Modeling High-Speed SerDes Links with Symbolic Regression

Soumyadeep Chatterjee
ECE Department
North Carolina State University
Raleigh, NC, United States
schatte5@ncsu.edu

Priyank Kashyap
Storage Division
Hewlett Packard Enterprise
Durham, NC, United States
priyank.kashyap@hpe.com

Nirjhor Rouf
ECE Department
North Carolina State University
Raleigh, NC, United States
nrouf2@ncsu.edu

Chris Cheng
Storage Division
Hewlett Packard Enterprise
Milpitas, CA, United States
chris.cheng@hpe.com

Paul D. Franzon
ECE Department
North Carolina State University
Raleigh, NC, United States
paulf@ncsu.edu

***Abstract*—Conventional machine learning approaches that model a serializer-deserializer (SerDes) are black-box and offer no insight into what the model actually learns. This paper presents an alternative approach using a symbolic regression (SR)–based framework for fast, interpretable modeling of Electronic Design Automation (EDA) systems, achieving highly accurate performance using minimal high-fidelity simulations. Further, the proposed approach discovers closed-form symbolic expressions that explicitly capture the relationships between design parameters and system-level performance metrics. The paper evaluates SR on an open-ended SerDes modeling task in which continuous-time linear equalizer (CTLE) and decision feedback equalizer (DFE) settings map to eye performance metrics extracted from circuit-level simulations. The results show that SR fits the data with $R^2 > 0.99$ relative to reference simulations, while requiring a fraction of the time. The paper then explores trade-offs between predictive accuracy and model complexity.**

***Index Terms*—symbolic regression, machine learning, SerDes modeling, high-speed links, equalizer**

I. Introduction and Motivation

The increasing complexity and performance requirements of modern electronic systems, coupled with aggressive power and reliability constraints, have placed unprecedented demands on advancements in Electronic Design Automation (EDA) tools. Workflows must contend with highly nonlinear, high-dimensional design spaces and increasingly expensive simulation and verification loops. Consequently, the availability of fast, accurate, and generalizable modeling techniques is pivotal to the overall success of integrated circuit and system design [1].

Many critical EDA tasks require repeated evaluation of complex physical models, including signal integrity (SI) and channel modeling for high-speed links, SerDes equalization design (e.g., continuous-time linear equalizer (CTLE) and decision feedback equalizer (DFE) tuning), timing and power analysis, and parasitic extraction and routing optimization [2]. These problems tend to involve limited data, high simulation costs, and the need to extrapolate reliably across a wide range of operating conditions. While data-driven machine learning (ML) models, particularly deep neural networks (DNNs), have shown promise in accelerating specific tasks, their deployment is frequently hindered by large data requirements, poor extrapolation behavior, limited physical interpretability, and the need for costly retraining when design parameters or operating conditions change [3].

Symbolic regression (SR) offers a compelling alternative to purely black-box ML approaches by directly discovering closed-form mathematical expressions that describe system behavior from data. Unlike DNNs, expressions derived via SR are inherently interpretable, computationally lightweight at inference time, and amenable to physical inspection and validation by domain experts [4]. SR has been successfully applied in materials science and physics-driven modeling to derive interpretable expressions that accurately capture complex system behavior while maintaining physical consistency [5], with hybrid approaches combining sparse optimization or inductive biases from machine learning to further improved robustness and scalability in high-dimensional scientific problems [6]. These properties are desirable in the EDA context, where understanding the underlying physical relationships and ensuring robustness across design corners are often as important as raw predictive accuracy.

In this work, we demonstrate the suitability of SR for SI analysis in complex systems by deriving human-interpretable equations for a 50 Gb/s high-speed SerDes link, achieving over 99% fidelity with respect to reference simulations. We show that such accuracy can be attained using relatively modest data requirements compared to conventional ML techniques, while simultaneously producing compact symbolic models that generalize well across the device's operating conditions. Based on these results, SR constitutes a practical and powerful modeling paradigm for accelerating EDA workflows and enabling accurate and interpretable system-level analysis.

