Beyond Accuracy and Complexity: The Effective Information Criterion for Structurally Stable Symbolic Regression

Zihan Yu [1] **Guanren Wang** [1] **Jingtao Ding** [1] **Huandong Wang** [1] **Yong Li** [1]

Abstract

Symbolic regression (SR) traditionally balances accuracy and complexity, implicitly assuming that simpler formulas are structurally more rational. We argue that this assumption is insufficient: existing algorithms often exploit this metric to discover accurate and compact but structurally irrational formulas that are numerically ill-conditioned and physically inexplicable. Inspired by the structural stability of real physical laws, we propose the Effective Information Criterion (EIC) to quantify formula rationality. EIC models formulas as information channels and measures the amplification of inherent rounding noise during recursive calculation, effectively distinguishing physically plausible structures from pathological ones without relying on ground truth. Our analysis reveals a stark structural stability gap between human-derived equations and SR-discovered results. By integrating EIC into SR workflows, we provide explicit structural guidance: for heuristic search, EIC steers algorithms toward stable regions to yield superior Pareto frontiers; for generative models, EIC-based filtering improves pretraining sample efficiency by 2–4 times and boosts generalization R^2 by 22.4%. Finally, an extensive study with 108 human experts shows that EIC aligns with human preferences in 70% of cases, validating structural stability as a critical prerequisite for human-perceived interpretability. We release our code at https://github.com/tsinghua-fib-lab/EIC.

[1]Department of Electronic Engineering, BNRist, Tsinghua University, Beijing, China. Correspondence to: Jingtao Ding <dingjt15@tsinghua.org.cn>, Yong Li <liyong07@tsinghua.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

1. Introduction

Symbolic regression (SR) is a machine learning technique that discovers interpretable mathematical formulas describing relationships in data (Camps-Valls et al., 2023). Unlike black-box models, it reveals how inputs map to outputs in a form that can be analyzed mathematically or logically, offering insights and new scientific knowledge (Makke & Chawla, 2024). In a typical workflow, researchers apply SR algorithms to *data* to generate candidate formulas that balance accuracy and complexity (i.e., the *Pareto front*). They can then assess the credibility of these candidates and select the most reasonable *formula* for insights into the underlying patterns (Liu et al., 2024).

SR methods range from heuristic search algorithms to recent generative and neural-guided approaches (Cranmer, 2023; Kamienny et al., 2022; Shojaee et al., 2023; Yu et al., 2025). Despite their different search mechanisms, most methods ultimately rank or select candidate formulas using a trade-off between fitting accuracy and formula complexity. This standard accuracy–complexity framework relies on the implicit assumption that lower complexity implies higher rationality, which is not always valid. Complexity metrics, usually measured by the number of symbols in a formula (i.e., formula length) (La Cava et al., 2021; Aldeia et al., 2025), capture overall size but ignore internal structure, which is critical for physical plausibility. As illustrated in Figure 1, two formulas can have identical complexity and similar accuracy while exhibiting drastically different structural rationality: the left one features a nested, irrational structure (e.g., $\sin(\sin(\cot(x)))$) that is physically inexplicable and numerically ill-conditioned, whereas the right one, as a linear combination of simpler functions, aligns well with scientific intuition. Thus, existing SR algorithms can discover formulas that are accurate and compact but structurally irrational, limiting knowledge extraction and hindering the application of SR in scientific discovery.

To address this problem, we draw inspiration from the structural stability of real physical formulas. In Figure 2, we observe a stark structural stability gap: while human-derived equations are typically robust to computational noise (e.g., floating-point errors), formulas discovered by SR, despite achieving high accuracy and compactness, often suffer from

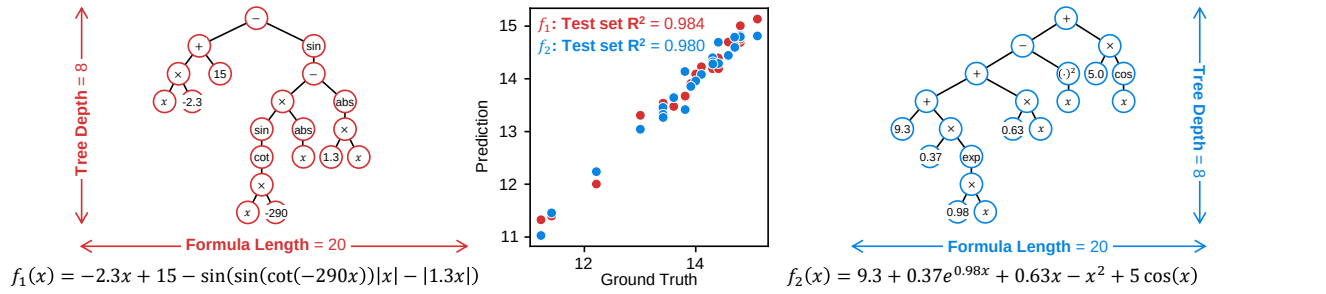


Figure 1. **Formulas with identical complexity and accuracy can exhibit distinct structural rationality.** Despite having the same length and R^2, the left formula contains pathological nesting, while the right one remains structurally sound, demonstrating that complexity fails to capture structural rationality.

critical structural instability. These mathematically pathological structures drastically amplify rounding noise and, critically, render the formulas irrational and uninterpretable. Consequently, we propose the Effective Information Criterion (EIC) to quantify structural stability. EIC measures the amplification of cumulative noise generated when inherent rounding noise in the calculation propagates in the formula, thereby effectively distinguishing physically plausible formulas from irrational ones.

Our contributions are as follows. First, we define EIC as a parameter-free metric that quantifies the structural stability of symbolic formulas. Second, we reveal a structural stability gap between real physical formulas and SR-discovered formulas. Third, we show that EIC improves both search-based and generative SR workflows: it guides heuristic search toward superior Pareto frontiers, and it filters unstable pre-training samples to boost sample efficiency by 2–4 times and achieve a 22.4% relative gain in R^2 on physical formulas. Finally, an evaluation involving 108 human experts confirms that EIC aligns with human preferences in 70% of cases, validating structural stability as a critical prerequisite for human-perceived interpretability.

2. Related Work

This section positions EIC relative to three lines of work: search-based SR algorithms, generative SR models, and existing evaluation metrics or constraints for formula selection.

Heuristic search-based SR methods. Traditionally, SR relies on heuristic search methods such as genetic programming (Augusto & Barbosa, 2000; Burlacu et al., 2020; Cranmer, 2023) or Monte Carlo tree search (Sun et al., 2023). Recently, advanced approaches have integrated pre-trained neural networks (Kamienny et al., 2023; Shojaee et al., 2023; Yu et al., 2025; Ying et al., 2025), online-trained deep reinforcement learning policies (Petersen et al., 2021; Tenachi et al., 2023; Xu et al., 2024), or large language models (Shojaee et al., 2025a) to guide the search algorithms. Furthermore, some works leverage neural networks to identify symmetries in data or perform ab initio variable selection (Ye & Li, 2025) to facilitate the search process (Udrescu & Tegmark, 2020). While these methods typically limit the maximal formula length to avoid excessive complexity, such coarse controls do not directly address the internal structural stability of candidate formulas. EIC is orthogonal to these search strategies: it provides a structural stability criterion that can be incorporated into different search pipelines to penalize unstable candidates.

Generative symbolic regression methods. In recent years, generative methods have emerged, which pre-train Transformer models on large-scale randomly generated formula – data pairs to predict mathematical symbols directly from data. These pre-trained models can serve various roles, including generating formulas directly (Biggio et al., 2021; Kamienny et al., 2022), guiding symbolic search algorithms (Shojaee et al., 2023; Kamienny et al., 2023; Yu et al., 2025; Ying et al., 2025), or acting as foundation models for downstream tasks (Meidani et al., 2023). Despite their promise, these methods face significant challenges in sample efficiency, partly because randomly generated training formulas can contain irrational structures that diverge from real-world physical laws. EIC provides a complementary data-filtering principle for pre-training: it removes structurally unstable formulas while preserving the diversity of synthetic data.

Evaluation metrics and constraints used in SR. Standard SR benchmarks, such as SRBench (La Cava et al., 2021), SRBench++ (de Franca et al., 2024), SRSD (Matsubara et al., 2022), and LLMSRBench (Shojaee et al., 2025b), typically employ a bi-objective framework balancing fitting accuracy (e.g., R^2, RMSE) and complexity (Fong & Motani, 2024). Common complexity metrics like formula length, depth, and minimum description length (Yu et al., 2025) capture overall size but, as Figure 1 illustrates, ignore internal structure. Other complexity measures have also been proposed. Visitation length counts the summed size of all subtrees and therefore mainly reflects the distribution of

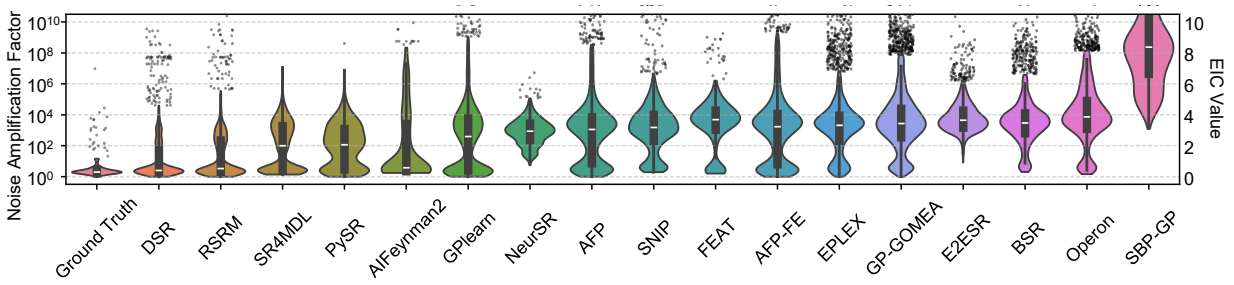


Figure 2. **The structural stability gap between real physical formulas and SR results.** We compare 133 ground truth formulas (Feynman, Strogatz) with results from 17 SR algorithms, quantified by the noise amplification factor (Section 3). The inner box shows the quartiles, and the black dots indicate outliers beyond $1.5\times$IQR.

nodes across depths (Kommenda et al., 2015). Function-symbol weights, as used in systems such as PySR (Cranmer, 2023), assign predefined costs to operators, while evaluation cost uses execution time as an operator-dependent weight (Dong & Zhong, 2025). These metrics account for operator counts or operator types, but still largely ignore how operators are structurally composed: for example, $\sin(x) + \sin(y)$ and $\sin(\sin(x + y))$ can receive similar operator-weight scores despite having very different internal structures. Order of nonlinearity instead measures complexity through the minimum polynomial degree needed to approximate a model (Vladislavleva et al., 2008), but this assumes that plausible formulas should be well represented by low-degree polynomials, which is not suitable for common physical expressions involving non-polynomial functions such as $\sin$, $\tan$, or $\sqrt{\cdot}$. Moreover, standard evaluations of out-of-distribution (OOD) performance or noise sensitivity focus solely on input-output mappings, overlooking internal rationality. For instance, while x and $(10000x - 9999x)$ yield identical mappings, the latter suffers from catastrophic cancellation, amplifying rounding noise by 4 orders of magnitude. Finally, regarding physical plausibility, methods imposing unit constraints (Tenachi et al., 2023) are largely limited to physics and difficult to apply in domains like sociology, where units are ambiguous. Meanwhile, manual expert evaluations of trustworthiness either reflect subjective preferences (La Cava et al., 2022) or are too time-consuming for standard benchmarks (Virgolin et al., 2023). Consequently, no scalable metric currently evaluates internal structural rationality. EIC fills this gap by explicitly measuring structural stability, complementing existing complexity-based and black-box evaluations.

3. Effective Information Criterion

3.1. Intuition and Definition for EIC

Motivation. As white-box models, symbolic formulas are defined critically by their internal structure, which determines their rationality (Figure 1), rather than just input-output mappings. We draw inspiration from history: foundational laws were verified using low-precision tools (e.g., slide rules with 3–4 significant digits), implying a strong inductive bias that valid formulas must remain robust under limited precision. Consequently, we propose evaluating rationality via structural stability. Viewing a formula as a computation tree (with inputs as leaves and operators as nodes), intrinsic rounding errors inevitably accumulate and amplify during propagation. Unlike real physical formulas, which are typically stable, SR-discovered models, even those with accurate fitting, often possess ill-conditioned structures that amplify computational noise by orders of magnitude (Figure 2), which renders them irrational. Therefore, we propose quantifying this noise amplification to effectively distinguish real physical formulas from irrational ones.

Mathematical Formalization. Formally, we represent a formula f as a symbol tree $\mathcal{T}[f]$ (as demonstrated in Figure 1). Each internal node k implements an elementary operator e_k (e.g., $+$, $\times$, $\sin$) with child nodes $\mathcal{C}[k]$. Given an input sample x drawn from a data distribution $\mathcal{D}$, the ideal, noise-free value $y_k(x)$ at node k can be obtained recursively via $y_k(x) = e_k(\{y_i(x)\}_{i\in\mathcal{C}[k]})$, where $e_k(\cdot)$ represents the ideal operator output computed under infinite precision. Under finite precision, however, the computed value $\tilde{y}_k$ at node k deviates from y_k due to the inherent rounding noise introduced at this node and the propagated error accumulated from the child nodes. Since the magnitude of rounding error is proportional to the value itself, it can be modeled as a multiplicative noise with zero mean and a fixed variance:

Assumption 3.1 (Rounding noise as multiplicative noise)**.**

$$\tilde{y}_k = e(\{\tilde{y}_i\}_{i\in\mathcal{C}[k]}) \times (1 + \epsilon_k), \tag{1}$$

where ϵ_k is an independent random variable[1] following any distribution with $\mathbb{E}[\epsilon_k] = 0$ and $\mathrm{Var}[\epsilon_k] = \sigma^2$.

[1]Although rounding error is deterministic for a fixed input x, quantization theory suggests that as computational precision increases (i.e., variance of the rounding error decreases), the error behaves pseudo-randomly and becomes asymptotically uncorrelated with x (Widrow et al., 1996)

To quantify the extent to which the introduced rounding error is amplified due to accumulation during propagation, we define the cumulated error $\eta_k \triangleq (\tilde{y}_k - y_k)/y_k$ and the variance amplification factor

$$s_k^2(x) \triangleq \lim_{\sigma \to 0} \frac{\mathrm{Var}[\eta_k|x]}{\sigma^2}, \tag{2}$$

which quantifies the sensitivity of the subtree rooted at node k to rounding noise at high precision limits ($\sigma \to 0$) by the degree to which accumulated noise is amplified compared to rounding noise. Note that $s_k(x)$ is a function of x since the same subtree structure can have different sensitivities under different input data. Intuitively, a small $s_k(x)$ suggests the subtree is structurally stable, while a large $s_k(x)$ suggests the subtree acts as a noise amplifier, significantly magnifying small perturbations from upstream calculations.

To analyze how the error accumulation is intrinsically driven by the mathematical structure of the formula, we derive its recursive calculation rule (See Appendix A.1 for detailed derivation):

Proposition 3.2 (Recursive Relation of $s_k(x)$)**.**

$$s_k^2(x) = 1 + \sum_{i \in \mathcal{C}[k]} \kappa_{k,i}^2(x) \times s_i^2(x), \tag{3}$$

where $\kappa_{k,i} \triangleq \frac{y_i}{y_k}\frac{\partial e_k}{\partial y_i}$ is the partial relative condition number of the operation e_k with respect to its operand i. For leaf nodes where $\mathcal{C}[k] = \emptyset$, we assume its $s_k^2(x) = 1 + 0 = 1$.

