Transformer-based Planning for Symbolic Regression




*Parshin Shojaee   *Kazem Meidani   Amir Barati Farimani   Chandan K. Reddy

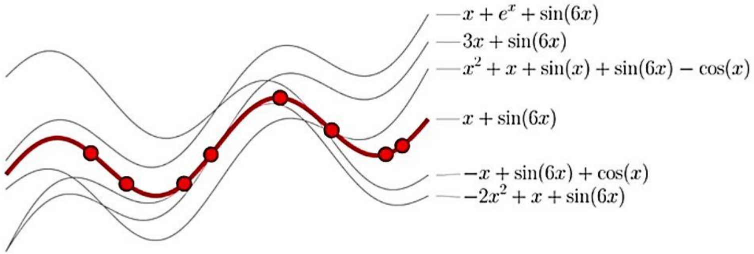
*Equal contribution

NeurIPS 2023 Presentation

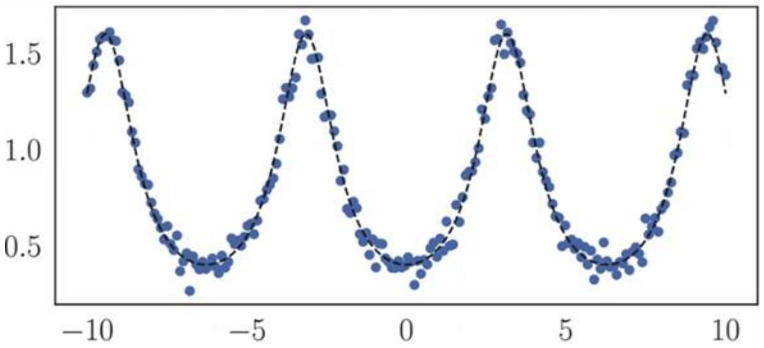
What is Symbolic Regression?

Given a dataset $(\boldsymbol{X}_i, \boldsymbol{y}_i)_{i \le N}$, where each point $\boldsymbol{X}_i \in \mathbb{R}^d$ **and** $\boldsymbol{y}_i \in \mathbb{R}$, find a **mathematical expression** $\boldsymbol{f} : \mathbb{R}^D \to \mathbb{R}$ such that $\boldsymbol{f}(\boldsymbol{X}_i) \approx \boldsymbol{y}_i$

Interpretability | Accuracy | Generalization



* Figure from Landajuela et al. (2023)



$$r = \frac{a(1 - e^2)}{1 + e\cos(\theta_1 - \theta_2)}$$

Related Work

Space of equations grows ***exponentially*** with equation length, containing both discrete and continuous components: $2.1\,x + \sin(6.5\,x)$

Symbolic Regression is NP-hard Virgolin et al. (2022)

SR without Prior Knowledge:

Genetic Programming

Search space exploration via genetic operators:
i) selection, ii) mutation/cross-overs
Virgolin et al (2019), Kommenda et al (2019), De Franca et al (2020)

Reinforcement Learning

Policy optimization for generating expressions
Petersen et al. (2020)
Monte Carlo tree search over expression tree
Sun et al. (2023)

- × **Lack of prior knowledge**
- × **Requires new search per dataset**
- × **Computationally inefficient**

Symbolic Regression is a Hard Combinatorial Problem

Large-scale Pre-training:

Train Encoder-Decoder Transformers to generate expressions (treat math as a language)

- ✓ **Leverage synthetic datasets**
- ✓ **Strong prior knowledge**
- ✓ **Fast Inference** ~ single forward pass

Skeleton-based Biggio et al. (2021)

i) Predict skeleton,
ii) Optimize constants

End-to-End Kamienny et al. (2022)

i) Predict full expression,
ii) Refine constants

- × **Pre-training mismatch with end task**
- × **Decoding depends only on logits**
- × **Lack of performance feedback**

Transformer-based Lookahead Planning for SR

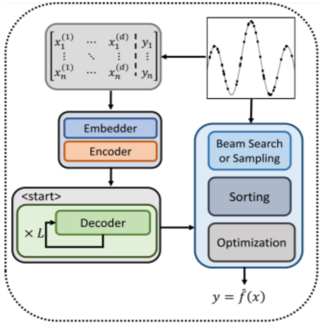
More details in the paper!