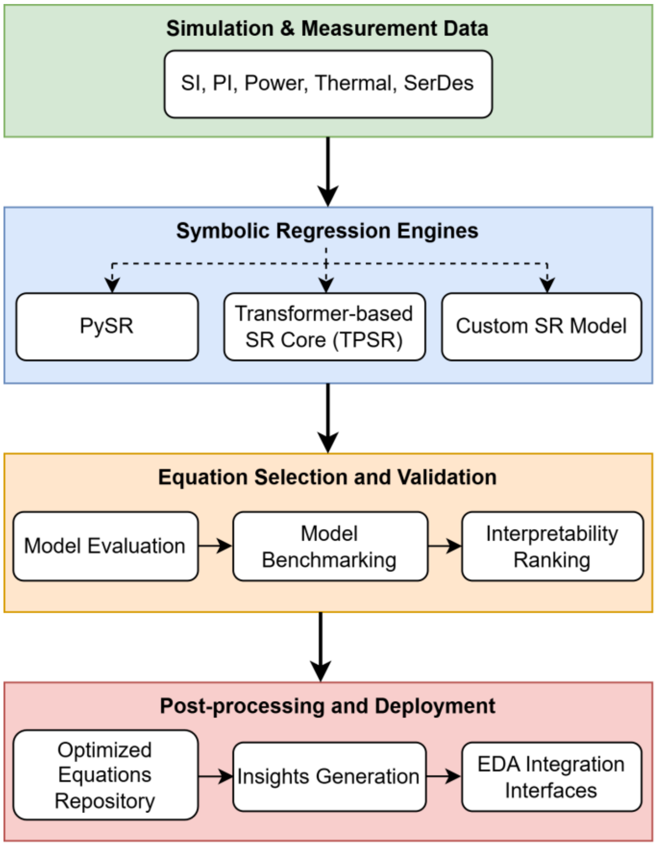


Fig. 1: The proposed framework for simulating EDA problems with Symbolic Regression

To our knowledge, this work represents the first application of SR to derive open- and constrained-form, human-interpretable models for SerDes performance modeling within an EDA workflow. Our contributions are as follows:

1) We present an end-to-end modeling framework augmented by SR, laying the path for developing faster, application-specific models for EDA workflows as shown in Fig.1.
2) We quantitatively evaluate tradeoffs with SR accuracy, data efficiency, and equation complexity in open-form EDA modeling.
3) We demonstrate the efficacy of transformer-based SR methods for discovering equation structure in closed-form problems.

II. Background

A. Symbolic Regression Definition

Symbolic Regression is a data-driven method for obtaining explicit mathematical expressions that map a given set of inputs to outputs by searching over a space of symbolic expressions and operators. In the most general sense, the goal is to simultaneously optimize accurate fitting to the data, minimize expression complexity to the minimum required, and maintain a degree of human interpretability in the final expression [7]. Further, SR methods go beyond recovering coefficients (as in traditional regression or curve-fitting techniques) and seek to recover structural relationships, thereby augmenting the final derived model.

B. Equation Structure Discovery

The central task in SR systems is discovering the structural form of an equation via combinatorial search of candidate solutions, which are represented as expression trees with internal nodes corresponding to mathematical operators linking external leaf nodes that represent input variables and constants. Over successive generations of search, this method gradually selects increasingly accurate symbolic structures. Classical approaches rely on genetic programming, where mutation and crossover operators explore the expression space guided by a fitness function that balances accuracy and model complexity. Recent SR Frameworks such as PySR augment the search with gradient-based optimization, improving convergence and computational efficiencies [8]. For further details we refer the reader to [8] and [9].

