nature communications

Article

https://doi.org/10.1038/s41467-025-59288-y

Interactive symbolic regression with co-design mechanism through offline reinforcement learning

Received: 20 August 2024

Accepted: 17 April 2025

Published online: 26 April 2025

Check for updates

Yuan Tian[1,2], Wenqi Zhou[1,3], Michele Viscione[3], Hao Dong[3,4], David S. Kammer[1] & Olga Fink[3]

Symbolic Regression holds great potential for uncovering underlying mathematical and physical relationships from observed data. However, the vast combinatorial space of possible expressions poses significant challenges for previous online search methods and pre-trained transformer models, which mostly do not consider the integration of domain experts' prior knowledge. To address these challenges, we propose the Symbolic Q-network, an advanced interactive framework for large-scale symbolic regression. Unlike previous transformer-based SR approaches, Symbolic Q-network leverages reinforcement learning without relying on a transformer-based decoder. Furthermore, we propose a co-design mechanism, where the Symbolic Q-network facilitates effective interaction with domain experts at any stage of the equation discovery process. Our extensive experiments demonstrate Sym-Q performs comparably to existing pretrained models across multiple benchmarks. Furthermore, our experiments on real-world cases demonstrate that the interactive co-design mechanism significantly enhances Symbolic Q-network's performance, achieving greater performance gains than standard autoregressive models.

Symbolic regression is a powerful form of regression analysis that searches the space of mathematical expressions to find the expression that best fits an observed dataset. Unlike traditional regression models that fit data to pre-specified equations, symbolic regression can discover the underlying equations or relationships between variables. This capability can lead to a deeper understanding of the inherent structure and dynamics of the underlying processes. It is particularly important in fields where the relationships between variables are complex and not well understood, as it provides a tool to uncover the form of the relationship without prior assumptions. Recently, symbolic regression has been instrumental in uncovering new relationships, such as astrophysical scaling relations[1] and analytical models of exoplanet transit spectroscopy[2]. However, a significant challenge in symbolic regression is its inherent combinatorial complexity. This complexity grows with the length of the symbolic expressions, making it a computationally demanding NP-hard problem[3,4].

To address this challenge, researchers have made significant advancements in symbolic regression, mainly in two methodological directions: online search techniques and transformer-based models trained on large-scale datasets. Online search methods aim to identify mathematical expressions that best describe the data by efficiently exploring the solution space. One of the most widely used methods in online search for symbolic regression is Genetic Programming (GP)[5–8]. GP iteratively evolves successive generations of mathematical expressions to approximate observed data by applying operations such as selection, crossover, and mutation. Another effective method in online search is reinforcement learning (RL), which takes a different approach to optimizing expressions by learning policies that guide the search process. For example, Petersen et al.[9], introduced Deep

[1]Institute for Building Materials, ETH Zürich, Zürich, Switzerland. [2]China Yangtze Power Co., Ltd, Yichang, Hubei, China. [3]Laboratory of Intelligent Maintenance and Operations Systems, EPFL, Lausanne, Switzerland. [4]Institute of Structural Engineering (IBK), ETH Zürich, Zürich, Switzerland. e-mail: olga.fink@epfl.ch

Symbolic Regression (DSR), which utilizes a risk-seeking policy gradient to optimize expressions using the normalized root-mean-square error as a reward signal. While promising, this approach has a significant drawback: it requires searching for every expression from scratch, making it computationally expensive and time-consuming. Moreover, combining GP and RL can leverage the strengths of both strategies. Mundhenk et al.[10] introduced a neural-guided component to initialize the starting population for a random-restart GP process. This hybrid approach enables the model to progressively learn and refine better starting populations, resulting in significant performance improvements over traditional methods.

However, these online search methods often require training a new model for each specific expression, resulting in limited generalization and substantial computational demands. To address these limitations, recent research has shifted focus toward transformer decoder-based models[3,11–14] to construct the output expression tree. Recent studies investigated different points: encoder[3], contrastive loss[3], End-to-End symbolic regression[12], and fusion strategies[13,14]. These models leverage powerful attention mechanisms and are trained on large-scale datasets, enabling them to effectively learn and understand different patterns. This allows them to autonomously generate plausible mathematical expressions in an autoregressive manner, dynamically adapting to the input data. Transformer-based approaches also excel at capturing long-range dependencies within data, which is crucial for tackling complex symbolic regression tasks. Recently, pretrained transformer models have been utilized to enhance the online search process by integrating advanced techniques such as Monte Carlo Tree Search[15] and Monte Carlo Sampling[16] into the pipeline, further improving efficiency and performance. Among these hybrid approaches, DGSR[17], uDSR[18], and TPSR[19] represent the state-of-the-art (SOTA) methods in this category.