Typical approach

Predict expression with
Beam Search / Sampling:

- Propose multiple candidates
- Optimize constants
- Sort+Select candidates

Biggio et al. (2021), Kamienny et al. (2022)

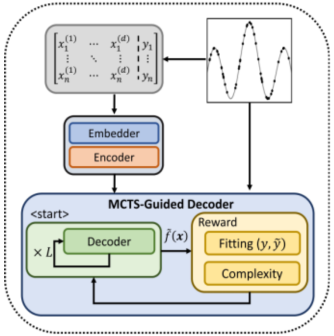


Inference with Beam Search/ Sampling

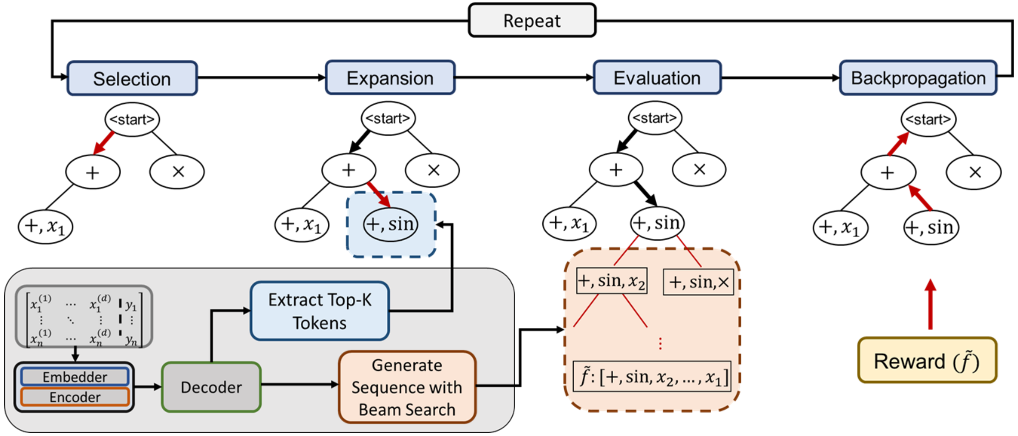
Our approach

Predict expression with
lookahead planning:

- Use Monte Carlo Tree Search (MCTS)
- Receive feedback during generation



Inference with MCTS



Selection Criterion

$$\text{P-UCB}(s,a) = \underbrace{Q(s,a)}_{\uparrow \textit{Exploitation}} + \underbrace{\beta \cdot P_\theta(a|s) \sqrt{\frac{\ln(N(s))}{1+N(s')}}}_{\uparrow \textit{Exploration}}$$