III. Methodology

A. Problem Definition

We define the modeling task in this work as discovering a symbolic function from the observed system's data to capture the underlying behavior of a given electronics system. The derived function, $f(\cdot)$, maps a d-dimensional input vector $\mathbf{x} \in \mathbb{R}^d$ (drawn from a dataset of n observations) to a target variable $y = f(\mathbf{x}) \in \mathbb{R}$ [9]. Further, we categorize **open-form** problems in EDA as cases where we do not know or expect to converge to a predetermined equation form [10]. In this methodology, we derive symbolic expressions that show the mathematical relationships between design parameters for systems without previously known equations. We can either leave the overall structure and terms of the expression completely unrestrained to obtain a purely numerical model, or constrain them to specific terms and complexity to streamline expression generation in a desired direction, modeling a **closed-form** problem.

In this work, we model a SerDes link, where system performance is governed by interactions among multiple equalization parameters. Specifically, the input vector $\mathbf{x}$ consists of CTLE settings and a set of DFE tap coefficients, while the target variables y correspond to performance metrics, namely Eye Height (EW) and Eye Width (EH). We then formulate the problem as follows -

$$\begin{aligned} y &= f(\text{CTLE}, \text{DFE}_1, \text{DFE}_2, \text{DFE}_3), \\ y &\in \{\text{EyeHeight}, \text{EyeWidth}\} \end{aligned} \tag{1}$$

B. SerDes Simulation and Dataset Generation

To generate data for SR, we simulated a 50 Gb/s SerDes link in Ansys Electronics Desktop (AEDT), with automation provided by the PyAEDT library [11]. The receiver contains a CTLE where the gain is tunable and a three-tap DFE. We performed a sweep over CTLE gain and corresponding DFE tap ranges to capture varying sets of equalization configurations.

For each input combination, we ran the simulation and extracted Eye Height and Eye Width metrics from the simulated eye diagrams. The final dataset thus consists of input–output pairs of the form $(\mathbf{x}, y)$, where the input vector $\mathbf{x} = [\mathrm{CTLE}, \mathrm{DFE}_1, \mathrm{DFE}_2, \mathrm{DFE}_3]$ represents the equalization settings, and the output variables y correspond to Eye Height or Eye Width. Here, 80% of the simulated configurations were used to derive the symbolic expressions, with the remaining 20% reserved for independent accuracy assessment.

C. Framework for Modeling SI Problems with SR

We obtained the SR models using the open-source PySR framework [8] for open-form problems, as well as an adapted transformer-based method (as laid out in TPSR [9]) to evaluate the suitability of SR for closed-form SI analysis. Three experimental configurations were considered to systematically assess the impact of model flexibility and input structure on accuracy and computational efficiency.

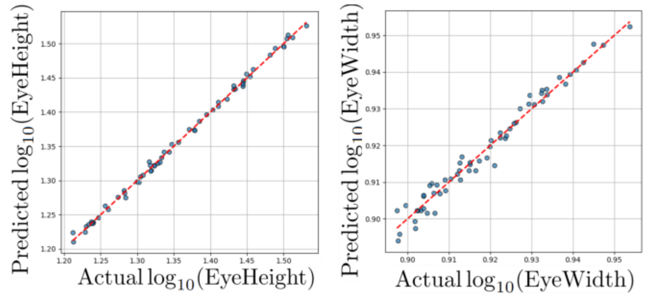
We provide a framework for integrating SR into SI analysis, as shown in Fig 1. We begin with generating or gathering simulation data, and organizing it as a set of input variables that map to the output. Next, we propose a choice of SR engines, with off-the-shelf engines such as PySR and TPSR providing speedy implementation over bespoke SR engines. To augment the selection of the most optimal engine, we propose a benchmark for evaluating the symbolic equations derived from each engine - this allows for flexibility in handling tradeoffs between accuracy and interpretability for our intended application. Finally, we build a repository of optimal symbolic expressions that completely model the system, which can be used to both generate unknown insights and speedily perform SI analysis during development cycles.

IV. Experimental Results

A. Open-Form SerDes Modeling with PySR

First, we used an unconstrained set of mathematical operators and expression structures, with minimal restrictions on expression complexity; this configuration was applied to a large dataset of 300 samples to validate the ability of SR to recover expressive models; we provide the full set of equalization parameters (CTLE, DFE_1, DFE_2, and DFE_3) as inputs. In this configuration, SR achieves excellent results, yielding R^2 values exceeding 0.99 and low MSE across validation samples, as shown in Fig. 2. The figure demonstrates the predicted-versus-actual comparisons for Eye Height and Eye Width, with the symbolic expressions yielding close alignment with the ideal input-output relationship.