While these transformer decoder-based methods[3,11,12] demonstrate strong potential in advancing symbolic regression—particularly in terms of scalability and generalizability—they also rely on a traditional teacher-forcing training paradigm. In this setup, the autoregressive model's next-token predictions are directly conditioned on the ground truth tokens, creating an exposure bias between training and inference. This not only introduces a gap that can lead to aggregated error during inference, but also allows gradients to propagate across multiple tokens, resulting in overly optimistic updates that may not reflect real inference dynamics[20–24].

The recently proposed Neural Symbolic Regression with Hypothesis[25] introduced an innovative framework that incorporates explicit assumptions about the underlying structure of the target expression to predict equations more effectively. This approach employs an additional transformer to process user prompts. However, despite these advancements, the model may not always reliably meet the specific requirements outlined in the input prompts, and its effectiveness remains limited without extensive prior knowledge to guide its predictions.

Inspired by recent efforts to integrate contextual knowledge and expert insights while addressing existing limitations, we propose a co-design mechanism for symbolic regression. This approach allows human experts to iteratively modify expressions based on discrepancies between observed data and the predicted outcomes. In our work, co-design involves an iterative interaction between an RL agent and domain experts. While theoretically possible in autoregressive (AR) models, co-design is hindered by exposure bias from the teacher-forcing training paradigm, which can degrade effectiveness. In contrast, RL's step-wise update mechanism mitigates compounding errors, enabling the model to learn a policy optimized for its observed contexts[20–24]. Additionally, RL naturally incorporates real-time human feedback, fostering a more interactive and efficient equation discovery process.

In this study, we propose the Symbolic Q-network (Sym-Q), an advanced interactive RL-based method for symbolic regression, as illustrated in Fig. 1. Sym-Q innovatively incorporates the existing expression tree as part of its observation (Fig. 1a). An RL agent then determines the optimal operations to expand and refine this tree until a satisfactory expression is formed (Fig. 1c). Unlike existing approaches, Sym-Q can handle the tree representation using various types of encoders, such as transformers or simple recurrent neural networks (RNNs), making it highly flexible and computationally efficient. This flexibility sets Sym-Q apart from other methods, as it does not rely on a transformer decoder structure to complete expressions. Instead, it leverages the strength of RL to guide the agent step-by-step through the process of building an equation, introducing an advanced paradigm in symbolic regression.

Importantly, Sym-Q supports prompt inputs like predefined expression trees and facilitates interactive design without the need for additional modules, such as extra prompt encoders[25]. Since the agent makes decisions at the level of individual operations rather than entire sequences, directly modifying the initial expression tree, the integration of prior knowledge is seamlessly preserved throughout the process. The proposed co-design mechanism optimally leverages the capabilities of RL and the contextual insights of domain experts. This collaboration enhances the accuracy, relevance, robustness, and meaningfulness of the models, ensuring that the agents align more closely with the true underlying physical processes and dynamics, resulting in more effective and impactful solutions. We demonstrate Sym-Q's comparable performance on the challenging SSDNC dataset against strong transformer-based baselines that do not incorporate online search. Additionally, we conduct a detailed analysis of fault patterns, examining error distributions and the specific stages at which errors appear during the expression generation process. Moreover, we conducted three experiments to evaluate the effectiveness of the proposed co-design mechanism. The first experiment tests its performance on the SSDNC benchmark. The second experiment involves recovering unseen drift terms in the Feynman dataset by utilizing the original equation as prior knowledge. The third experiment derives an analytic expression for a synthetic transit spectra dataset through an iterative co-design process guided by domain expert insights. The results demonstrate that incorporating prior knowledge significantly enhances expression recovery performance.

Results

Training datasets

For our training dataset, we generated five million mathematical expressions based on 100, 000 predefined expression skeletons with up to two independent variables, following the same setup as used in T-JSL[3]. These skeletons were created using the method described by Deep Learning for Symbolic Mathematic[26], which involves constructing a random unary-binary tree, populating its nodes with operators, and filling its leaves with independent variables or constants. We varied the constants in each expression 50 times and sampled 100 random data points for each variation. Further details about the dataset and the corresponding action space are available in Supplementary Material.