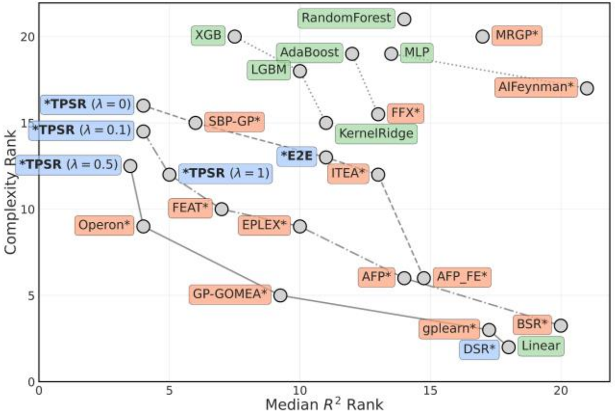
Reward Function

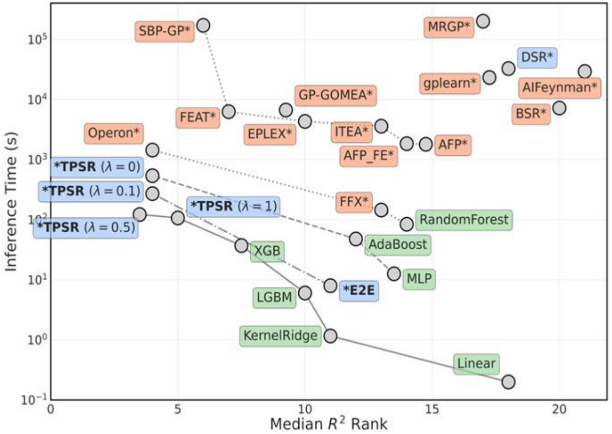
$$\text{R}(\tilde{f}) = \underbrace{\frac{1}{1+\text{NMSE}(y, \tilde{f}(x))}}_{\textit{Promoting Fitting Accuracy}} + \underbrace{\lambda \exp\left(-\frac{\ell(\tilde{f}(\cdot))}{L}\right)}_{\textit{Regulates Equation Complexity}}$$

Performance on SRBench

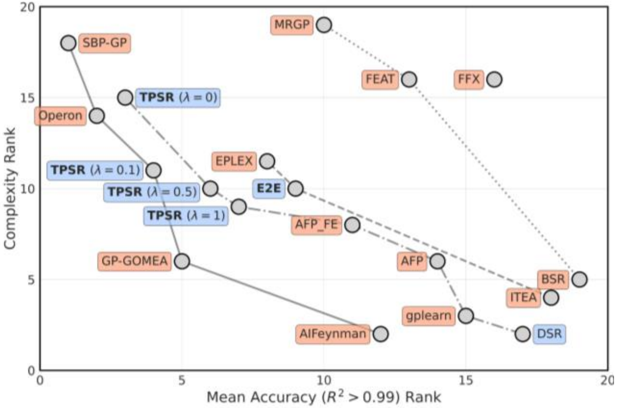
DL-based SR   GP-based SR   ML methods

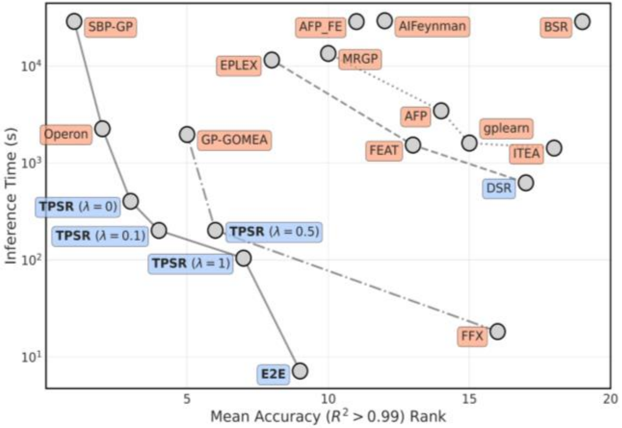
Real-world datasets
(Penn ML benchmark)





Symbolic regression datasets
(Feynman problems)