This reveals that $s_k(x)$ is a deterministic structural property, computable solely from the formula's structure and input data x without relying on external parameters. Physically, it decomposes the total instability into two components, including the *intrinsic rounding noise* generated at the current node (represented by the constant term 1) and the *propagated noise* from child nodes ($s_i^2(x)$), which is amplified by the squared condition number $\kappa_{k,i}^2$.

Definition of EIC. Based on the node-wide structural instability measure $s_k(x)$, we define the Effective Information Criterion (EIC) as

$$\mathrm{EIC} \triangleq \max_{k \in \mathcal{T}[f]} \log_{10} \bar{s}_k. \tag{4}$$

Here, $\bar{s}_k^2 \triangleq \mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)]$ averages local point-wise instability into a global metric. This expectation ensures the evaluation is contextualized within the scientifically relevant domain $\mathcal{D}$, accounting for nonlinear structures where stability fluctuates across regions (e.g., $1/x$ is more unstable when x is smaller). The $\log 10$ term maps $s_k \in [1, \infty)$ to a linear scale of precision loss, providing clear physical meaning. Crucially, we employ $\max_k$ rather than the root metric s_{root} to impose a strict worst-case constraint. This prevents internal instabilities from being masked by insensitive downstream operations (specifically where $\kappa_{k,i} \approx 0$; see Appendix A.2). This definition, combined with Eq (3), allows recursively computing $s_k(x)$ and EIC with $O(N)$ linear complexity via post-order traversal (Algorithm 1).

Algorithm 1 CalculateEIC (Recursively calculate partial relative condition number)

Input: Formula f and data distribution $\mathcal{D}$ with N samples.
Output: Variance amplification factor $s(x)$ and output value $y(x)$ at root node, as well as EIC value of f.

$k \leftarrow f$.root % Start from the root node
if k is a Leaf Node (Variable or Constant) **then**
  $y_k(x) \leftarrow$ Evaluate$(k, \mathcal{D})$ % Get values of k in $\mathcal{D}$
  $s_k^2(x) \leftarrow 1$
  $\bar{s}_k^2 \leftarrow \mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)]$ % $\bar{s}_k = 1$ for leaf nodes
  $\mathrm{EIC}_k \leftarrow \log_{10} \bar{s}_k$ % EIC $= 0$ for leaf nodes
  return $(s_k(x), y_k(x), \mathrm{EIC}_k)$
else
  $\mathrm{EIC}_k \leftarrow 0$
  $e_k \leftarrow k$.operator
  for each child $i \in \mathcal{C}[k]$ **do**
    $f' \leftarrow$ Subtree rooted at i
    $(s_i(x), y_i(x), \mathrm{EIC}_i) \leftarrow$ CalculateEIC$(f', \mathcal{D})$
    $\mathrm{EIC}_k \leftarrow \max\{\mathrm{EIC}_k, \mathrm{EIC}_i\}$
  end for
  $y_k(x) \leftarrow e_k(\{y_i\}_{i\in\mathcal{C}[k]})$
  for each child $i \in \mathcal{C}[k]$ **do**
    $\kappa_{k,i}(x) \leftarrow \frac{y_i(x)}{y_k(x)}\frac{\partial e_k}{\partial y_i}|_{x\sim\mathcal{D}}$
  end for
  $s_k^2(x) \leftarrow 1 + \sum_{i\in\mathcal{C}[k]} \kappa_{k,i}^2(x) s_i^2(x)$
  $\bar{s}_k^2 \leftarrow \mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)]$
  $\mathrm{EIC}_k \leftarrow \max\{\mathrm{EIC}_k, \log_{10} \bar{s}_k\}$
  return $(s_k(x), y_k(x), \mathrm{EIC}_k)$
end if

Intuitive understanding of EIC. The EIC quantifies the structural stability of a formula f over the data distribution $\mathcal{D}$ by the amplification of cumulative noise η_k generated when inherent rounding noise η_k in the calculation propagates in the formula. This can be seen more clearly through (see Appendix A.3 for proof)

$$\mathrm{EIC} = \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\mathrm{Std}[\eta_k]}{\mathrm{Std}[\epsilon_k]}, \tag{5}$$

where $\mathrm{Std}[\cdot] = \sqrt{\mathrm{Var}[\cdot]}$ is the standard deviation of random variables that reflects their strength. This relationship clearly illustrates that EIC is the magnitude by which the intensity of the accumulated noise is amplified compared to the added perturbation when a sufficiently small perturbation is added during the calculation of the formula, which is just as shown

by the coordinate axes on both sides of Figure 2.

Theoretical Properties. Besides its intuitive understanding, EIC exhibits several desirable properties. First, it is parameter-free. As Eq. (3) shows, calculating $s_k(x)$ depends solely on structure and partial condition numbers, independent of external hyperparameters like noise level σ. Second, EIC is scale-invariant. Since condition numbers are dimensionless, EIC remains invariant under unit changes (e.g., rescaling meters to centimeters), as detailed in Appendix A.4. Finally, EIC is sensitive to algebraic rewriting. For instance, $f(x) = x$ and its redundant equivalent $g(x) = \ln(\exp(x))$ yield distinct values. Specifically, when $\mathbb{E}_{\mathcal{D}}[x] \triangleq \mu \gg 1$, we find $\text{EIC}(g) \approx \log_{10}(\mu) > \text{EIC}(f) = 0$ (see Appendix A.5). Although mathematically equivalent, they differ as calculation graphs: g introduces unnecessary operations that amplify noise. This aligns with parsimony but relies on computational stability rather than symbol counting.

3.2. Physical Interpretations

While the definition of EIC clearly elucidates its capability to quantify structural stability, linking it to established physical concepts can further strengthen its theoretical foundation

Numerical Precision Perspective. EIC measures structural stability under multiplicative noise, which can be physically interpreted as the loss of significant digits. According to quantization theory (Widrow et al., 1996), the operation of rounding to N significant digits is statistically equivalent (in the variance sense) to injecting multiplicative noise with variance $\sigma^2 \propto 10^{-2N}$. Leveraging this equivalence, we can formally link EIC to the reduction in effective precision (see Appendix A.6 for the detailed derivation):

$$\text{EIC} = \lim_{N\to\infty} (N - \min_{k\in\mathcal{T}} M_k), \tag{6}$$

where N denotes the working precision in significant digits (corresponding to σ^2), and M_k represents the effective significant digits remaining at node k after noise accumulation. Therefore, EIC quantifies the maximum magnitude of the intrinsic precision loss imposed by the formula's structure in the asymptotic limit of infinite precision ($N \to \infty$). As demonstrated in Figure 2, EIC values of SR-discovered formulas can reach up to 6–8, implying a catastrophic intrinsic loss of 6–8 significant digits. In sharp contrast, ground-truth physical laws typically exhibit $\text{EIC} < 1$, reflecting a natural preference for structurally robust representations that preserve information fidelity.

Signal Processing Perspective. Complementing the static numerical precision view, a dynamic perspective treats the formula as a signal transmission channel, where the input data represents the source signal and the rounding errors act as channel noise. In this signal processing system, noise figure serves as a fundamental metric that quantifies the degradation of the Signal-to-Noise Ratio (SNR) as information traverses the system: $F_k \triangleq 10\log_{10}\frac{\text{SNR}_{\text{intrinsic}}}{\text{SNR}_k}$ (in decibels), where $\text{SNR}_{\text{intrinsic}} \triangleq 1/\sigma^2$ represents the theoretical upper bound of signal quality determined by machine precision, and $\text{SNR}_k \triangleq 1/\text{Var}[\eta_k]$ denotes the actual SNR remaining at node k after structural noise amplification. As derived in Appendix A.6, EIC can be formally interpreted as the noise figure (scaled by $\frac{1}{20}$) at the bottleneck node:

$$\text{EIC} = \frac{1}{20}\lim_{\sigma\to 0}\max_{k\in\mathcal{T}[f]} F_k. \tag{7}$$

This mapping reveals a profound insight: a high EIC indicates that the formula acts as a noisy channel that severely corrupts the information flow. For instance, while real physical formulas act as high-fidelity channels ($\text{EIC} < 1$, or < 20dB loss) that preserve information integrity, a SR-discovered formula with a $\text{EIC} \approx 6$ acts as destructive channels with a 120dB SNR drop, implying that the signal quality (SNR) has degraded by twelve orders of magnitude (a $10^{12}\times$ loss in power) relative to the input and rendering the output signal virtually indistinguishable from noise.

4. Experiments

In this section, we first reveal the structural stability gap in existing SR methods using EIC (Section 4.1). Then, we demonstrate the practical utility of EIC in enhancing search-based performance (Section 4.2), improving generative sample efficiency (Section 4.3), and validating its alignment with human expert preferences for interpretability (Section 4.4).

4.1. Structural Stability Gap in Existing SR Methods

Experimental setups. To assess the structural rationality of existing SR discoveries, we benchmark 17 symbolic regression methods on 133 white-box problems from SRBench (119 Feynman and 14 Strogatz formulas), comparing the EIC distributions of the discovered formulas against the ground-truth physical formulas. The dataset of each problem is split into 75% training and 25% test data. Each method is executed 10 times per problem, with a filtering step that excludes poorly fitting results (Test set $R^2 < 0.8$) to ensure the analysis focuses on accurate candidates.

Figure 3 presents the average complexity and structural stability (EIC) of formulas discovered by 17 SR methods. Compared to the ground-truth physical formulas, the results reveal the following patterns regarding structural rationality:

Rigid constraints limit expressiveness. Methods such as DSR (Petersen et al., 2021) and RSRM (Xu et al., 2024) achieve the lowest EIC and complexity values, primarily due to their rigorous, manually designed constraints (e.g. for-

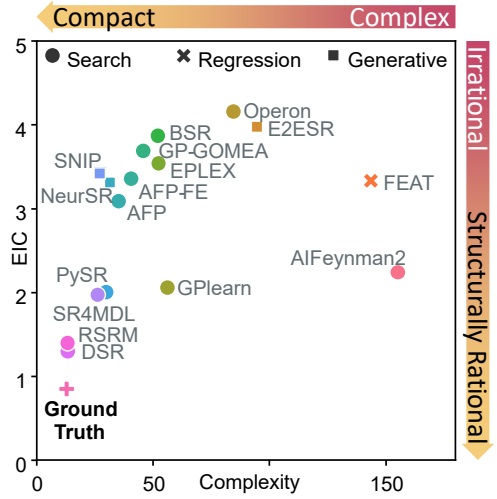


Figure 3. **The structural stability gap between real physical formulas and SR-discovered ones.** The figure shows the average complexity and EIC of formulas discovered by 17 SR methods on 133 SRBench white-box problems (SBP-GP is not displayed due to its high EIC). The underfit formulas with test set R^2 below 0.8 are ignored to focus on accurate candidates.

bidding nested trigonometric functions or mutually inverse operators). While these hard constraints effectively prune ill-conditioned structures, they overly restrict the search space, leading to inferior predictive accuracy (as we will demonstrate in the Pareto front analysis in Section 4.2). This highlights the necessity for a flexible, quantitative metric like EIC to guide the search rather than restricting it.

Difference between complexity and structural stability. Despite exhibiting the highest complexity, AIFeynman2 maintains a moderate EIC. We attribute this to physics-inspired strategies, such as divide-and-conquer and symmetry discovery, which build formulas from physically meaningful components. This confirms EIC identifies AI Feynman's structural rationality, measuring intrinsic robustness rather than merely penalizing length.

Current objectives overlook structural stability. Apart from DSR and RSRM that overly restrict the search space, existing approaches like PySR (Cranmer, 2023) and SR4MDL (Yu et al., 2025), despite matching the compactness of the ground truth, fail to match its stability. Specifically, their EIC values ($\approx$ 2.0) are significantly higher than real physical formulas ($<$ 1.0), indicating a difference in noise amplification of over an order of magnitude. This suggests that the standard bi-objective framework (balancing accuracy and complexity, or even absolute Pareto optimality (Fong & Motani, 2024)) is insufficient to prevent structurally irrational solutions, which motivates us to integrate EIC as a third objective to explicitly steer the search toward structurally rational regimes (as implemented in Section 4.2).

Generative priors bias toward structural instability. While generative methods like SNIP (Meidani et al., 2023) and NeurSR (Biggio et al., 2021) match search-based counterparts (PySR, SR4MDL) in complexity, they exhibit significantly higher EIC. This instability extends to Large Language Models (Appendix C.1), whose synthesized formulas (Shojaee et al., 2025b) remain far less stable than physical ground truths. This suggests that pre-trained models, despite producing compact formulas, often hallucinate noise-amplifying substructures. We attribute this to the distributional shift between random pre-training data and physical reality, leading to poor generalization. Consequently, we employ EIC to filter pre-training corpora, aligning the training distribution with reality to boost sample efficiency and structural rationality (Section 4.3).

4.2. Incorporating EIC into Heuristic Search Methods

Experimental setups. To evaluate EIC's efficacy in guiding search, we integrated it into two representative heuristic algorithms: Genetic Programming (GP) (Augusto & Barbosa, 2000) and Monte Carlo Tree Search (MCTS) (Sun et al., 2023), which underpin widespread methods like PySR, TPSR (Shojaee et al., 2023), SR4MDL, etc. We modify the standard fitness function $f = \eta^C/(1 + \text{NMSE})$, where C denotes the formula complexity (length), $\text{NMSE} = \text{MSE}/\text{Var}(y)$, $\eta < 1$ penalizes formula complexity (Sun et al., 2023), introducing an EIC penalty term: $f_{\text{new}} = f - \alpha \cdot \text{EIC}$, where α steers the search away from structurally irrational solutions (see Appendix B.1 for details). Notably, we employ linear regression to optimize constants during the search, avoiding more costly nonlinear optimization algorithms such as BFGS(Fletcher, 2000) and accelerating the search. We benchmark these EIC-enhanced variants against 17 baselines on SRBench, comprising 133 white-box and 122 black-box problems. For each problem, we conduct 10 independent trials with a 4-hour limit using a 75%/25% train-test split. For white-box tasks, we further assess robustness by injecting Gaussian noise at three levels $\{10^{-3}, 10^{-2}, 10^{-1}\}$. Performance is evaluated based on the Accuracy-Complexity Pareto frontier, which is the standard metric in SR. Besides, we also evaluate the results' EIC to assess structural rationality.