Evaluation on benchmark datasets

In this study, our primary focus is on generating accurate expression skeletons, which are essential for uncovering the underlying physical relationships. It is important to recognize that while various combinations of operator terms can achieve a high coefficient of determination (R^2)[27], a widely used metric in the field of symbolic regression, this does not inherently ensure the precision of final predictions or the correctness of the derived expressions[3]. R^2 is defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \quad (1)$$

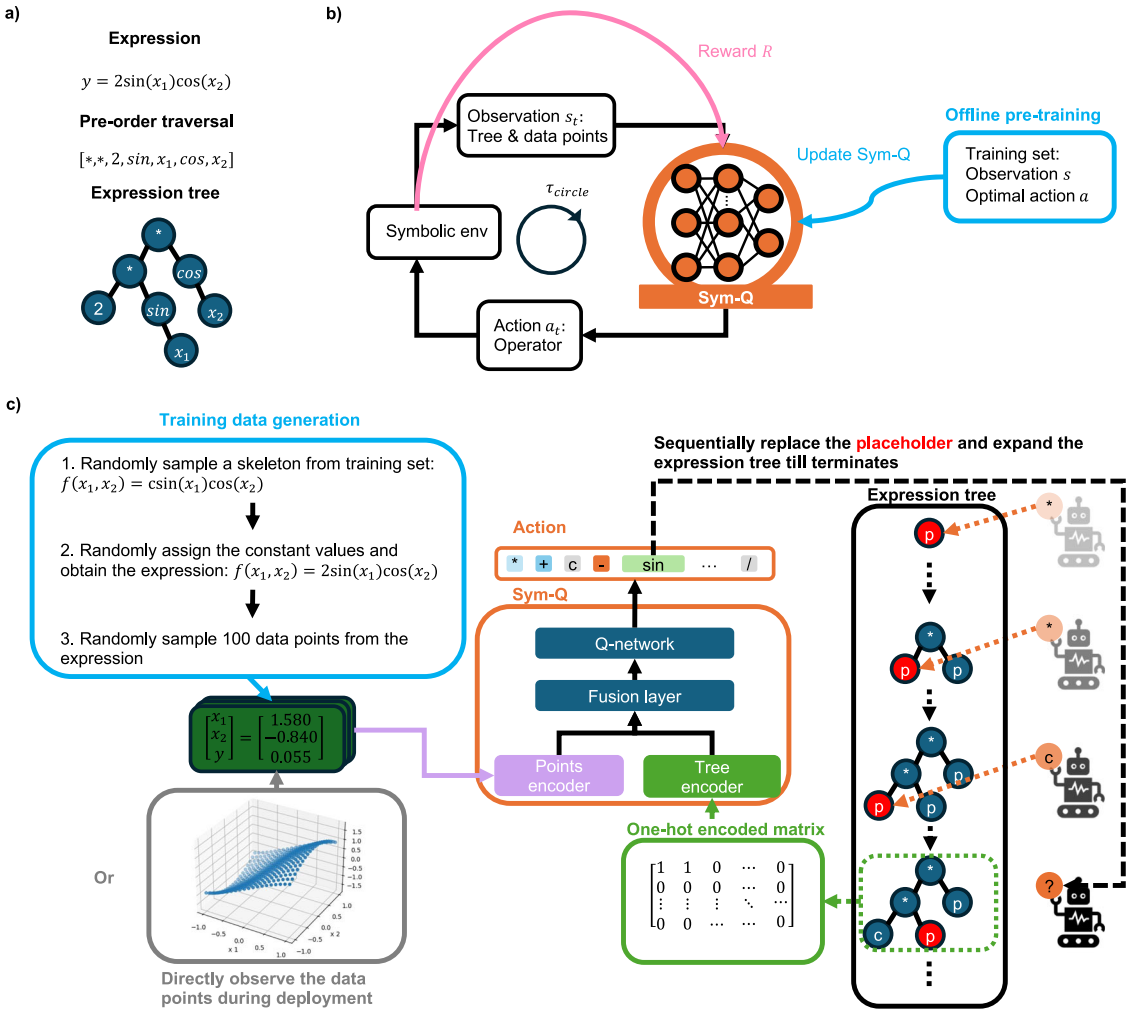

Fig. 1 | Overview of the proposed framework. a The expression and its corresponding expression tree. **b** The proposed Sym-Q agent supports both offline training with ground truth human knowledge and potentially online searching with reward signals. τ_{circle} represents the agent trajectories within the symbolic environment. **c** The Sym-Q architecture and step-wise decision-making mechanism.