Additional Results

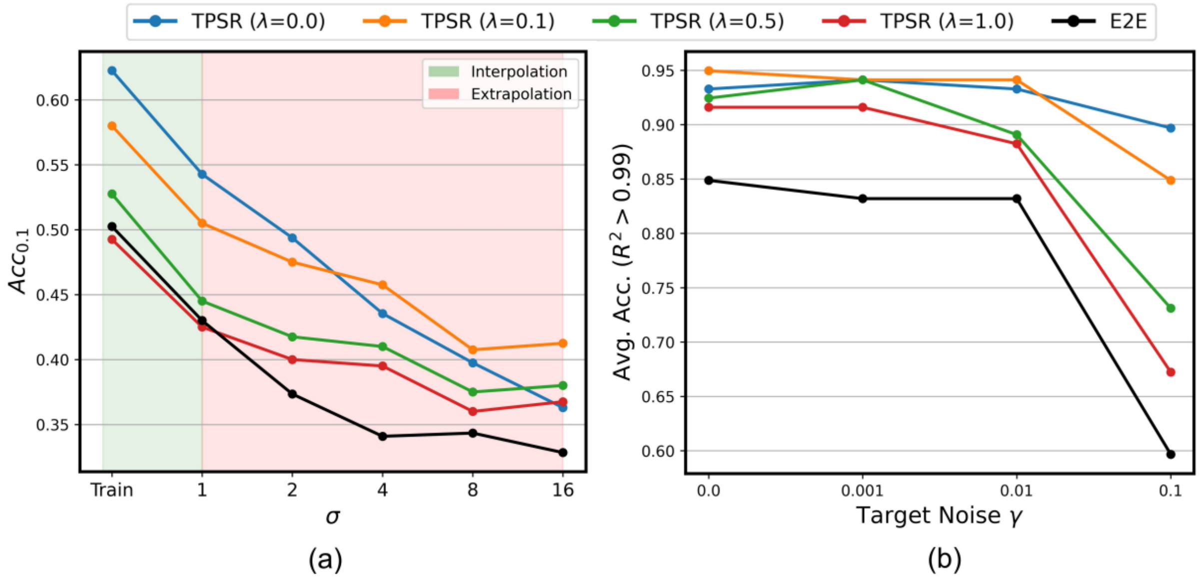
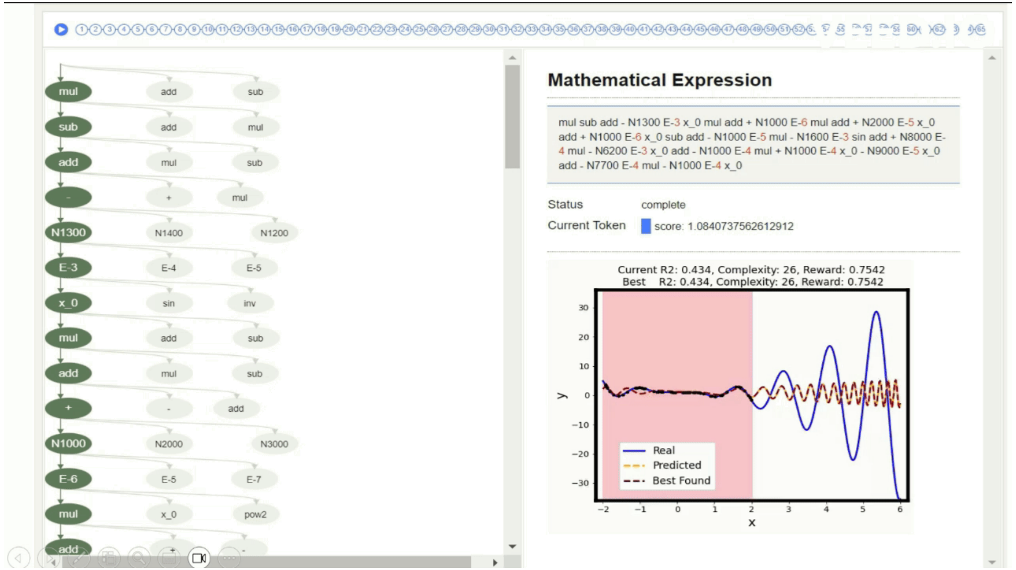


Figure 6: TPSR with $\lambda \in \{0, 0.1, 0.5, 1\}$ compared to E2E for **(a) Extrapolation performance** where in-domain accuracy is shown for different input variances (σ), and **(b) Robustness to noise**, where mean accuracy ($R^2 > 0.99$) is shown for various target noise levels (γ).

Thank you!



Code on GitHub:

parshinshojaee@vt.edu

VIRGINIA TECH

Carnegie Mellon University