White-box experimental results. Figure 4 (left) summarizes white-box results (full results in Appendix C.3). Integrating EIC advanced MCTS and GP from the second to the first Pareto tier, improving both their accuracy and compactness. This improvement can be attributed to EIC explicitly penalizing structurally irrational patterns, which prevents excessive operator stacking and acts as a regularizer to enhance generalization. Consequently, although EIC targets an objective distinct from the standard Accuracy-Complexity trade-off, it synergistically improves both metrics. This benefit stems directly from EIC's unique physical defini-

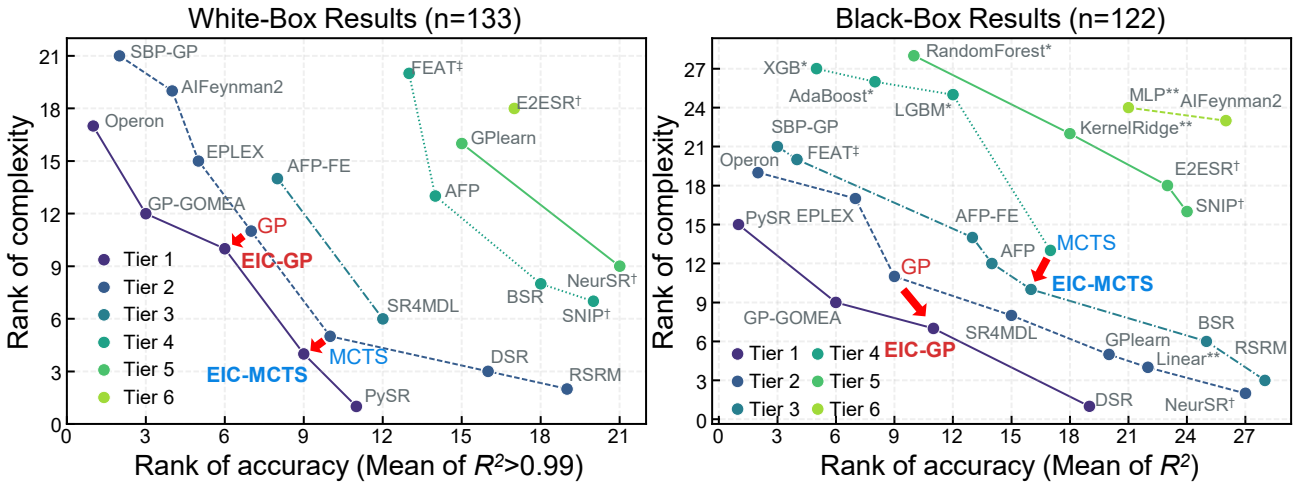


Figure 4. **Pareto fronts on the SR benchmark.** †, ‡, ∗, and ∗∗ indicate generative, regression, decision-tree, and black-box methods, respectively, while others are search-based methods. The lines show the Pareto front tiers, from bottom-left (best) to top-right (worst).

tion; merely adding an orthogonal complexity metric cannot improve search performance in the same way, as shown in Appendix C.4. This structural improvement is corroborated by Appendix Figure 9, where EIC values for both methods drop significantly toward ground-truth levels—a natural consequence of explicitly minimizing EIC during the search. In contrast, while DSR and RSRM yield compact formulas with low EIC (Figure 3), their significantly lower accuracy suggests that their manual constraints, while effectively avoiding unreasonable structures, inevitably shrink the search space and limit expressiveness.

Black-box experimental results. Figure 4 (right) presents results on 122 black-box problems (full results in Appendix C.3). Incorporating EIC advanced the Pareto fronts of GP (Tier 2 → 1) and MCTS (Tier 4 → 3), demonstrating that excluding structurally unreasonable formulas enhances performance even without ground truth. This is corroborated by significantly reduced EIC values in Appendix Figure 9. Notably, this reduction is more pronounced than in white-box settings, suggesting EIC effectively counters SR's tendency to rely on overly complex forms when underlying relationships are opaque. However, while MCTS improved in both accuracy and simplicity, GP improved simplicity but slightly reduced accuracy. This is probably due to a trade-off between numerical approximation and structural rationality. Standard GP often exploits numerically unstable structures (e.g., high-degree polynomials) as universal approximators to minimize error, yielding accurate but physically meaningless formulas. EIC constrains the search to stable forms, purposely sacrificing the marginal accuracy gained from such unstable approximations to ensure physical plausibility.

4.3. Incorporate EIC into Generative SR Methods

Current generative approaches, such as E2ESR (Kamienny et al., 2022), SNIP (Meidani et al., 2023), and SR4MDL (Yu et al., 2025), typically rely on the random formula generation algorithm developed by Lample & Charton (2020) for pretraining. While this mechanism provides unlimited training data, the generated synthetic formulas often exhibit unstable structures rarely seen in real-world physical equations. Consequently, models pre-trained on such data suffer from severe sample inefficiency, often requiring tens of millions of samples to achieve effective generalization.

To address this, we propose an EIC-guided rejection sampling strategy to filter the pretraining corpus. Specifically, during data generation, we evaluate the EIC value of each sample and discard those exceeding a threshold θ, regenerating them until the condition is met. This process removes structurally implausible samples while preserving the overall diversity of the training data. Based on the distribution of real physical formulas shown in Figure 2, we set $\theta = 2$ to enforce structural alignment with physical reality (see Appendix B.2 for details).

Experimental setups. In this experiment, we applied this EIC-guided rejection sampling strategy to E2ESR, SNIP, and SR4MDL, covering the progression from early baselines to state-of-the-art methods. Each method was trained on both unfiltered (random) and filtered (EIC < 2) formulas, and evaluated on the 133 SRBench white-box problems. Performance was measured using R^2 for E2ESR and SNIP, and RMSE/MAE for SR4MDL[2]. Training on unfiltered data

[2]Strictly speaking, SR4MDL predicts minimum description length rather than tokens. However, as it shares the same architecture and data scheme as E2ESR, we include it for comparison.

was conducted until test performance plateaued (approx. 5 million samples), serving as the baseline target to evaluate the sample efficiency gains of the EIC-filtered training.

Additionally, we benchmark against a recent data construction baseline, PhyE2E (Ying et al., 2025), which uses LLMs to generate "look-physical" formulas with unit constraints. We trained E2ESR on 180k PhyE2E formulas. To ensure a fair comparison, given its limited dataset size, we employed a mixed sampling strategy: sampling from PhyE2E formulas with a probability of 0.1 and generating random formulas with a probability of 0.9.

EIC-guided filtering reduces the distribution gap between training samples and real physical formulas. We evaluated whether EIC filtering produces a training distribution closer to real-world formulas. We generated 1024 formulas from the unfiltered random generator and 1024 from the filtered generator, and compared them with three benchmark sets, including Feynman (119 physics formulas), Strogatz (14 ODE formulas), and Wiki Named Equations (984 equations, see Appendix C.5 for details). Similarity was measured via variable counts, constant counts, operator counts, and formula length, using Jensen-Shannon (JS) and Kullback-Leibler (KL) divergences.

Table 1 shows that EIC filtering yields distributions substantially closer to real formulas, with $20 \sim 50\%$ reduction in JS divergence and $30 \sim 70\%$ reduction in KL divergence compared to the unfiltered random formulas. This demonstrates that EIC effectively reduces the gap between synthetic and real-world formulas and improves train-test alignment.

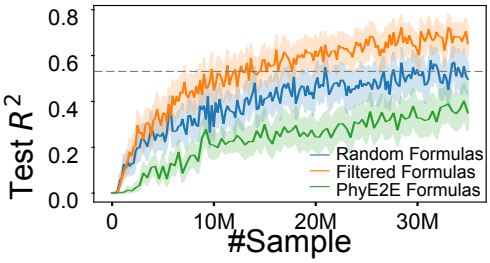


Figure 5. **Generalization performance of E2ESR trained on different samples.** The grey line shows the convergence trained on random formulas.

EIC-guided filtering boosts sample efficiency and generalization performance. The results are summarized in Table 2, where models trained on randomly generated formulas eventually converged on the Feynman test set, but typically required tens of millions of samples. In contrast, training on EIC-filtered formulas achieved equal or better performance with fewer training samples, improving sample efficiency by 357%, 233%, and 287% on E2ESR, SNIP, and SR4MDL, respectively. This demonstrates that EIC filtering removes structurally unreasonable formulas, producing training data that is more physically meaningful and transferable to real-world tasks, thereby enhancing pretraining-based symbolic regression.

Table 1. **Distribution differences between real physical formulas and generated ones.** Each value reports the JS or KL divergence between a generated formula set and a real physical formula set: **F.** = Feynman ($n = 119$), **S.** = Strogatz ($n = 14$), and **W.** = Wiki Named Equations ($n = 984$). "Naive" denotes 1024 randomly generated formulas, and "Filtered" denotes 1024 EIC-filtered formulas (EIC < 2.0).

		#Variables		#Coefficients		#Operators		Formula Length	
		JS↓	KL↓	JS↓	KL↓	JS↓	KL↓	JS↓	KL↓
F.	Naive	0.1196	1.4390	0.5163	16.78	0.3233	9.539	0.3973	13.615
	Filtered	0.0768	0.3523	0.3769	5.582	0.1862	0.7741	0.2573	3.4397
	$-\Delta(\%)$	**36%**	**76%**	**27%**	**67%**	**42%**	**92%**	**35%**	**75%**
S.	Naive	0.4014	18.784	0.6016	23.37	0.5726	22.24	0.6002	22.27
	Filtered	0.2650	12.920	0.5048	20.78	0.4646	18.74	0.5225	20.05
	$-\Delta(\%)$	**34%**	**31%**	**16%**	**11%**	**19%**	**16%**	**13%**	**10%**
W.	Naive	0.1459	0.6654	0.5566	14.888	0.3873	4.269	0.4219	7.060
	Filtered	0.0747	0.2899	0.4660	3.5307	0.2485	1.124	0.3114	1.541
	$-\Delta(\%)$	**49%**	**56%**	**16%**	**76%**	**36%**	**74%**	**26%**	**78%**

Table 2. **Sample efficiency of different methods trained with random and filtered formulas.** #S is the number of samples.

Pre-training	**E2ESR**		**SNIP**		**SR4MDL**		
Formulas	#S↓	R^2↑	#S↓	R^2↑	#S↓	MAE↓	RMSE↓
Random	35M	0.5190	40M	0.5300	50M	6.972	8.731
Filtered	**9.79M**	**0.5399**	**25.15M**	**0.5415**	**17.4M**	**6.812**	**8.701**
ΔEfficiency	357.5%		198.8%		287.4%		

Figure 5 shows that training on EIC-filtered samples further improved final performance compared to random formulas. For E2ESR, training for the same number of steps increased R^2 from 0.55 to 0.68, representing a 22.4% relative gain. Similar improvements were observed for SNIP and SR4MDL, with relative gains of 13.5% and 5.14%, respectively (see Appendix C.5). Incorporating PhyE2E formulas into pretraining did not improve performance and slightly reduced generalization, likely due to a trade-off between unit consistency and formula diversity: although PhyE2E formulas satisfy unit constraints, their limited diversity reduces pretraining effectiveness. In contrast, EIC filtering preserves both structural plausibility and distributional diversity, thus yielding superior generalization.

4.4. EIC Aligns with Human Preferences for Interpretability

Experimental setups. To validate EIC's alignment with human interpretability preferences, we conducted a controlled study comparing formula pairs with similar accuracy and complexity but distinct EIC values. We pooled formulas discovered by search-based symbolic regression methods across black-box problems (Section 4.2). To ensure intuitive assessment, we restricted analysis to 1D and 2D problems allowing clear visualization. From this pool, we rigorously selected pairs with indistinguishable fitting

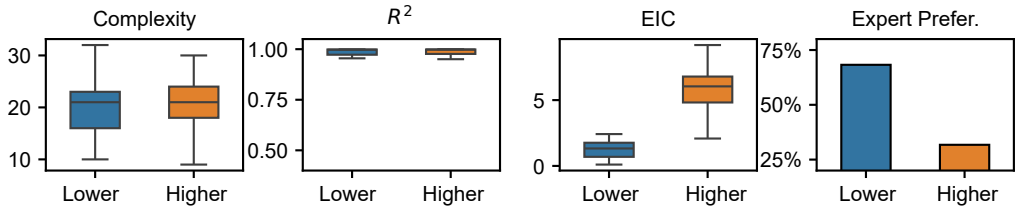


Figure 6. **Comparison of formula pairs**. "Lower" and "Higher" are formulas with the relatively lower/higher EIC values within each pair, whose complexity and accuracy (R^2) are indistinguishable but elicit a decisive expert preference for the lower candidates.

accuracy ($\Delta R^2 \leq 0.02$) and complexity ($\Delta C \leq 2$), yet significant structural divergence ($|\Delta\text{EIC}| \geq 2$). We invited 108 science/engineering volunteers to evaluate 10 random pairs each, selecting the more interpretable formula based on visualized behavior and mathematical form. In total, 870 valid evaluations were collected (see Appendix C.6).

EIC Aligns with Human Expert Preferences for Interpretability. The results, summarized in Figure 6, demonstrate a strong concordance between EIC scores and expert preferences. Although the paired formulas possess comparable distributions of complexity and accuracy (left panels), human experts exhibited a decisive preference for the formulas with lower EIC scores, favoring them in 69.8% of evaluations (95% CI: [67%, 73%]). A one-sided binomial test confirms this preference is statistically significant far beyond random chance ($p < 10^{-30}$). This result indicates that when accuracy and complexity are controlled, EIC serves as a critical third dimension that effectively captures the structural rationality preferred by humans. To assess the reliability of these subjective judgments, we calculated the inter-rater reliability using Fleiss' Kappa, yielding $\kappa = 0.37$. This indicates a fair level of agreement among experts, suggesting that EIC captures an objective structural quality recognized by humans rather than random individual preferences. Furthermore, Large Language Models (LLMs) acting as domain experts exhibited a similar preference pattern (favoring low-EIC formulas in 72.19% of cases), providing independent corroboration of the human rating results.

5. Conclusion and Discussion

We propose the Effective Information Criterion (EIC), a metric to quantify the structural stability of symbolic regression formulas. EIC measures the amplification of inherent rounding noise during recursive calculation, offering a grounded interpretation from both numerical precision and signal-processing perspectives. Unlike traditional complexity-based metrics, EIC effectively evaluates structural rationality and detects ill-conditioned structures that render formulas physically implausible. Our results demonstrate that integrating EIC can steer search-based SR methods away from irrational formulas to yield superior Pareto frontiers and can boost the sample efficiency of generative SR methods by filtering implausible samples. Furthermore, EIC aligns with human expert preference for interpretability in 70% of cases, validating structural stability as a critical prerequisite for human-perceived interpretability. Despite its effectiveness, several avenues remain for future work. First, integrating EIC guidance with pre-trained model guidance may further boost performance. Second, analyzing EIC at each subformula, rather than only the maximum, may more precisely identify redundant or unreasonable structures, enabling finer-grained evaluation. Finally, in complex systems where simple microscopic rules produce emergent macroscopic behaviors, such as phase transitions, it remains to be studied whether the resulting dynamics formulas retain low EIC scores.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. U23B2030 and 62476152. This work is also supported in part by the Beijing National Research Center for Information Science and Technology (BNRist).

Impact Statement

This paper presents the Effective Information Criterion (EIC), a metric designed to quantify the structural rationality of formulas discovered by symbolic regression algorithms. By prioritizing formulas with structural stability, our work contributes to more reliable AI-driven scientific discovery. We do not foresee any direct negative social impacts from this research.

References

Aldeia, G. S. I., Zhang, H., Bomarito, G., Cranmer, M., Fonseca, A., Burlacu, B., La Cava, W. G., and de França, F. O. Call for action: towards the next generation of symbolic regression benchmark. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '25 Companion, pp. 2529–2538, New York, NY, USA, 2025. Associa-

tion for Computing Machinery. ISBN 9798400714641. doi: 10.1145/3712255.3734309. URL https://doi.org/10.1145/3712255.3734309.

Augusto, D. A. and Barbosa, H. J. Symbolic regression via genetic programming. In *Proceedings. Vol. 1. Sixth Brazilian symposium on neural networks*, pp. 173–178. IEEE, 2000.

Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A., and Parascandolo, G. Neural symbolic regression that scales. In *ICML*, pp. 936–945. PMLR, 2021.