where y_i and $\hat{y}_i$ represent the ground truth and predicted values for point i, respectively. $\bar{y}$ is the average of the y_i values across all data points, and n denotes the number of observed data points. Since our focus is on learning the correct expressions, we emphasize evaluating the skeleton recovery rate. The skeleton recovery rate is defined as the percentage of cases where the model fully recovers the correct expression structure, relative to the total number of expressions in the dataset. For the evaluation, we use the challenging SSDNC dataset proposed by Li et al.[3]. This test dataset includes the same skeletons as those generated in the training set but features different numerical coefficients. The SSDNC dataset, currently the most comprehensive of its kind, comprises 963 unseen equations. Further details about the SSDNC dataset are available in Supplementary Material.

Table 1 | Recovery rate of expression skeletons and R^2 values on the SSDNC benchmark dataset

NatureBlue Methods	Recovery rate	R^2
NatureGray SymbolicGPT[28]	50.3%	0.74087
NeSymReS[29]	63.4%	0.85792
NatureGray T-JSL[3]	75.2%	0.94782
Ours	**82.3%**	**0.95135**

All models implemented the beam search strategy with a beam size of 128.
The recovery rate represents the percentage of correctly recovered expression skeletons. R^2 is the coefficient of determination for the predicted results.
Numbers in bold indicate the best result within baselines.

To evaluate the effectiveness of our approach, we compare Sym-Q's against three SOTA transformer-based supervised learning methods—SymbolicGPT[28], Neural Symbolic Regression that Scales (NeSymReS)[29], and Transformer-based Model for Symbolic Regression via Joint Supervised Learning (T-JSL)[3]. These methods are directly comparable to Sym-Q since they rely solely on supervised learning without additional search components, making them well-suited for assessing Sym-Q's pretraining and inference capabilities. We exclude evolutionary algorithms, such as PySR[8], as they employ a fundamentally different search paradigm. Similarly, we do not include approaches that incorporate online search modules (e.g., DGSR[17]) or methods introducing end-to-end symbolic regression mechanisms (e.g., E2E[12]) to predict constants or coefficients, as these techniques rely on additional optimization strategies beyond supervised learning, making direct comparison less meaningful in this context.

As shown in Table 1, Sym-Q achieves a skeleton recovery rate of 42.7% without beam search and significantly improves to 82.3% when beam search is applied. All comparative methods, including those utilizing beam search, were evaluated under the same configuration with a beam size of 128. Under this setup, our model's inference time is ~1 min per expression with 100 data points, while NeSymReS[29] needs around 6 min. As detailed in Table 1, our proposed method