The derived expressions for Eye Height and Eye Width are given in (2) and (3) respectively. The derived symbolic expressions are consistent with known SerDes behavior - saturating nonlinearities in the Eye Height model reflect the effects of gain saturation and noise. At the same time, the sinusoidal CTLE–DFE interaction captures coupled multi-tap ISI cancellation effects. The Eye Width expression exhibits exponential sensitivity, consistent with timing margin and jitter behavior in the equalization of high-speed links.



(a) Eye Height (b) Eye Width

Fig. 2: Results of SerDes modeling with full data regime of 300 samples; the model exhibits high accuracy in modeling unseen data.

$$\mathrm{EyeHeight} = 1.811 \tanh(1.115^{\mathrm{CTLE}} - 0.0239(1.186^{\mathrm{DFE3}} - \mathrm{DFE1} + \mathrm{DFE2})) \quad (2)$$

$$\mathrm{EyeWidth} = \frac{0.9194}{\left(1.008^{(1.339\mathrm{DFE2} + \mathrm{CTLE} + \sin \mathrm{DFE1})}\right)^{(0.8747^{\mathrm{CTLE}})}} \quad (3)$$

B. Low Data Regime Modeling with PySR

TABLE I
Low Data Regime

Configuration	Operators / Inputs	Compute Time	R^2	MSE
Unconstrained	All unary and binary operators	1 hour	0.9957	1.7×10^{-5}
Limited operators	No trigonometric operators	40 minutes	0.9906	3.8×10^{-5}
Limited inputs	No DFE tap values; all operators	25 minutes	0.8803	4.7×10^{-4}

Open-form SR results and corresponding fits for three low-data SerDes modeling configurations. Minimal performance degradation is observed under operator constraints. All experiments were run on an Intel i9-13900HX and an NVIDIA RTX 4090 Laptop GPU.

Second, we examined a constrained SR configuration by **restricting the data to 30 samples** to create a low data regime, as is often seen during development cycles. Simultaneously, we performed an ablation study, limiting the set of allowable operators and expression complexity. This setup enabled us to study the trade-off between model interpretability, computational overhead, and predictive accuracy when prior structural and operator constraints are imposed. Imposing these constraints slightly increases approximation error, the resulting models retain high predictive accuracy and exhibit reduced computational overhead of 20% during training, demonstrating a practical method for further reducing model complexity without significant performance tradeoffs.

These results showcase SR's ability to maintain high accuracy in derived models when available data is scarce. Furthermore, we clearly illustrate that operational overheads can be significantly reduced by limiting equation complexity through operator selection, thereby laying the groundwork for the development of lightweight, accurate, and bespoke models for numerous applications.

C. Closed-Form Symbolic Regression with TPSR

Third, we investigated the closed-form case, where we provide the known functional structure of the output expression; this setup demonstrated SR's ability to recover equation coefficients within a given structure. We model CTLE responses, and provide TPSR with an expected structure to converge to, with the goal of correctly recovering the same form of symbolic expression, and fitting the coefficients accurately. We use a combination of TPSR for structure discovery followed by Least Squares Coefficient Recovery to recover the symbolic expression, demonstrating the efficacy of SR for modeling known relationships from a structural and numerical standpoint. The canonical second-order CTLE transfer function is given in (4). The discovered symbolic model obtained via SR is shown in (5).

$$\hat{H}(s') = \frac{a_0 + a_1 s'}{1 + b_1 s' + b_2 s'^2}, \qquad s' = \frac{s}{\omega_0} \tag{4}$$

$$H(s) = \frac{1.00182 + 5.04894\, s}{1 + 1.14124\, s + 0.325607\, s^2},\ s = jx,\ x = \frac{\omega}{\omega_0} \tag{5}$$

V. Conclusion

In this work, we demonstrate the viability of symbolic regression (SR) as a fast, accurate, and interpretable modeling paradigm. As shown in (2) and (3), we derive compact, open-form expressions that map CTLE and DFE parameters to Eye Height and Eye Width with greater than 99% fidelity. Table I investigates SR's capabilities when the available data is significantly reduced, while the allowed operators are curtailed. A few interesting observations were made while evaluating the framework's performance in the open-form case.