Burlacu, B., Kronberger, G., and Kommenda, M. Operon c++ an efficient genetic programming framework for symbolic regression. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, pp. 1562–1570, 2020.

Camps-Valls, G., Gerhardus, A., Ninad, U., Varando, G., Martius, G., Balaguer-Ballester, E., Vinuesa, R., Diaz, E., Zanna, L., and Runge, J. Discovering causal relations and equations from data. *Physics Reports*, 1044:1–68, 2023.

Cranmer, M. Interpretable machine learning for science with pysr and symbolicregression.jl. *arXiv preprint arXiv:2305.01582*, 2023.

de Franca, F. O., Virgolin, M., Kommenda, M., Majumder, M., Cranmer, M., Espada, G., Ingelse, L., Fonseca, A., Landajuela, M., Petersen, B., et al. Srbench++: principled benchmarking of symbolic regression with domain-expert interpretation. *IEEE transactions on evolutionary computation*, 2024.

Domingos, P. The role of occam's razor in knowledge discovery. *Data mining and knowledge discovery*, 3(4): 409–425, 1999.

Dong, J. and Zhong, J. Recent advances in symbolic regression. *ACM Computing Surveys*, 57(11):1–37, 2025.

Fletcher, R. *Practical methods of optimization*. John Wiley & Sons, 2000.

Fong, K. S. and Motani, M. Pareto-optimal fronts for benchmarking symbolic regression algorithms. In *Forty-second International Conference on Machine Learning*, 2024.

Guimerà, R., Reichardt, I., Aguilar-Mogas, A., Massucci, F. A., Miranda, M., Pallarès, J., and Sales-Pardo, M. A bayesian machine scientist to aid in the solution of challenging scientific problems. *Science advances*, 6(5): eaav6971, 2020.

Kamienny, P.-A., d'Ascoli, S., Lample, G., and Charton, F. End-to-end symbolic regression with transformers. *NeurIPS*, 35:10269–10281, 2022.

Kamienny, P.-A., Lample, G., Lamprier, S., and Virgolin, M. Deep generative symbolic regression with monte-carlo-tree-search. In *ICML*, pp. 15655–15668. PMLR, 2023.

Kommenda, M., Beham, A., Affenzeller, M., and Kronberger, G. Complexity measures for multi-objective symbolic regression. In *International Conference on Computer Aided Systems Theory*, pp. 409–416. Springer, 2015.

La Cava, W., Burlacu, B., Virgolin, M., Kommenda, M., Orzechowski, P., de França, F. O., Jin, Y., and Moore, J. H. Contemporary symbolic regression methods and their relative performance. *Advances in neural information processing systems*, 2021(DB1):1, 2021.

La Cava, W., Burlacu, B., Virgolin, M., Jin, Y., Kommenda, M., and Moore, J. H. The 2022 symbolic regression competition. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1784–1787, 2022.

Lample, G. and Charton, F. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2020.

Liu, S., Li, Q., Shen, X., Sun, J., and Yang, Z. Automated discovery of symbolic laws governing skill acquisition from naturally occurring data. *Nature Computational Science*, pp. 1–12, 2024.

Makke, N. and Chawla, S. Interpretable scientific discovery with symbolic regression: a review. *Artificial Intelligence Review*, 57(1):2, 2024.

Matsubara, Y., Chiba, N., Igarashi, R., and Ushiku, Y. Srsd: Rethinking datasets of symbolic regression for scientific discovery. In *NeurIPS 2022 AI for Science: Progress and Promises*, 2022.

Meidani, K., Shojaee, P., Reddy, C. K., and Farimani, A. B. SNIP: Bridging Mathematical Symbolic and Numeric Realms with Unified Pre-training. In *The Twelfth International Conference on Learning Representations*, October 2023.

Petersen, B. K., Landajuela, M., Mundhenk, T. N., Santiago, C. P., Kim, S. K., and Kim, J. T. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *ICLR*, 2021.

Shojaee, P., Meidani, K., Barati Farimani, A., and Reddy, C. Transformer-based planning for symbolic regression. *Advances in Neural Information Processing Systems*, 36: 45907–45919, 2023.

Shojaee, P., Meidani, K., Gupta, S., Farimani, A. B., and Reddy, C. K. Llm-sr: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Shojaee, P., Nguyen, N.-H., Meidani, K., Farimani, A. B., Doan, K. D., and Reddy, C. K. Llm-srbench: A new benchmark for scientific equation discovery with large language models. In *International Conference on Machine Learning*, pp. 55325–55359. PMLR, 2025b.

Sun, F., Liu, Y., Wang, J.-X., and Sun, H. Symbolic physics learner: Discovering governing equations via monte carlo tree search. In *The Eleventh International Conference on Learning Representations*, 2023.

Tenachi, W., Ibata, R., and Diakogiannis, F. I. Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws. *The Astrophysical Journal*, 959(2):99, 2023.

Udrescu, S.-M. and Tegmark, M. Ai feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Virgolin, M., Burlacu, B., La Cava, W., Derner, E., Kommenda, M., Kronberger, G., Wang, N., Jin, Y., Korns, M., O'Neill, M., et al. The 2023 symbolic regression competition. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 467–470, 2023.

Vladislavleva, E. J., Smits, G. F., and Den Hertog, D. 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.

Widrow, B., Kollar, I., and Liu, M.-C. Statistical theory of quantization. *IEEE Transactions on instrumentation and measurement*, 45(2):353–361, 1996.

Xu, Y., Liu, Y., and Sun, H. Rsrm: Reinforcement symbolic regression machine. *ICLR*, 2024.

Ye, S. and Li, M. Ab initio nonparametric variable selection for scalable symbolic regression with large p. In *International Conference on Machine Learning*, pp. 72041–72062. PMLR, 2025.

Ying, J., Lin, H., Yue, C., Chen, Y., Xiao, C., Shi, Q., Liang, Y., Yau, S.-T., Zhou, Y., and Ma, J. A neural symbolic model for space physics. *Nature Machine Intelligence*, 2025.

Yu, Z., Ding, J., Li, Y., and Jin, D. Symbolic regression via mdlformer-guided search: from minimizing prediction error to minimizing description length. In *The Thirteenth International Conference on Learning Representations*, 2025.

A. Effective Information Criterion (EIC)

A.1. Proof of Proposition 3.2

Proof. First, we show that the cumulated error η_k is a zero-mean random variable with a variance that tends to 0 as $\sigma \to 0$.

Lemma A.1. *Under a first-order approximation, η_k is a zero-mean random variable given by the recursive linearization:*

$$\eta_k \approx \epsilon_k + \sum_{i \in \mathcal{C}[k]} \kappa_{k,i} \times \eta_i, \tag{8}$$

where $\kappa_{k,i} \triangleq \frac{y_i}{y_k}\frac{\partial e_k}{\partial y_i}$ is the partial relative condition number of the operation at node k with respect to input i. The approximation becomes exact in the limit $\sigma \to 0$.

Proof. Let the inputs to node k be $y_1, \ldots, y_i, \ldots, y_m$, which correspond to the outputs of child nodes. For the leaf nodes, we have $\tilde{y}_k = y_k(1+\epsilon_k)$ and $\eta_k = (\tilde{y}_k - y_k)/y_k$, so $\eta_k = \epsilon_k$ for leafe nodes, whose mean is zero and whose variance, σ, tends to zero when $\sigma \to 0$. Then, according to Assumption 3.1, we have

$$\tilde{y}_k = e_k(\tilde{y}_1, \ldots, \tilde{y}_m) \times (1+\epsilon_k) \tag{9}$$

Substituting $\tilde{y}_i$:

$$\tilde{y}_k = e_k(y_1 \times (1+\eta_1), \ldots, y_m \times (1+\eta_m)) \times (1+\epsilon_k) \tag{10}$$

We perform a first-order Taylor expansion of e around the noise-free values $\mathbf{y} = (y_1, \ldots, y_m)$, obtaining

$$e_k(y_1 \times (1+\eta_1), \ldots, y_m \times (1+\eta_m)) \approx e_k(\mathbf{y}) + \sum_{i=1}^{m} \frac{\partial e_k}{\partial y_i}(y_i \eta_i) \tag{11}$$

Substituting this back into the expression for $\tilde{y}_k$, and noting that $y_k = e_k(\mathbf{y})$:

$$\tilde{y}_k \approx \left(y_k + \sum_{i=1}^{m} \frac{\partial e_k}{\partial y_i} y_i \eta_i \right) \times (1+\epsilon_k) \tag{12}$$

Expanding the product and ignoring the second-order error term ($\eta_i \cdot \epsilon_k \ll \min\{\eta_i, \epsilon_k\}$, since $|\eta_i| \ll 1$ and $|\epsilon_k| \ll 1$ when $\sigma \to 0$):

$$\tilde{y}_k \approx y_k + y_k \epsilon_k + \sum_{i=1}^{m} \frac{\partial e_k}{\partial y_i} y_i \eta_i \tag{13}$$

Dividing both sides by y_k, we identify the accumulated relative error $\eta_k \triangleq (\tilde{y}_k - y_k)/y_k$ as:

$$\eta_k \approx \epsilon_k + \sum_{i=1}^{m} \underbrace{\left(\frac{y_i}{y_k} \frac{\partial e_k}{\partial y_i} \right)}_{\kappa_{k,i}} \eta_i \tag{14}$$

Since ϵ_k has zero mean and is independent of η_i (which are linear combinations of zero-mean noise from previous steps and is thus independent of each other), the expectation $\mathbb{E}[\eta_k]$ is a linear combination of zeros, satisfying $\mathbb{E}[\eta_k] = 0$, and its variance $\mathrm{Var}[\eta_k] \to 0$ when $\sigma \to 0$ (since both ϵ_k and $\eta_i \to 0$), which further suggests $|\eta_k| \to 0$ when $\sigma \to 0$. This confirms that the cumulated error η_k is a zero-mean random variable whose variance tends to zero when $\sigma \to 0$. □

Then, we can prove the Eq (3) by applying the conditional variance operator $\mathrm{Var}[\cdot|x]$ to both sides in Eq (8). Two key independence properties apply here:

1. ϵ_k is the intrinsic noise introduced at the current node and is independent of the propagated errors $\{\eta_i\}_{i \in \mathcal{C}[k]}$ from previous steps.
2. Since the formula is structured as a computational tree, the sub-expressions corresponding to different children are disjoint. Thus, the accumulated errors η_i from different children are mutually independent random variables.

Using these independence properties, the variance of the sum becomes the sum of the variances:

$$\mathrm{Var}[\eta_k|x] \approx \mathrm{Var}[\epsilon_k] + \sum_{i \in \mathcal{C}[k]} \kappa_{k,i}^2(x) \mathrm{Var}[\eta_i|x], \tag{15}$$

where the approximation becomes exact in the limit $\sigma \to 0$.

Substituting $\mathrm{Var}[\epsilon_k] = \sigma^2$, dividing both sides by σ^2, and taking the limit $\sigma \to 0$ (which makes the linear approximation exact), we get

$$\lim_{\sigma \to 0} \frac{\mathrm{Var}[\eta_k|x]}{\sigma^2} = 1 + \sum_{i \in \mathcal{C}[k]} \kappa_{k,i}^2(x) \left(\lim_{\sigma \to 0} \frac{\mathrm{Var}[\eta_i|x]}{\sigma^2} \right) \tag{16}$$

which provides the recursive relation to conclude the proof:

$$s_k^2(x) = 1 + \sum_{i \in \mathcal{C}[k]} \kappa_{k,i}^2(x) s_i^2(x) \tag{17}$$

□

A.2. Remark for the definition of EIC

Maximum over internal nodes. In Eq (4), we define EIC as the maximum value of the logarithm $\bar{s}_k$ of all nodes $k \in \mathcal{T}[f]$, rather than the logarithm of the root node $\bar{s}_{\text{root}}$. The choice of the maximum operator is motivated by the non-monotonic nature of error propagation. Observing the recurrence relation in eq (3), if the squared relative condition number is small ($\kappa_{k,i}^2 \ll 1$), the cumulative relative error $s_k(x)$ does not necessarily increase relative to its children's S_i. Physically, a vanishing $\kappa_{k,i}$ implies that the child

node i exerts negligible influence on its parent k. In symbolic regression, this indicates *structural redundancy*, that is, the existence of components that do not contribute to describing the data, which violates the principle of Occam's Razor (Domingos, 1999). Crucially, such a redundant substructure i can be numerically ill-conditioned (exhibiting a very high $s_i(x)$), yet its instability is masked by the small $\kappa_{k,i}$, resulting in a deceptively low $s_{\text{root}}(x)$. By defining EIC as the maximum over all nodes ($\max_k$), we explicitly detect and penalize these *redundant yet ill-conditioned* internal structures, ensuring that a physically plausible formula implies stability not just at the output, but throughout its entire computational graph.

Ordering of expectation and maximum. The ordering of the expectation over x and the maximum over nodes is determined by the interpretation of EIC as rounding-noise amplification inside the computational graph. Since the inherent rounding error is modeled as statistical multiplicative noise, its strength is naturally measured by noise power, i.e., the second moment $\mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)]$ for each fixed subtree. Taking the maximum over k afterwards then identifies the bottleneck node that causes the largest average structural instability. However, if one only considers how to aggregate $s_k^2(x)$ over samples and nodes, alternative orderings such as $\mathbb{E}_{x\sim\mathcal{D}}[\max_{k\in\mathcal{T}[f]} s_k^2(x)]$ or even $\max_x \max_{k\in\mathcal{T}[f]} s_k^2(x)$ may also appear reasonable. The former measures the expected pointwise worst-node instability, but the maximizing node may change with x, producing a blended quantity that no longer corresponds to the noise power of any specific computational substructure. The latter measures the worst instability over both samples and nodes, which is useful for adversarial or corner-case analysis, but is overly sensitive to isolated outliers and does not characterize the overall signal-to-noise degradation over the scientifically relevant distribution $\mathcal{D}$. In contrast, our definition $\max_{k\in\mathcal{T}[f]} \mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)]$ preserves a clear node-wise physical meaning while still imposing a worst-case constraint over the internal computational graph.

Scientifically relevant domain $\mathcal{D}$. In principle, $\mathcal{D}$ should correspond to the scientifically relevant domain over which the discovered formula is expected to be used. In practice, however, this domain is difficult for a symbolic regression algorithm to determine automatically before discovery. Therefore, we use the empirical distribution of the provided dataset as a proxy for the scientifically relevant domain. This is reasonable because a formula discovered by symbolic regression is only constrained by, and hence only directly responsible for, the observed data regime. This limitation also appears in real scientific discovery: without observations near the speed of light, data collected only in the low-velocity regime would naturally support Newtonian approximations such as $p = mv$ or $E_k = \frac{1}{2}mv^2$, rather than Einstein's relativistic formulas. If additional prior knowledge specifies a broader or different scientifically relevant domain, it can be incorporated by evaluating EIC on samples drawn from that user-specified domain.