https://doi.org/10.1038/s41467-025-59288-y

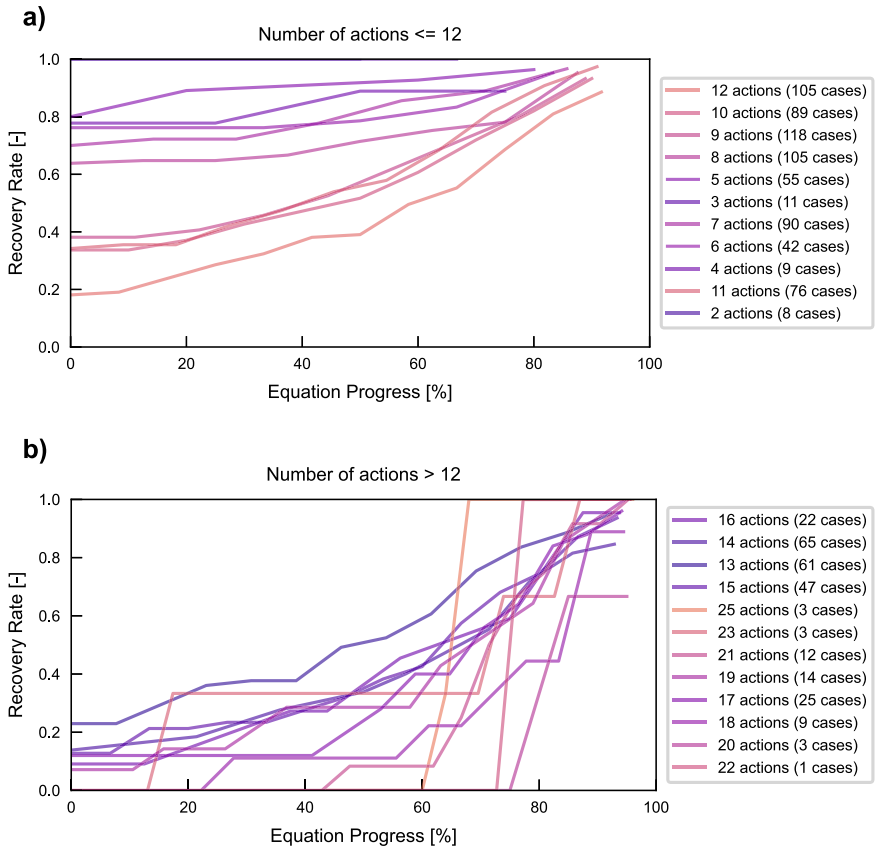

Fig. 2 | Performance improvement via co-design. The figures illustrate enhanced model performance on the SSDNC dataset when partial ground truth of the equation is provided by domain experts. The *x*-axis represents the percentage of given action sequences relative to the total. We analyzed equations of varying lengths, excluding categories with fewer than five samples to ensure statistical validity. **a** The results for expressions with fewer than 12 operations, and **b** the results for expressions with more than 12 operations. Recovery rates increase as more ground truth sequence steps are incorporated, demonstrating that domain expert guidance significantly improves model performance.

outperforms SOTA methods by up to 32.0% and achieves a superior average R^2 score of 0.95135.

Beyond the SSDNC dataset, Sym-Q demonstrates superior weighted average R^2 fitting accuracy across five well-recognized benchmarks, including Nguyen[30], Constant, Keijzer[31], R rationals[32], and AI-Feynman[33]. In addition to the three transformer-based methods previously outlined, we compared our approach with two online search approaches: DSR[9] and standard GP-based symbolic regression[34]. As detailed in Supplementary Material, Sym-Q outperforms all these methods, achieving a weighted average R^2 of 0.95044 across these six benchmarks. Since skeleton recovery rates have not been previously reported for these benchmarks, our analysis focuses solely on the weighted average R^2. Further details are provided in Supplementary Material. All evaluations and baseline implementations were conducted using the setup described by Li et al.[3].

Furthermore, to demonstrate scalability, we trained an additional model with three independent variables and evaluated its performance on SRBench against the pretrained model NeSymReS, which also utilizes three input variables. Sym-Q achieved superior equation recovery and higher R^2, as detailed in Supplementary Material.

Evaluation of the co-design mechanism

To validate the effectiveness of the proposed co-design mechanism in enhancing model performance, we conducted a series of targeted experiments.

In the first experiment, we evaluated Sym-Q's co-design mechanism by testing its ability to improve expression recovery rates for unseen mathematical equations using the SSDNC dataset[3]. We compared its performance against NeSymReS[29], which was not originally designed for co-design. To adapt NeSymReS for the co-design process, we provided the decoder with known tokens and allowed it to predict subsequent tokens autoregressively until reaching the "end" token. To ensure a fair comparison, we grouped expressions by the length of their ground truth decision sequences and averaged performance metrics within each group. Groups with fewer than five cases were excluded to ensure statistical validity. For each equation with a decision sequence of n actions, we performed a series of n experiments.

Initially, the model was presented with an empty expression tree, corresponding to zero actions taken from the ground truth decision sequence, representing a scenario with no prior expert knowledge. Subsequently, we incrementally incorporated domain knowledge, simulating human interaction or co-design at each step. Specifically, in the ith experiment, the expression tree included the first $i-1$ actions provided from the human expert's ground truth sequence. This setup allowed us to evaluate the model's performance improvement as more human interaction was integrated. We recorded performance at each increment and calculated the average for the respective groups. Trends observed in Fig. 2 show a clear enhancement in skeleton recovery rate metrics as additional ground truth steps are provided, demonstrating Sym-Q's potential to improve performance through strategic co-design integration. Moreover, Sym-Q's co-design mechanism outperformed NeSymReS[29]. Figure 3 illustrates the improvement in recovery rates across different percentages of provided ground truth information, specifically focusing on cases where the initial recovery rate is below 75%. The results show that while both