1) Allowing the entire set of operators and input variables while modeling gives us an extremely accurate fit with negligible error.
2) Experimental results show that the performance does not significantly degrade when we limit operators; in fact, we achieve a 1.5× speedup.
3) The model struggles if we remove simulation variables in the modeling task, validating the relationship between overall system performance and variables present in the obtained symbolic expressions.

In the closed-form case, we demonstrate the efficacy of transformer-based SR in handling the challenging case of discovering a required equation's structure during symbolic modeling; as shown in (5), we achieve exact symbolic structure recovery and numerically accurate adherence to the canonical CTLE transfer function given in (4)

Future work will focus on extending this framework to additional problem domains, incorporating physics-guided constraints and priors, and scaling the approach to higher-dimensional design spaces and more complex equalization architectures. These directions suggest that symbolic regression has the potential to serve as a lightweight, interpretable surrogate modeling tool for accelerating design-space exploration and analysis in next-generation EDA flows.

VI. Acknowledgment

This research is supported in part by the NSF under Grant No. CNS 2137283 (Center for Advanced Electronics through Machine Learning) and the industry members of the CAEML IUCRC.

References

[1] R. A. d. L. Reis, "Eda: Overview and some trends," *Journal of Integrated Circuits and Systems*, vol. 17, no. 3, pp. 1–10, 2022, accessed: 2026-01-14. [Online]. Available: https://jics.org.br/ojs/index.php/JICS/article/view/679

[2] K. I. Gubbi, S. A. Beheshti-Shirazi, T. Sheaves, S. Salehi, S. M. PD, S. Rafatirad, A. Sasan, and H. Homayoun, "Survey of machine learning for electronic design automation," in *Proceedings of the Great Lakes Symposium on VLSI 2022*, ser. GLSVLSI '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 513–518. [Online]. Available: https://doi.org/10.1145/3526241.3530834

[3] J. Sifakis, "System design automation: Challenges and limitations," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 2093–2103, 2015.

[4] S.-M. Udrescu and M. Tegmark, "Ai feynman: A physics-inspired method for symbolic regression," *Science Advances*, vol. 6, no. 16, p. eaay2631, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.aay2631

[5] Y. Wang, N. Wagner, and J. M. Rondinelli, "Symbolic regression in materials science," *MRS Communications*, vol. 9, no. 3, pp. 793–805, 2019.

[6] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the National Academy of Sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[7] L. Biggio, T. Bendinelli, A. Neitz, A. Lucchi, and G. Parascandolo, "Neural symbolic regression that scales," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds., vol. 139. PMLR, 18–24 Jul 2021, pp. 936–945. [Online]. Available: https://proceedings.mlr.press/v139/biggio21a.html

[8] M. Cranmer, "Pysr: High-performance symbolic regression in python and julia," *Astrophysics Source Code Library*, pp. ascl–2409, 2024.

[9] P. Shojaee, K. Meidani, A. B. Farimani, and C. K. Reddy, "Transformer-based planning for symbolic regression," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. [Online]. Available: https://openreview.net/forum?id=0rVXQEeFEL

[10] Y. Chen, Y. Luo, Q. Liu, H. Xu, and D. Zhang, "Symbolic genetic algorithm for discovering open-form partial differential equations (sga-pde)," *Phys. Rev. Res.*, vol. 4, p. 023174, Jun 2022. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevResearch.4.023174

[11] Ansys, Inc., "Ansys Electronics Desktop," https://www.ansys.com/products/electronics, 2025, version 2025 R2.