Difference from conditional error. Finally, the recursive form of $s_k^2(x)$ in Eq. (3) may appear similar to the chain-rule recursion of conditional error. Specifically, the conditional error $p_n(x) = \frac{x_n}{y}\frac{\partial y}{\partial x_n}$ measures how the final output responds to perturbations of an input variable, and by the chain rule, its node-wise form satisfies

$$p_{k,n}(x) = \sum_{i\in\mathcal{C}[k]} \kappa_{k,i}(x) p_{i,n}(x), \tag{18}$$

whose recursive form may appear similar to Eq. (3). However, $s_k^2(x)$ measures how rounding noise is injected and accumulated inside the computational graph, while conditional error is determined only by the input-output mapping and is therefore invariant to the internal computational structure. For example, $y = x$ and $y = \ln(\exp(x))$ have the same conditional error $p(x) \equiv 1$, but EIC can distinguish them because the latter contains an intermediate exponentiation node that amplifies rounding noise. This difference comes from the different recursive forms: Eq. (3) contains a constant 1 that represents newly injected intrinsic rounding noise at each operation node, making EIC accumulate structural penalties throughout the computational tree rather than merely tracking the sensitivity of the realized function.

A.3. Proof of Eq (5)

Proof. Based on the definition of EIC and $\bar{s}_k, s_k(x)$, we have

$$\begin{aligned}
\text{EIC} &\triangleq \max_{k\in\mathcal{T}[f]} \log_{10} \bar{s}_k \\
&\triangleq \max_{k\in\mathcal{T}[f]} \log_{10} \sqrt{\mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)]} \\
&= \frac{1}{2} \max_{k\in\mathcal{T}[f]} \log_{10} \mathbb{E}_{x\sim\mathcal{D}}[s_k^2(x)] \\
&\triangleq \frac{1}{2} \max_{k\in\mathcal{T}[f]} \log_{10} \mathbb{E}_{x\sim\mathcal{D}} \left[\lim_{\sigma\to 0} \frac{\text{Var}[\eta_k|x]}{\sigma^2}\right]
\end{aligned} \tag{19}$$

Here, the limit operator can be sequentially commuted with the expectation, logarithm, and maximum operators due to the linearity of expectation, the continuity of the logarithm, and the finiteness of the node set $\mathcal{T}[f]$, respectively:

$$\begin{aligned}
\text{EIC} &= \frac{1}{2} \max_{k\in\mathcal{T}[f]} \log_{10} \lim_{\sigma\to 0} \mathbb{E}_{x\sim\mathcal{D}} \left[\frac{\text{Var}[\eta_k|x]}{\sigma^2}\right] \\
&= \frac{1}{2} \max_{k\in\mathcal{T}[f]} \lim_{\sigma\to 0} \log_{10} \mathbb{E}_{x\sim\mathcal{D}} \left[\frac{\text{Var}[\eta_k|x]}{\sigma^2}\right] \\
&= \frac{1}{2} \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \mathbb{E}_{x\sim\mathcal{D}} \left[\frac{\text{Var}[\eta_k|x]}{\sigma^2}\right]
\end{aligned} \tag{20}$$

Finally, to merge the conditional variance and expectation into the global variance, consider law of total variance decomposition:

$$\mathbb{E}_x[\text{Var}[\eta_k|x]] = \text{Var}[\eta_k] - \text{Var}_x[\mathbb{E}[\eta_k|x]], \tag{21}$$

where the second term is a higher-order infinitesimal of the first term when $\sigma \to 0$: consider that the conditional mean relative error is $\mathbb{E}[\eta_k|x] \approx 0$. The non-zero bias arises only from second-order terms (e.g., η^2), implying $\mathbb{E}[\eta_k|x] = O(\sigma^2)$. Consequently, the variance of this bias scales as the square of the order:

$$\text{Var}_x[\mathbb{E}[\eta_k|x]] = \text{Var}[O(\sigma^2)] = O(\sigma^4). \tag{22}$$

Substituting this back, the fraction inside the limit becomes:

$$\frac{\mathbb{E}_x[\text{Var}[\eta_k|x]]}{\sigma^2} = \frac{\text{Var}[\eta_k] - O(\sigma^4)}{\sigma^2} = \frac{\text{Var}[\eta_k]}{\sigma^2} - \underbrace{O(\sigma^2)}_{\to 0}. \tag{23}$$

Thus, in the limit $\sigma \to 0$, the bias term vanishes, allowing us to replace the expected conditional variance with the global variance:

$$\mathbb{E}_x[\text{Var}[\eta_k|x]] = \text{Var}[\eta_k] \quad (\sigma \to 0). \tag{24}$$

Substitude this into Eq (20), we finally prove that

$$\begin{aligned} \text{EIC} &= \frac{1}{2} \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\text{Var}[\eta_k]}{\sigma^2} \\ &= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \sqrt{\frac{\text{Var}[\eta_k]}{\sigma^2}} \\ &= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\sqrt{\text{Var}[\eta_k]}}{\sqrt{\text{Var}[\epsilon_k]}} \\ &= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\text{Std}[\eta_k]}{\text{Std}[\epsilon_k]}. \end{aligned} \tag{25}$$

□

A.4. Invariance of EIC Under Unit Transformation

Proof. Here we prove that the EIC is invariant under physical unit transformations (e.g., rescaling input variables from meters to centimeters). This invariance arises from the fact that the partial relative condition number κ is intrinsically dimensionless, rendering the subsequent $s_k(x)$ and EIC metric independent of the chosen system of units.

Specifically, let $[x]$ denote the physical dimension of a variable x (e.g., Length, Time). For any computational operation $y_k = e(\{y_i\}_{i\in\mathcal{C}[k]})$, the partial derivative $\frac{\partial y_k}{\partial y_i}$ represents the rate of change of the output with respect to the input, and thus carries the dimension $[y_k][y_i]^{-1}$. The normalization term $\frac{y_i}{y_k}$ carries the dimension $[y_i][y_k]^{-1}$.

Multiplying these terms to obtain $\kappa_{k,i}$:

$$[\kappa_{k,i}] = \left[\frac{y_i}{y_k}\right] \cdot \left[\frac{\partial y_k}{\partial y_i}\right] = ([y_i][y_k]^{-1}) \cdot ([y_k][y_i]^{-1}) = 1. \tag{26}$$

This confirms that $\kappa_{k,i}$ is a dimensionless quantity (a pure number).

Physical unit transformations (e.g., $x \to \lambda x$) act as linear scalings on the numerical values of dimensional quantities. However, dimensionless quantities are, by definition, invariant to such scaling because the scaling factors in the numerator and denominator cancel out exactly. Thus, the value of $\kappa_{k,i}$ remains constant regardless of the unit system.

As established in Proposition 3.2, the pointwise variance amplification $s_k(x)$ is computed recursively using solely the invariant κ terms and the constant 1. Consequently, both $s_k(x)$ and the final EIC derived from it are invariant under physical unit transformations. □

A.5. Calculation of EIC

Here, we use a simple example to illustrate how EIC is calculated and demonstrate its ability to assess the structural stability of formulas. Specifically, consider the identity $f(x) = x$ versus its mathematically equivalent but redundant form $g(x) = \ln(\exp(x))$. For the concise form $f(x)$, the pointwise amplification is constantly $s_f(x) = 1$. Thus, we have $\bar{s}_f = 1$ and $\text{EIC}(f) = \log_{10} 1 = 0$, suggesting no cumulated information loss during the calculation. For the redundant form $g(x)$, however, the intermediate node $u = \exp(x)$ has a condition number $\kappa_{u,x} = x$, leading to $s_u^2(x) = 1 + x^2$. The final node $y = \ln(u)$ has $\kappa_{y,u} = 1/x$, yielding $s_y^2(x) = 1 + (1/x)^2(1 + x^2) = 2 + x^{-2}$. When inputs x follow a distribution with a large mean (e.g., $x \sim \mathcal{N}(\mu, 1)$ with $|\mu| \gg 1$), the intermediate node u dominates the instability ($\bar{s}_u^2 = \mathbb{E}[s_u^2(x)] = 2 + \mu^2 \gg \bar{s}_y^2 = \mathbb{E}[s_y^2(x)] = 2 + \mathbb{E}[x^{-2}]$), yielding:

$$\text{EIC}(g) = \log_{10} \bar{s}_u = \log_{10} \sqrt{2 + \mu^2} \approx \log_{10} \mu, \tag{27}$$

which is larger than $\text{EIC}(f) = 0$ since $\mu \gg 1$. This result aligns with intuition: when inputs are distributed in a numerically large region, the exponentiation operation induces severe structural instability. Even though the subsequent logarithm restores the value to the correct range, the precision of the intermediate result remains hypersensitive to microscopic perturbations in x. This example highlights that EIC assesses a formula not merely as a functional mapping (Input → Output), but as an *information processing system*. By evaluating the stability of the entire computational graph, EIC discerns the structural rationality of the representation, allowing it to identify the superior, robust realization, even among mathematically equivalent candidates.

This computation process can be formally described by

Algorithm 2 EstimateEIC (Recursively add stochastic perturbation)

Input: Formula f and data distribution $\mathcal{D}$ with N samples.
Parameter: A small enough noise scale σ.
Output: Noisy output $\tilde{y}(x)$ and clean output $y(x)$ at root node, as well as EIC value of f.

$k \leftarrow f$.root % Start from the root node
if k is a Leaf Node (Variable or Constant) **then**
  $y_k(x) \leftarrow$ Evaluate$(k, \mathcal{D})$ % Get values of k in $\mathcal{D}$
  Generate $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ of size N
  $\tilde{y}_k(x) \leftarrow y_k(x) \times (1 + \epsilon_k)$
  $\eta_k \leftarrow \frac{\tilde{y}_k(x) - y_k(x)}{y_k(x)}$ % $\eta_k = \epsilon_k$ for leaf nodes
  $\mathrm{EIC}_k \leftarrow \log_{10} \sqrt{\mathrm{Var}[\eta_k]/\sigma^2}$ % EIC $= 0$ for leaf nodes
  return $(\tilde{y}_k(x), y_k(x), \mathrm{EIC}_k)$
else
  $\mathrm{EIC}_k \leftarrow 0$
  $e_k \leftarrow k$.operator
  for each child $i \in \mathcal{C}[k]$ **do**
    $f' \leftarrow$ Subtree rooted at i
    $(\tilde{y}_i(x), y_i(x), \mathrm{EIC}_i) \leftarrow$ EstimateEIC$(f', \mathcal{D}, \sigma)$
    $\mathrm{EIC}_k \leftarrow \max\{\mathrm{EIC}_k, \mathrm{EIC}_i\}$
  end for
  $y_k(x) \leftarrow e_k(\{y_i\}_{i \in \mathcal{C}[k]})$
  Generate $\epsilon_k \sim \mathcal{N}(0, \sigma^2)$ of size N
  $\tilde{y}_k(x) \leftarrow e_k(\{\tilde{y}_i\}_{i \in \mathcal{C}[k]}) \times (1 + \epsilon_k)$
  $\eta_k \leftarrow \frac{\tilde{y}_k(x) - y_k(x)}{y_k(x)}$
  $\mathrm{EIC}_k \leftarrow \max\{\mathrm{EIC}_k, \log_{10} \sqrt{\mathrm{Var}[\eta_k]/\sigma^2}\}$ % Eq. (5)
  return $(\tilde{y}_k(x), y_k(x), \mathrm{EIC}_k)$
end if

pseudocode Algorithm 1. It is worth noting that although Algorithm 1 requires a differential engine capable of calculating conditional partial derivatives, EIC can be estimated based on Eq (5). Specifically, this estimation strategy chooses a small enough σ and traverses the symbolic tree in post-order, maintaining two parallel computation paths to simulate the noise propagation process: a "clean" reference path computed using standard double-precision arithmetic, and a "noisy" perturbed path where multiplicative noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is injected at every operation node. Consequently, the instability factor $\bar{s}_k$ is estimated via the variance of the relative error between these two paths, which is then mapped to the EIC value according to Eq. (4). While this stochastic approach shares the same time complexity $O(N)$ as the analytical method (where N is the number of nodes), it is significantly easier to integrate into existing evaluation pipelines as it requires only standard forward passes. We detailedly describe this estimation strategy in Algorithm 2.

The stochastic estimation strategy in Algorithm 2 allows EIC to be computed efficiently using standard forward evaluations, which motivates a closer look at its computational cost. For a formula tree with N nodes and a data batch with B samples, both the analytical recursive computation and the stochastic estimation strategy require one post-order traversal of the tree, giving a time complexity of $O(NB)$. When the batch size is fixed, this reduces to $O(N)$ with respect to the formula length. This linear scaling follows from the fact that each node is visited once and only performs a constant number of vectorized operations over the data samples.

In practice, the overhead is small. In our timing experiments with $B = 200$ samples, computing EIC for a typical formula of length 20 takes approximately 2 ms, while a substantially more complex formula of length 80 takes approximately 6 ms. For comparison, evaluating the formula output $y = f(X)$ alone takes about 1 ms and 3 ms for length-20 and length-80 formulas, respectively. Thus, EIC evaluation costs roughly twice a standard forward evaluation. Moreover, the stochastic estimation procedure naturally maintains two parallel computation paths, a clean reference path and a noisy perturbed path, and returns the clean prediction y together with the EIC value. Therefore, when EIC is used during symbolic regression search, the prediction needed for the fitting loss does not need to be recomputed, and the additional overhead is approximately equivalent to one extra forward evaluation of the formula.

A.6. Interpretations for EIC

Numerical Precision Perspective. EIC measures the stability of a formula's computational structure when multiplicative noise of a given intensity is introduced, which, from the perspective of numerical precision, corresponds to a specified number of significant digits.

According to quantization theory, the operation of rounding a value to retain N significant decimal digits is statistically equivalent to injecting a multiplicative noise with variance $\sigma^2 \approx 10^{-2N}$, or $\sigma \approx 10^{-N}$ (Widrow et al., 1996). Based on this equivalence, the injected noise ϵ_k with standard error $\mathrm{Std}[\epsilon_k] = \sigma$ in Assumption 3.1 can be considered as the operation of rounding with a precision of $N \approx -\log_{10}(\mathrm{Std}[\epsilon_k])$ significant digits. [3] On the other hand, the accumulated relative (i.e., multiplicative) error η_k implies that only $M_k \approx -\log_{10}(\mathrm{Std}[\eta_k])$ effective significant digits remain at node k due to the accumulation

[3] Although the rounding error typically follows a Uniform distribution, our derivation relies solely on the second moment. Therefore, the number of digits N (and M_k) discussed here is defined in the *variance-equivalent* sense, mapping the noise intensity to the precision level that would produce such variance regardless of the specific underlying distribution.

of injected noise. Using this relationship, we can formally link EIC to the loss of precision. Specifically, in the high-precision limit ($\sigma \to 0$, corresponding to calculation precision $N \to \infty$), while the effective precision M_k at any node k increases with N indefinitely, it asymptotically lags behind N by a constant margin. We define the maximum inherent precision loss induced by the computational structure as:

$$\Delta_{\max} \triangleq \lim_{N\to\infty} (N - \min_{k\in\mathcal{T}} M_k). \tag{28}$$

which represents the maximum asymptotic gap between the applied computational precision and the effective precision achievable at any node K. Based on the relationship between variance and precision, the EIC equals this loss:

$$\text{EIC} = \Delta_{\max}. \tag{29}$$

This indicates that the EIC measures the structural instability in terms of lost significant digits.

Proof. To proof this, we can substitute the relationship $N \approx -\log_{10}(\text{Std}[\epsilon_k])$ and $M_k \approx -\log_{10}(\text{Std}[\eta_k])$ into Eq (5), obtaining:

$$\begin{aligned}
\text{EIC} &= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\text{Std}[\eta_k]}{\text{Std}[\epsilon_k]} \\
&= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} (\log_{10} \text{Std}[\eta_k] - \log_{10} \text{Std}[\epsilon_k]) \\
&= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}} ((-M_k) - (-N)) \\
&= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}} (N - M_k) \\
&= \lim_{\sigma\to 0} (N - \min_{k\in\mathcal{T}} M_k)
\end{aligned} \tag{30}$$

This completes the proof. □

Signal Processing Perspective. EIC can alternatively be interpreted through the perspective of signal processing. By treating the computational graph as a signal transmission channel, we can analyze how the quality of the "information signal" degrades as it propagates from input to output.