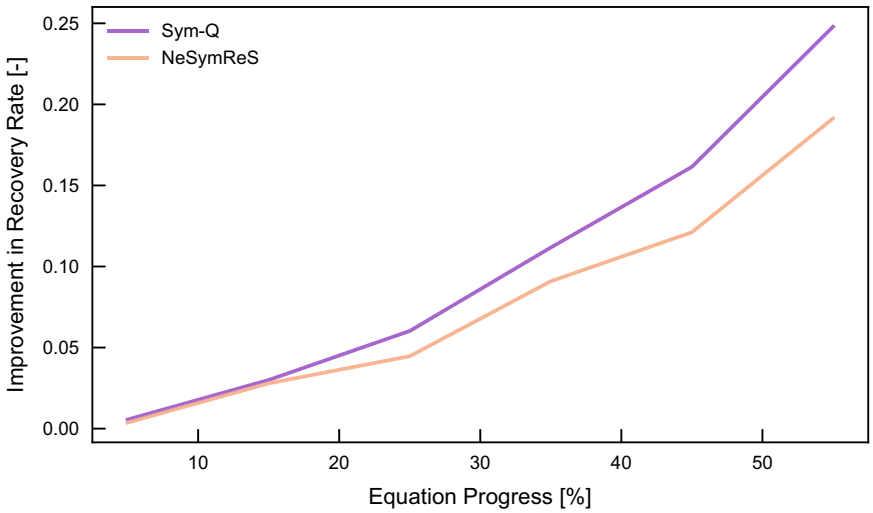

Fig. 3 | Comparison of performance improvement between Sym-Q and NeSymReS. Overall performance improvement with respect to the ratio of ground truth information provided for Sym-Q and NeSymReS, showing the co-design impact with up to a maximum of 60% additional information provided.

Sym-Q and NeSymReS improve performance as more information becomes available, Sym-Q exhibits a steeper and more consistent improvement curve. This indicates that Sym-Q's design is better suited to effectively utilize partial information, achieving higher recovery rates under the co-design mechanism. For more detailed analysis, please refer to Supplementary Material.

In the second experiment, we evaluated the model's ability to extend known equations to account for unmodeled dynamics using the Feynman dataset. We selected all equations with three or fewer variables, excluding those exceeding the model's size limits. The experimental setting addresses unmodeled dynamics by aiming to extend "textbook" equations to account for deviations in observations. Specifically, we aim to identify potential "drifts," which are missing multiplicative or additive terms in known equations that describe the general behavior of observations or underlying laws. For example, additional factors such as air friction may influence a free-fall experiment. Using the AI-Feynman dataset, which contains physics-based equations, we evaluated the model's accuracy in retrieving different types of drift, including multiplicative and additive terms, exponential decay, periodic terms, and squared terms of the original equations. The results highlight that the co-design mechanism significantly improves the model's performance in addressing such tasks. Figure 4 demonstrates the effectiveness of Sym-Q in recovering different drift scenarios, where human experts provide the core components of textbook formulas. Both models were evaluated without beam search, as it is challenging to decouple model performance from the online search enhancements in this setting. For consistency, we employed the same BFGS setup for both models to determine the constants of the output skeleton.

The performance of Sym-Q and NeSymRes is comparable when prior information is not provided. However, a notable distinction arises when the models are provided with "textbook" equations. In such cases, Sym-Q demonstrates superior capabilities in co-design contexts, leveraging prior information. Interestingly, while prior knowledge can sometimes limit a model's expressivity and degrade its performance, as observed with NeSymRes in Fig. 4, Sym-Q remains unaffected by this limitation. This challenge often arises when out-of-distribution (OOD) priors are provided, where equation skeletons differ structurally from those the model was trained on, as the same equation can often be expressed in multiple forms. Sym-Q's ability to overcome these constraints underscores its exceptional generalization capabilities.

A key distinction that can partly explain these differences is NeSymRes's ability to generate an end-of-equation token. While this token allows the model to decide when an equation is "good enough," it can also hinder performance by prematurely halting the generation process. In contrast, Sym-Q does not rely on an explicit end-of-equation token; instead, it uses consistency checks to ensure that the generated symbolic expressions are complete and contain no unfinished terms. To prevent NeSymRes from stopping prematurely when provided with a context equation, we restricted the model from outputting the end-of-equation token as an initial output.

