measure for models generated by symbolic regression via pareto genetic programming. *IEEE Transactions on Evolutionary Computation*, 13(2):333–349, 2008.

[17] Silviu-Marian Udrescu and Max Tegmark. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.