For any node k, we define the "signal" as the noise-free output value y_k, and the "noise" as the deviation caused by the accumulated relative error $\tilde{y}_k - y_k = y_k\eta_k$. The local Signal-to-Noise Ratio (SNR) is defined as the ratio of signal power to noise power:

$$\text{SNR}_k \triangleq \frac{\text{Signal Power}}{\text{Noise Power}} = \frac{y_k^2}{\text{Var}[y_k\eta_k]} = \frac{y_k^2}{y_k^2\text{Var}[\eta_k]} = \frac{1}{\text{Var}[\eta_k]}. \tag{31}$$

Similarly, the intrinsic SNR is.

$$\text{SNR}_{\text{intrinsic}} \triangleq \frac{y_k^2}{\text{Var}[y_k\epsilon_k]} = \frac{1}{\sigma^2} \tag{32}$$

To quantify the degradation of signal quality, we utilize the *Noise Figure* $F_k \triangleq 10\log_{10}\frac{\text{SNR}_{\text{intrinsic}}}{\text{SNR}_k}$, which measures the ratio of input SNR to output SNR.

From this perspective, we can interpret EIC as the noise figure (scaled by $\frac{1}{20}$ at the bottleneck nodes:

$$\text{EIC} = \frac{1}{20} \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} F_k \tag{33}$$

Proof. Begin from the Eq (5), we have

$$\begin{aligned}
\text{EIC} &= \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\text{Std}[\eta_k]}{\text{Std}[\epsilon_k]} \\
&= \frac{1}{2} \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\text{Var}[\eta_k]}{\sigma^2} \\
&= \frac{1}{2} \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} \log_{10} \frac{\text{SNR}_{\text{intrinsic}}}{\text{SNR}_k} \\
&= \frac{1}{20} \lim_{\sigma\to 0} \max_{k\in\mathcal{T}[f]} F_k.
\end{aligned} \tag{34}$$

This completes the proof. □

A.7. EIC in Chaotic Dynamical Systems

It is important to distinguish the numerical instability evaluated by EIC from the physical instability of a dynamical system, especially in chaotic systems. In symbolic regression for dynamical systems, the goal is typically to discover a single-step governing equation, such as $\frac{dx}{dt} = f(x)$. EIC is applied to the algebraic structure of f, measuring whether one evaluation of this expression catastrophically amplifies arithmetic rounding noise through poorly conditioned or redundant computational structures.

By contrast, physical chaos arises from the repeated integration of the governing equation over time, where small perturbations in the state are amplified by the dynamics through a feedback process. This trajectory-level sensitivity is a property of the physical system, not necessarily a flaw in the algebraic expression governing the system. Indeed, many chaotic systems are governed by smooth and numerically well-conditioned equations, such as polynomial terms in the Lorenz system or standard differential operators in fluid dynamics. Such equations can have low EIC even though their long-term trajectories are highly sensitive to initial conditions.

As an empirical illustration, we simulated the Lorenz system,

$$\frac{dx}{dt} = 10(y-x), \quad \frac{dy}{dt} = x(28-z)-y, \quad \frac{dz}{dt} = xy - \frac{8}{3}z, \tag{35}$$

for 4,000 time steps to obtain a chaotic trajectory manifold $\mathcal{D}$. We then computed EIC directly for the three true

governing expressions on this empirical distribution. The resulting EIC values are 0.146, 1.531, and 1.692, respectively, confirming that these chaotic governing equations remain structurally stable under the EIC measure. Therefore, EIC does not penalize physical chaos itself; instead, it filters out numerically unstable candidate formulas so that divergence observed in downstream simulations is more likely to reflect genuine physical dynamics rather than artifacts of an ill-conditioned symbolic representation.

B. Methodology Details

B.1. Enhance Search-based Methods with EIC

To demonstrate that EIC, as a criterion for evaluating unreasonable structures in formulas, can serve as an effective guidance signal for improving symbolic regression, we integrated it into various classical heuristic search algorithms as an auxiliary search objective. Specifically, we focused on two representative approaches: genetic programming (GP) and Monte Carlo tree search (MCTS).

- Genetic programming maintains a population of candidate formulas, where each individual represents a candidate formula. New individuals are generated by crossover or mutation of high-fitness candidates, while low-fitness candidates are eliminated to increase the overall quality of the population. A common fitness function is defined as

$$f(C, \text{NMSE}; \eta) = \eta^C/(1 + \text{NMSE}), \qquad (36)$$

  where *Complexity* measures the structural size of the formula, NMSE denotes the normalized mean squared error (MSE/Var(y)), and $\eta < 1$ is a regularization constant. This formulation penalizes formulas with large errors or excessive complexity, thereby guiding the search toward accurate and compact formulas. It is worth noting that, when calculating the MSE, we first decompose formulas into additive terms and then apply linear regression to fit the coefficients for these terms. This approach integrates the idea of SINDy and effectively improves the performance of algorithms.

- MCTS constructs a search tree where each node represents a candidate formula. Child nodes are those formulas that can be obtained by mutating parent formulas, and their reward values are computed according to Equation (36). In each search iteration, the algorithm starts from the root node and selects a promising leaf node based on its upper confidence bound (UCB) score, which balances average reward and visitation counts. The chosen leaf is then expanded through mutation, and the resulting reward is backpropagated to update the average reward and visitation counts of its ancestors. This process iteratively guides the search toward structurally and numerically promising formulas.

The proposed EIC can be easily incorporated into both algorithms by augmenting their fitness or reward functions. Specifically, we adopt

$$\text{Fitness}_\alpha = \eta^{\text{Complexity}}/(1 + \text{NMSE}) - \alpha \cdot \text{EIC}$$

as the modified fitness and reward functions, where $\alpha > 0$ penalizes formulas with higher EIC, thereby steering the search away from structurally unreasonable solutions. In this work, we use $\eta = 0.999$ in (36) as suggested by Yu et al. (2025) and Sun et al. (2023). For the choice of α, we note that the first term of (36) has a value range of 0 to 1, while, as shown in Figure 2, EIC ranges from 0 to 10. Since EIC serves as an auxiliary search objective, its weight should generally be an order of magnitude smaller than that of the primary objective. Accordingly, we set $\alpha = 0.01$ for MCTS. For the genetic algorithm, we observed that it is more sensitive to the auxiliary objective. Therefore, we chose a smaller value of $\alpha = 0.002$ to optimally balance the algorithm's ability to discover formulas with low EIC while still ensuring high R^2 accuracy.

B.2. Enhance Generative Methods with EIC

To demonstrate that EIC can enhance the sample efficiency and performance of pretraining-based symbolic regression methods, we focus on three representative approaches: E2ESR(Kamienny et al., 2022), SNIP(Meidani et al., 2023), and SR4MDL(Yu et al., 2025), spanning the progression of this research line from earlier efforts to more recent advances in the last year. E2ESR is trained to predict formula tokens directly from (X, y) data pairs; SNIP extends E2ESR with a CLIP-inspired contrastive loss; and SR4MDL further modifies SNIP by changing the prediction target to the minimum description length of formulas to guide the search. All three approaches rely on the formula generation algorithm proposed by Lample & Charton (2020), which generates formulas with random and diverse forms to pretrain generative models.

For each method, we pretrained the model on randomly generated formulas until it achieved the performance reported in the corresponding original papers, and then repeated the pretraining process using a filtered dataset where formulas with EIC > 2.0 are discarded, ensuring that only formulas with reasonable structures were included in pre-training. For E2ESR and SNIP, since they directly predict formula tokens from data, we evaluated their performance by the R^2 scores of the formulas they generated on the Feynman dataset. For SR4MDL, which instead predicts the minimum description length of formulas, we evaluated its performance using the mean absolute error (MAE) and root mean squared error (RMSE) of predicted formula lengths on the Feynman

dataset. This setup allowed us to compare the number of training samples required to reach the same target performance under both settings. All experiments used their official implementations, with hyperparameters selected via a grid search over batch sizes $\in \{32, 64, 128, 256\}$ and learning rates $\in \{10^{-5}, 10^{-4}, 10^{-3}\}$. The hyperparameters were selected under the unfiltered training condition and then kept fixed when training with the filtered dataset.

We also considered a recently proposed data construction strategy baseline, PhyE2E(Ying et al., 2025), which leverages LLMs fine-tuned on physical equations to generate "look-physical" formulas with unit constraints for pretraining generative models. We used the 180k formulas provided by PhyE2E to pretrain the model and compared the results against those obtained with the EIC-filtered dataset. To ensure a fair comparison with our approach and to mitigate the limited size of the PhyE2E corpus, we adopted a hybrid sampling scheme in which, at each training step, a formula was drawn from the PhyE2E dataset with probability 0.1 and from the model's own random generator with probability 0.9.

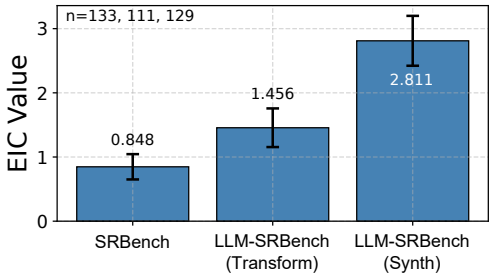


Figure 7. **Average EIC comparison between real physical formulas and LLM-generated formulas.** Error bars denote 95% confidence intervals.

C. Detailed Experimental Results

C.1. Structural Instability of LLM-Generated Formulas

In addition to search-based and generative symbolic regression methods, we further investigate the structural properties of formulas generated by large language models (LLMs) in LLM-SRBench (Shojaee et al., 2025b). This benchmark includes two categories of LLM-generated formulas: *Transform* and *Synth*. The Transform subset is obtained by applying symbolic transformations (e.g., inversion) to real physical formulas, while the Synth subset is constructed via *LLM-assisted synthesis of known and novel terms*, where LLMs combine established scientific components with newly synthesized symbolic terms.

Figure 7 and Figure 8 present the EIC distributions of formulas from SRBench physical formulas, Transform, and Synth subsets, respectively. Both LLM-generated subsets exhibit substantially higher EIC values than real physical formulas from SRBench, indicating a higher degree of structural instability. In particular, formulas in the Synth subset show the highest EIC distribution, suggesting that LLM-synthesized symbolic expressions tend to involve structurally fragile or numerically unstable constructions, despite their surface-level plausibility.

These results lead to three key observations. First, formulas synthesized by LLMs (Synth) exhibit significantly higher EIC values than real physical formulas, indicating that both LLM-generated symbolic expressions and LLM-based symbolic regression methods inherit the challenge of structural instability. Second, formulas derived from physical transfor-

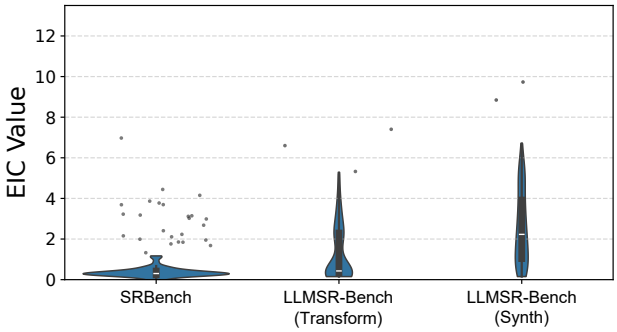


Figure 8. **Violin plots of EIC distributions for SRBench physical formulas, Transform, and Synth subsets.** Scattered points indicate outliers.

mations (Transform) show notably lower EIC than Synth formulas, providing further evidence that symbolic expressions rooted in natural scientific laws differ fundamentally from those synthesized purely by LLMs. Third, LSR-Transform formulas still exhibit higher EIC than the original SRBench physical formulas. We attribute this increase to the fact that some physical equations are not naturally invertible from a causal or physical perspective. When such equations are algebraically inverted to construct LSR-Transform tasks, the resulting expressions may involve divisions by quantities that can become arbitrarily small over the data domain, thereby introducing numerically and structurally unstable forms.

C.2. Definition of Pareto Front Tiers

In Figure 4, we follow the standard practice in symbolic regression and evaluate algorithmic performance using Pareto front tiers, which jointly consider predictive accuracy and formula complexity. This metric can be computed exactly from each algorithm's two-dimensional performance point, where higher accuracy and lower complexity are preferred. Specifically, a method $\mathcal{A}_i$ is said to dominate another method $\mathcal{A}_j$ if $\mathcal{A}_i$ achieves no lower accuracy and no higher complexity than $\mathcal{A}_j$, with a strict improvement in at least one of the two metrics. Tier 1 consists of all non-dominated methods. After removing Tier 1, Tier 2 is formed by the non-dominated methods among the remaining methods, and the same procedure is repeated to obtain lower tiers. This non-dominated sorting procedure is deterministic given the reported accuracy and complexity values, and is only related with the relative performance of methods but not the absolute values of accuracy and complexity.

C.3. Enhance Search-based Methods with EIC

Figure 9 illustrates the distribution of Effective Information Criterion (EIC) values for formulas discovered by all benchmarked methods. The left panel depicts the distributions on white-box problems across varying noise levels. Notably, the integration of EIC consistently drives both MCTS and GP toward lower EIC values, aligning them more closely with the distribution of ground-truth physical formulas compared to their vanilla versions. This confirms that EIC effectively suppresses structurally unreasonable and numerically unstable patterns. The right panel presents the results on black-box problems. Here, incorporating EIC leads to a substantial reduction in the prevalence of high-EIC structures for both MCTS and GP. Crucially, this reduction is more pronounced compared to the white-box setting, suggesting that in the absence of a known underlying physical law, standard SR methods are prone to overfitting via overly complex and unstable functional forms. EIC effectively mitigates this tendency, enforcing structural plausibility even when the ground truth is unknown.

Corresponding to Figures 4 (in the main text) as well as Figure 9, we provide the raw experimental results for 17 baseline methods and our four methods under four noise levels on white-box data and on black-box data, as shown in Tables 3, 4, 5, 6, and 7, respectively. The tables report formula accuracy, complexity, running duration, and the average EIC of the resulting formulas. It is worth noting that some methods such as PySR are configured to utilize multiprocessing by default, which can lead to significantly shorter runtimes compared to single-core implementations. To ensure a fair comparison of runtimes across all methods, we report the duration under a single-core setting for all methods.

C.4. Comparison with Order of Nonlinearity

To test whether the benefit of EIC comes merely from adding another orthogonal complexity measure, we replaced EIC with Order of Nonlinearity (ON) in the same search framework and repeated the white-box experiment under the same setup as Figure 4. ON encourages formulas that can be approximated by low-degree polynomials, but this criterion does not directly capture the rounding-noise stability of a computational graph. As shown in Table 8, incorporating EIC improves both GP and MCTS in terms of recovery rate and complexity, whereas incorporating ON mainly reduces complexity at the cost of lower recovery rate. This suggests that EIC's practical value comes from its physical definition based on structural stability, not simply from adding an additional complexity objective.

C.5. Enhance Generative Methods with EIC

We demonstrate that EIC improves the alignment between pre-training formulas and real-world physical formulas by measuring the similarity between randomly generated formulas and EIC-filtered formulas with three datasets of real physical formulas. These datasets include the Feynman dataset, the Strogatz ODE dataset, and the Wiki Named Equation dataset. The first two are derived from the white-box models in SRBench, while the third was collected by Guimerà et al. (2020) from Wikipedia, containing over a thousand named formulas. We cleaned and deduplicated the formulas, removing those with special operators (e.g., matrix operations) and inherently redundant formulas (e.g., numerous occurrences of simple expressions like $a \times b$, retaining only one). The final set contains 984 physical formulas.

In addition to the training process we reported in Figure 5, we also provide the training process of SNIP and SR4MDL in Figure 10 and Figure 11. The SNIP model trained on random formulas reaches a final R^2 performance of 0.5299, while it trained on EIC-filtered formulas reaches a final R^2 of 0.6016, which is 13.5% higher. For the SR4MDL,

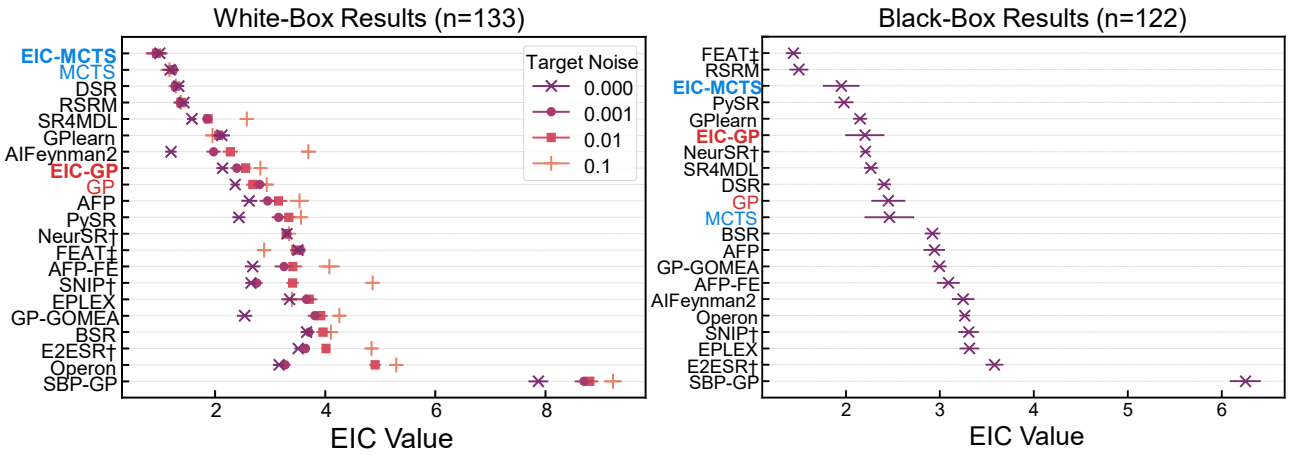


Figure 9. **EIC distributions of formulas discovered by baseline methods and our EIC-enhanced variants.** The left panel presents results on 133 white-box problems under varying noise levels ($\{0, 10^{-3}, 10^{-2}, 10^{-1}\}$), while the right panel presents results on 122 black-box problems. The markers represent the mean EIC values across all problems, and error bars indicate 95% confidence intervals. Notably, the EIC-enhanced methods (highlighted in bold color) consistently exhibit lower EIC values compared to their vanilla counterparts, demonstrating that incorporating EIC effectively steers the search toward structurally rational formulas.

Table 3. Whitebox results at noise-free condition

Type	algorithm	$R^2 > 0.99 \uparrow$	complexity↓	duration↓	EIC↓
Regression	**FEAT**	0.621(±0.031)	195.3(±8.5)	2269(±1.9e+02)	3.504(±0.22)
Generative	**E2ESR**	0.2773(±0.03)	89.63(±2.1)	4.024(±0.14)	3.509(±0.066)
	NeurSR	0.07681(±0.014)	31.42(±0.26)	23.19(±0.39)	3.307(±0.049)
	SNIP	0.1541(±0.019)	25.2(±0.39)	1.845(±0.07)	2.652(±0.1)
Search	**AFP**	0.551(±0.03)	37.01(±1.2)	3282(±2.4e+02)	2.622(±0.15)
	AFP-FE	0.7154(±0.025)	40.63(±1.3)	17090(±750)	2.684(±0.14)
	AIFeynman2	0.8487(±0.02)	113.2(±21)	844.2(±1.9e+02)	1.2(±0.2)
	BSR	0.249(±0.026)	26.95(±0.62)	29310(±790)	3.663(±0.11)
	DSR	0.3365(±0.029)	14.94(±0.49)	1746(±1.8e+02)	1.343(±0.1)
	EPLEX	0.7375(±0.027)	52.64(±1.1)	11450(±690)	3.355(±0.15)
	GP-GOMEA	0.8808(±0.02)	34.77(±0.97)	4786(±4.4e+02)	2.537(±0.14)
	GPlearn	0.468(±0.03)	67.71(±16)	3606(±3.6e+02)	2.126(±0.15)
	Operon	0.9392(±0.013)	68.73(±1.4)	1947(±64)	3.162(±0.11)
	PySR	0.6767(±0.064)	9.349(±0.46)	4244(±44)	2.586(±0.2)
	RSRM	0.2455(±0.025)	13.4(±0.37)	116.9(±3)	1.435(±0.092)
	SBP-GP	0.9365(±0.015)	513.4(±12)	2.768e+04(±2.4e+02)	7.872(±0.34)
	SR4MDL	0.6271(±0.026)	21.7(±0.79)	470(±34)	1.58(±0.086)
	GP	0.7215(±0.029)	34.61(±1.7)	1728(±67)	2.367(±0.12)
	EIC-GP	0.7288(±0.038)	32.47(±2.2)	1718(±90)	2.136(±0.15)
	$\Delta(\%)$	1%	-7%	-0.50%	-9.80%
	MCTS	0.6917(±0.046)	17.68(±1.2)	10580(±1000)	1.185(±0.14)
	EIC-MCTS	0.7107(±0.045)	17.64(±1.2)	12080(±1.1e+03)	0.9983(±0.12)
	$\Delta(\%)$	2.70%	-0.20%	14%	-16%

Table 4. Whitebox Results at 0.001 noise

Type	algorithm	$R^2 > 0.99 \uparrow$	complexity↓	duration↓	EIC↓
Regression	**FEAT**	0.6236 (±0.03)	186.4 (±7.2)	1598 (±89)	3.556 (±0.18)
Generative	**E2ESR**	0.2595 (±0.03)	89.56 (±2.1)	3.846 (±0.13)	3.641 (±0.07)
	NeurSR	0.0767 (±0.01)	31.34 (±0.26)	23.52 (±0.39)	3.309 (±0.05)
	SNIP	0.1579 (±0.02)	25.2 (±0.39)	1.865 (±0.07)	2.759 (±0.11)
Search	**AFP**	0.5646 (±0.03)	39.27 (±1)	3327 (±104)	2.954 (±0.14)
	AFP-FE	0.7308 (±0.02)	46.71 (±1.1)	24216 (±511)	3.252 (±0.17)
	AIFeynman2	0.8305 (±0.02)	118.7 (±20)	575.1 (±27)	1.974 (±0.25)
	BSR	0.2515 (±0.02)	27.12 (±0.56)	29613 (±1691)	3.709 (±0.1)
	DSR	0.3815 (±0.03)	16.32 (±0.47)	794.6 (±27)	1.273 (±0.09)
	EPLEX	0.77 (±0.02)	55.3 (±0.8)	11057 (±318)	3.66 (±0.14)
	GP-GOMEA	0.9038 (±0.02)	44.79 (±0.77)	2678 (±197)	3.811 (±0.13)
	GPlearn	0.4809 (±0.03)	57.12 (±8.4)	3054 (±234)	2.093 (±0.13)
	Operon	0.9538 (±0.01)	69.39 (±1.4)	1967 (±70)	3.276 (±0.10)
	PySR	0.6083 (±0.07)	9.930 (±0.5)	4321 (±34)	3.434 (±0.23)
	RSRM	0.263 (±0.03)	13.22 (±0.47)	128.3 (±2.5)	1.367 (±0.12)
	SBP-GP	0.9362 (±0.01)	600.9 (±9.4)	27934 (±172)	8.697 (±0.32)
	SR4MDL	0.6099 (±0.03)	26.8 (±0.65)	540.5 (±28)	1.855 (±0.08)
	GP	0.691 (±0.05)	35.65 (±2.3)	1813 (±103)	2.811 (±0.16)
	EIC-GP	0.7103 (±0.04)	31.7 (±2.2)	1703 (±103)	2.394 (±0.16)
	$\Delta(\%)$	+3%	-11%	-6.08%	-14.82%
	MCTS	0.6404 (±0.03)	19.3 (±0.6)	3641 (±469)	1.246 (±0.09)
	EIC-MCTS	0.6692 (±0.08)	20.46 (±1.8)	15228 (±2045)	0.9275 (±0.18)
	$\Delta(\%)$	+4.49%	+6.03%	+318%	-26%

Table 5. Whitebox Results at 0.01 noise

Type	algorithm	$R^2 > 0.99 \uparrow$	complexity↓	duration↓	EIC↓
Regression	**FEAT**	0.6198 (±0.03)	159.3 (±6.2)	1357 (±73)	3.464 (±0.2)
Generative	**E2ESR**	0.2388 (±0.03)	97.61 (±2.3)	4.175 (±0.1)	4.015 (±0.08)
	NeurSR	0.0759 (±0.01)	31.44 (±0.25)	23.66 (±0.38)	3.292 (±0.04)
	SNIP	0.1414 (±0.02)	27.05 (±0.37)	2.109 (±0.08)	3.407 (±0.11)
Search	**AFP**	0.5808 (±0.03)	40.62 (±1)	3666 (±118)	3.152 (±0.16)
	AFP-FE	0.7262 (±0.02)	47.12 (±1.1)	25731 (±414)	3.413 (±0.17)
	AIFeynman2	0.797 (±0.02)	140.2 (±23)	562.8 (±28)	2.276 (±0.27)
	BSR	0.2685 (±0.02)	29.19 (±0.71)	29680 (±1554)	3.961 (±0.12)
	DSR	0.3838 (±0.03)	16.45 (±0.48)	882.8 (±33)	1.289 (±0.09)
	EPLEX	0.7792 (±0.02)	53.9 (±0.82)	9901 (±240)	3.711 (±0.15)
	GP-GOMEA	0.9085 (±0.02)	44.46 (±0.72)	2777 (±197)	3.919 (±0.13)
	GPlearn	0.4813 (±0.03)	56.85 (±11)	3084 (±235)	2.064 (±0.13)
	Operon	0.9438 (±0.01)	87.29 (±0.42)	2835 (±80)	4.908 (±0.1)
	PySR	0.5068 (±0.07)	9.094 (±0.5)	4236 (±22)	3.575 (±0.22)
	RSRM	0.2703 (±0.03)	13.38 (±0.37)	132.7 (±2.3)	1.384 (±0.09)
	SBP-GP	0.9338 (±0.01)	623 (±9.2)	28074 (±154)	8.804 (±0.31)
	SR4MDL	0.6084 (±0.03)	26.6 (±0.64)	805.6 (±41)	1.876 (±0.08)
	GP	0.6642 (±0.05)	34.83 (±2)	1784 (±100)	2.684 (±0.16)
	EIC-GP	0.6884 (±0.05)	33.65 (±2.2)	1582 (±104)	2.554 (±0.16)
	$\Delta(\%)$	+4%	-3%	-11.31%	-4.84%
	MCTS	0.6186 (±0.03)	19.33 (±0.63)	4158 (±464)	1.226 (±0.09)
	EIC-MCTS	0.7083 (±0.08)	18.99 (±1.8)	15996 (±1928)	0.9407 (±0.2)
	$\Delta(\%)$	+14.51%	-1.77%	+285%	-23%

Table 6. Whitebox Results at 0.1 noise

Type	algorithm	$R^2 > 0.99 \uparrow$	complexity↓	duration↓	EIC↓
Regression	**FEAT**	0.4606 (±0.03)	97.84 (±3.8)	742.3 (±32)	2.891 (±0.17)
Generative	**E2ESR**	0.0176 (±0.01)	103.3 (±2.1)	6.119 (±0.34)	4.841 (±0.1)
	NeurSR	0.0421 (±0.01)	31.83 (±0.24)	24.93 (±0.39)	3.344 (±0.04)
	SNIP	0.0346 (±0.01)	30.95 (±0.35)	2.655 (±0.1)	4.862 (±0.11)
Search	**AFP**	0.5485 (±0.03)	41.18 (±0.99)	3485 (±95)	3.535 (±0.17)
	AFP-FE	0.7262 (±0.02)	49.1 (±1)	26795 (±319)	4.077 (±0.19)
	AIFeynman2	0.1949 (±0.02)	157.3 (±21)	629.5 (±91)	3.695 (±0.26)
	BSR	0.22 (±0.02)	31.01 (±0.84)	31506 (±2521)	4.104 (±0.11)
	DSR	0.3823 (±0.03)	16.3 (±0.47)	781 (±24)	1.288 (±0.09)
	EPLEX	0.7585 (±0.02)	46.46 (±0.95)	9219 (±192)	3.394 (±0.15)
	GP-GOMEA	0.8885 (±0.02)	46.13 (±0.7)	2886 (±207)	4.258 (±0.12)
	GPlearn	0.4786 (±0.03)	46.32 (±6.3)	2715 (±209)	1.952 (±0.13)
	Operon	0.8892 (±0.02)	88.61 (±0.32)	2768 (±71)	5.29 (±0.1)
	PySR	0.0722 (±0.08)	10.29 (±0.78)	4366 (±17)	3.824 (±0.23)
	RSRM	0.2543 (±0.03)	13.11 (±0.34)	134.2 (±2.1)	1.379 (±0.09)
	SBP-GP	0.8662 (±0.02)	652 (±8.9)	28173 (±140)	9.227 (±0.31)
	SR4MDL	0.4717 (±0.03)	29.81 (±0.59)	833.9 (±44)	2.577 (±0.07)
	GP	0.3258 (±0.05)	29.67 (±1.7)	1157 (±96)	2.938 (±0.15)
	EIC-GP	0.3182 (±0.05)	29.97 (±1.9)	861.7 (±91)	2.823 (±0.14)
	$\Delta(\%)$	-2%	+1%	-25.51%	-3.92%
	MCTS	0.5064 (±0.06)	18.77 (±1.1)	2778 (±325)	1.174 (±0.17)
	EIC-MCTS	0.5012 (±0.06)	18.67 (±1.3)	3005 (±403)	1.092 (±0.16)
	$\Delta(\%)$	-1%	+0.5%	+8.17%	-6.98%

Table 7. blackbox results

Type	algorithm	$R^2 \uparrow$	complexity ↓	EIC ↓
Regression	**FEAT**	0.7621 (±0.01)	82.49 (±3.3)	1.441 (±0.09)
Generative	**E2ESR**	0.3612 (±0.02)	61.09 (±1)	3.581 (±0.13)
	NeurSR	0.1228 (±0.01)	13.33 (±0.12)	2.208 (±0.07)
	SNIP	0.3335 (±0.02)	38.91 (±0.55)	3.307 (±0.14)
Search	**AFP**	0.6333 (±0.01)	34.89 (±1)	2.941 (±0.12)
	AFP-FE	0.64 (±0.01)	36.04 (±1)	3.091 (±0.12)
	AIFeynman2	0.211 (±0.02)	2240 (±250)	3.248 (±0.14)
	BSR	0.2725 (±0.02)	22.52 (±0.91)	2.92 (±0.08)
	DSR	0.5625 (±0.01)	9.465 (±0.26)	2.408 (±0.07)
	EPLEX	0.7372 (±0.01)	53.14 (±0.73)	3.315 (±0.11)
	GP-GOMEA	0.7381 (±0.01)	30.27 (±0.96)	2.996 (±0.08)
	GPlearn	0.539 (±0.01)	19.06 (±0.96)	2.151 (±0.08)
	Operon	0.7945 (±0.02)	65.69 (±1.3)	3.264 (±0.06)
	SBP-GP	0.7869 (±0.01)	634 (±18)	6.252 (±0.22)
	SR4MDL	0.6258 (±0.01)	29.88 (±0.56)	2.267 (±0.10)
	GP	0.6881 (±0.04)	34.84 (±2.4)	2.452 (±0.18)
	EIC-GP	0.6433 (±0.05)	27.48 (±2.1)	2.202 (±0.21)
	$\Delta(\%)$	-6.50%	-26.00%	-10%
	MCTS	0.6038 (±0.06)	35.91 (±1.9)	2.463 (±0.27)
	EIC-MCTS	0.6227 (±0.05)	33.89 (±1.9)	1.951 (±0.2)
	$\Delta(\%)$	+3.13%	-5.62%	-20.80%

Table 8. **Comparison between EIC and Order of Nonlinearity (ON) as auxiliary objectives.** The setup follows the white-box experiment in Figure 4.

Alg.	$R^2 > 0.99 \uparrow$	Complexity ↓
GP	0.7215	34.61
EIC-GP	0.7288	32.47
ON-GP	0.7117	30.28
MCTS	0.6917	17.68
EIC-MCTS	0.7107	17.64
ON-MCTS	0.6728	15.89

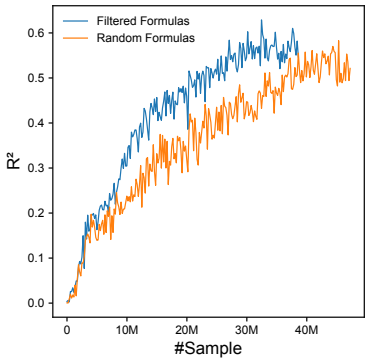


Figure 10. Training process of SNIP using both random and filtered formulas

trained on random formulas, it reaches final RMSE and MAE of 8.6778 and 6.9254, respectively. While trained on EIC-filtered formulas, it can reach final RMSE and MAE of 8.2319 and 6.5763, which are 5.14% and 5.04% higher, respectively.

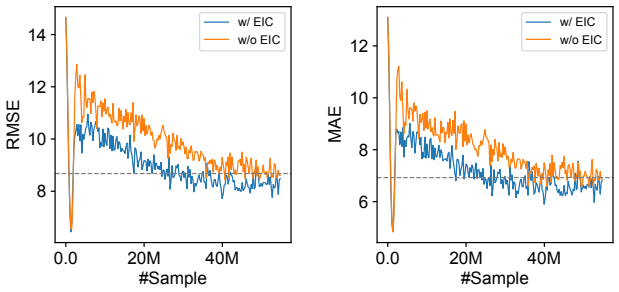


Figure 11. Training process of SR4MDL using both random and filtered formulas

C.6. Expert Evaluation Results

Generation of Formula Pairs for Human Evaluation. To rigorously validate the alignment between EIC and human interpretability preferences, we constructed a dataset of paired formulas by strictly controlling for accuracy and complexity. Specifically, we aggregated formulas discovered by search-based symbolic regression methods in Section 4.1, keeping those one- and two-dimensional black-box problems in SRBench (in total of 19 problems). This resulted in a comprehensive candidate pool covering diverse structural patterns. Then, from this aggregated pool, we selected formula pairs to maximize structural divergence (EIC difference) while maintaining strict performance parity. This was formulated as minimizing the following distance function:

$$L(f_1, f_2) = (\text{Complexity}[f_1] - \text{Complexity}[f_2])^2 + (R^2[f_1] - R^2[f_2])^2 - (\text{EIC}[f_1] - \text{EIC}[f_2])^2, \quad (37)$$

subject to the following constraints to ensure the condition that all else being equal:

$$\begin{aligned} &|\text{Complexity}[f_1] - \text{Complexity}[f_2]| \leq 2, \\ &|R^2[f_1] - R^2[f_2]| \leq 0.02, \\ &|\text{EIC}[f_1] - \text{EIC}[f_2]| \geq 2, \\ &\max(R^2[f_1], R^2[f_2]) > 0.85. \end{aligned} \quad (38)$$

Here, f_1 and f_2 represent any two distinct formulas selected from the aggregated pool. The negative sign for the EIC term in L drives the selection toward pairs with the largest possible stability gap, while the constraints ensure that the paired formulas are indistinguishable in terms of fitting accuracy and length. The threshold $R^2 > 0.85$ ensures that only high-quality candidates are evaluated. This procedure yielded a total of 172 valid formula pairs across the 19 problems.

Human Expert Evaluation. For the human rating experiment, we invited 108 volunteer participants with at least a bachelor's degree in science or engineering. Each participant was randomly assigned 10 formula pairs. For each pair, participants were presented with the data sample visualization, dataset description, and the two candidate formulas alongside their R^2 and complexity metrics. They were then asked to choose the more interpretable one based on their domain knowledge and intuition (importantly, EIC scores were withheld to ensure blind evaluation), as demonstrated in Figure 13 and 14. We collected 1080 evaluations in total. After discarding responses completed in under 60 seconds to ensure quality, 840 valid evaluations remained, with each pair assessed on average 4.9 times.

LLM Corroboration. To provide independent corroboration of the human evaluation results, we also presented each pair to Large Language Models (LLMs) acting as domain experts. We employed two models, GPT-4o-Mini and Qwen3, which are trained on different linguistic distributions, thereby reducing the risk of model-specific bias. We formatted the prompt as shown in Figure 15, providing the LLMs with the dataset context, sampled points (x, y), mathematical expressions, and performance metrics (R^2, Complexity), while withholding EIC scores. We evaluated every formula pair on each LLM five times under a temperature setting of 0.7 to account for sampling variability. The results, shown in Figure 12, indicate that LLMs exhibited a preference rate of 72.19% for the lower-EIC formulas. This alignment with human experts (who preferred lower-EIC formulas in 69.8% of cases) further validates the reliability of EIC as a proxy for interpretability.

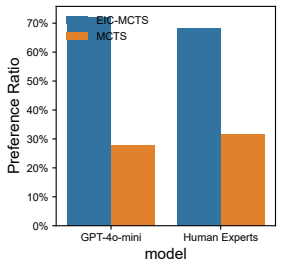


Figure 12. Results of LLM and human experts' preferences for formula interpretability.

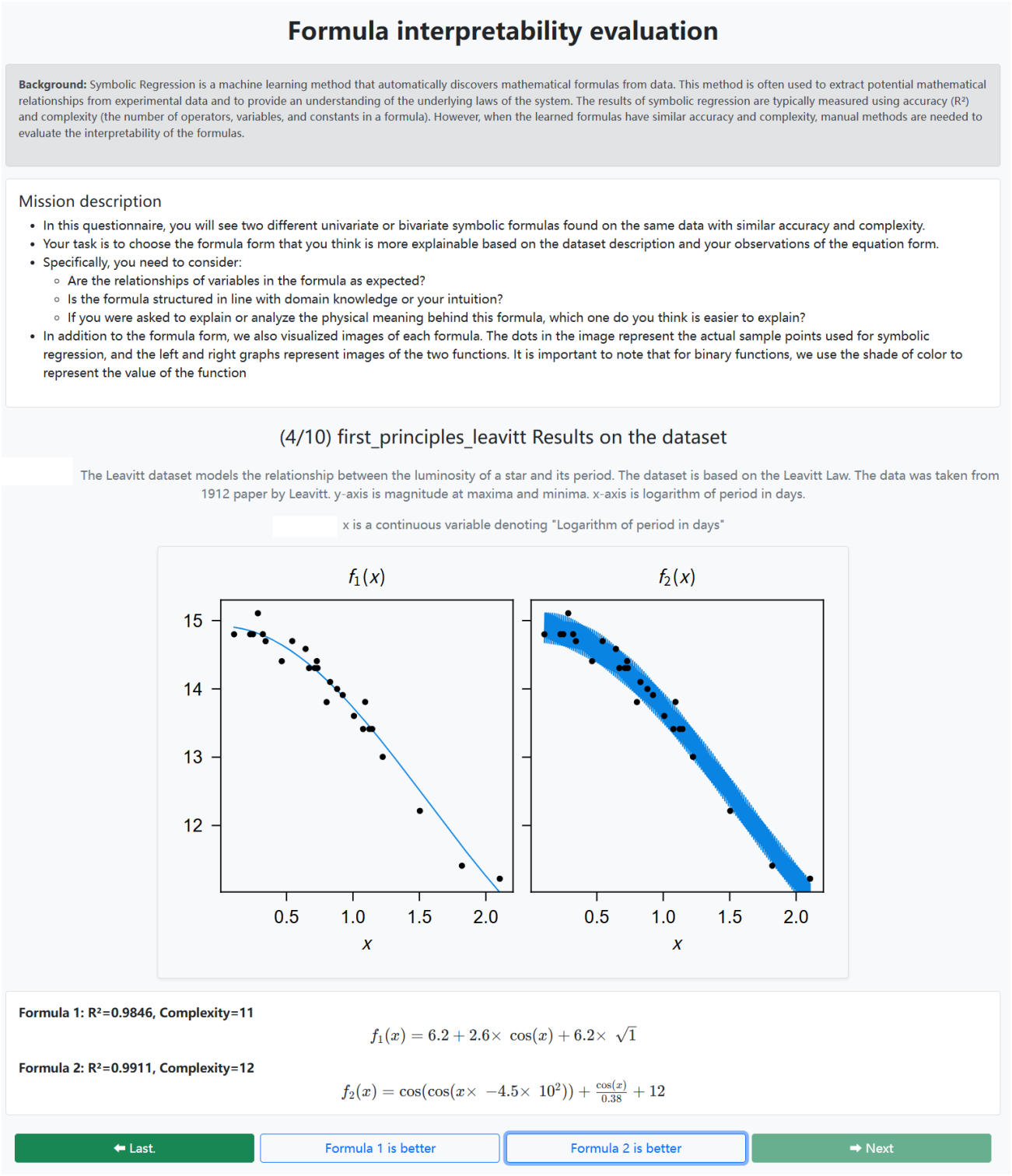


Figure 13. Website interface used by human experts for scoring, where a one-dimensional dataset is demonstrated

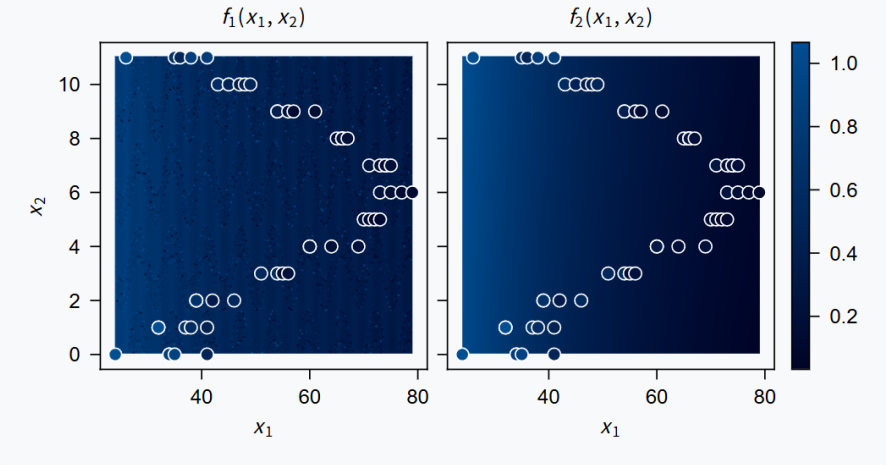
Formula interpretability evaluation

Background: Symbolic Regression is a machine learning method that automatically discovers mathematical formulas from data. This method is often used to extract potential mathematical relationships from experimental data and to provide an understanding of the underlying laws of the system. The results of symbolic regression are typically measured using accuracy (R^2) and complexity (the number of operators, variables, and constants in a formula). However, when the learned formulas have similar accuracy and complexity, manual methods are needed to evaluate the interpretability of the formulas.

Mission description

- In this questionnaire, you will see two different univariate or bivariate symbolic formulas found on the same data with similar accuracy and complexity.
- Your task is to choose the formula form that you think is more explainable based on the dataset description and your observations of the equation form.
- Specifically, you need to consider:
  - Are the relationships of variables in the formula as expected?
  - Is the formula structured in line with domain knowledge or your intuition?
  - If you were asked to explain or analyze the physical meaning behind this formula, which one do you think is easier to explain?
- In addition to the formula form, we also visualized images of each formula. The dots in the image represent the actual sample points used for symbolic regression, and the left and right graphs represent images of the two functions. It is important to note that for binary functions, we use the shade of color to represent the value of the function

(3/10) 228_elusage Results on the dataset



Formula 1: R^2=0.9060, Complexity=23

$$f_1(x_1, x_2) = \frac{\sin(\frac{x_1}{0.45}) \times 2.9 + -13 + \frac{1}{\cos(\cot(x_2 \cdot \cos(x_1)))}}{0.51} + \frac{1}{x_1} \times 3.2 \times 10^3$$

Formula 2: R^2=0.9084, Complexity=23

$$f_2(x_1, x_2) = -43 + -1.4 \times 10^5 \times (\frac{1}{x_1 \times 2})^2 + 0.6x_2 + 9.9 \times 10^3 \times \mathrm{Fishy}(\frac{1}{x_1 \times 2})$$

⬅ Last. | Formula 1 is better | Formula 2 is better | ➡ Next

Figure 14. Website interface used by human experts for scoring, where a two-dimensional dataset is demonstrated

```
Assume you're an expert. Your have derived two expressions describing a absorption dataset you're
studying using symbolic regression. They have similar R² and formula lengths, making them
indistinguishable by accuracy or complexity. Therefore, you need to analyze the two equations based
on your domain knowledge, and choose a formula you prefer.

- **Dataset Description**: A real-world dataset containing the absorption of light for a solution
containing a specific molecule at different levels of concentration. The original publication has
data for 4 different molecules, and here we include data from only one of them (`real_data/
absorption/examples/example0.csv`), the one with highest number of samples.

- **Variable Description**: a continuous variable `Xaxis0` denoting the Concentration (mol/L).
- **Data Samples (x, y)**: [(0.2,0.062),(0.5,0.22),(1,0.25),(1.5,0.38),(2.3,0.53),(3,0.62),(4,0.85),
(5,1),(6,1.2),(7.2,1.5),(10,2),(15,2.8),(30,3.3),(45,3.4)]
- **Result 1**:
    - R2: 0.9541
    - Formula Length: 5
    - Equation: log(Xaxis0 - log(Xaxis0))
- **Result 2**:
    - R2: 0.9436
    - Formula Length: 6
    - Equation: -0.265 + 0.623 * sqrt(Xaxis0)

Your response should end with a single line indicating your choice: either \"I prefer Equation 1\"
or "I prefer Equation 2".
```

Figure 15. Prompt we used to ask LLM to act as domain experts to evaluate formula interpretability.