Another key factor influencing the performance difference is the training mechanism. Like other autoregressive symbolic models, NeSymRes follows a traditional teacher-forcing paradigm, where the model's next-token predictions are conditioned directly on ground truth tokens during training. This introduces exposure bias, as the model never learns to recover from its own errors, leading to a mismatch between training and inference. This discrepancy may explain why incorporating prior knowledge can sometimes degrade its performance, as observed with NeSymRes in Fig. 4. In contrast, Sym-Q utilizes RL's step-wise update mechanism. Unlike autoregressive models trained with teacher-forcing, where gradients propagate through all previous tokens during training, Sym-Q's RL-based training updates each step independently without gradient backpropagation through past predictions. This design prevents compounding errors during training and allows the model to learn a policy optimized for its observed contexts. Although Sym-Q may still generate expressions autoregressively at inference time, the RL training paradigm reduces exposure bias by directly optimizing for sequence-level correctness, enabling better utilization of prior knowledge without performance degradation.

Additionally, we implemented the same consistency checks used by Sym-Q as an extra stopping policy, ensuring a fair comparison between the two approaches.

Regarding the drifts, Sym-Q successfully recovered additive and multiplicative exponential factors over 75% of the equations, demonstrating its ability to identify unmodeled decaying factors (e.g., resistance or friction) as well as exponentially increasing drifts (e.g., unstable modes in the underneath dynamics). Furthermore, the model effectively recovered additive constant and linear terms, which are often associated with biased or miscalibrated sensors. Recovery rates ranged from 40%–80%, depending on the relative magnitude of the bias within the data.

https://doi.org/10.1038/s41467-025-59288-y

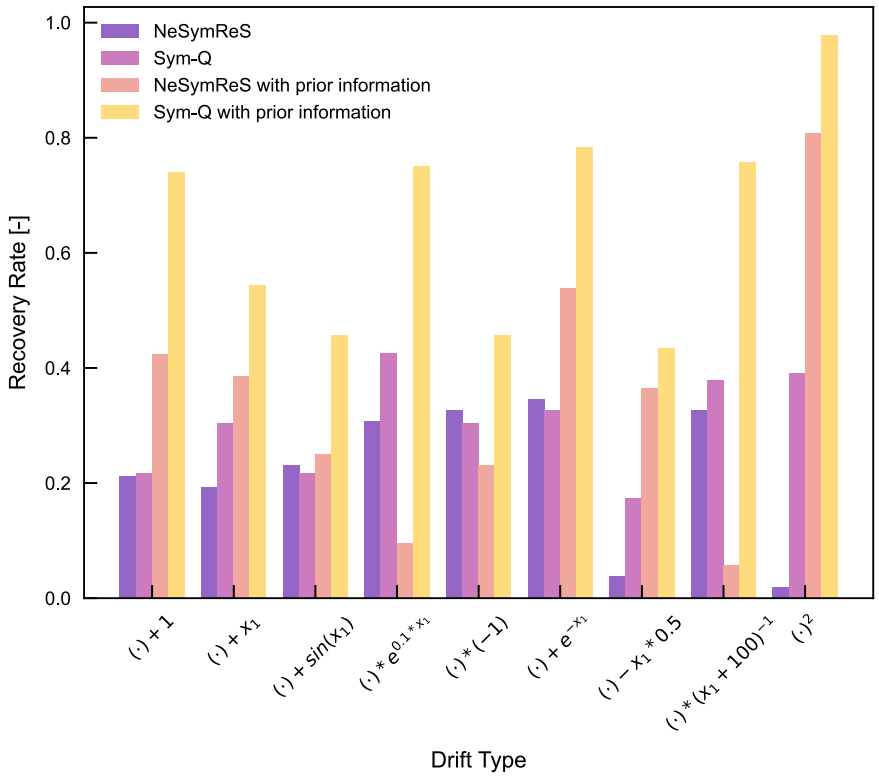

Fig. 4 | Model accuracy in retrieving different additive and multiplicative drifts over the AI-Feynman benchmark. Violet (NeSymRes) and magenta (Sym-Q) bars depict the recovery rate of equations with additional drifts, without giving any prior knowledge. Salmon (NeSymRes) and yellow (Sym-Q) bars refer to the case where the textbook equation is given as a prior and, therefore, only the additional components need to be recovered.

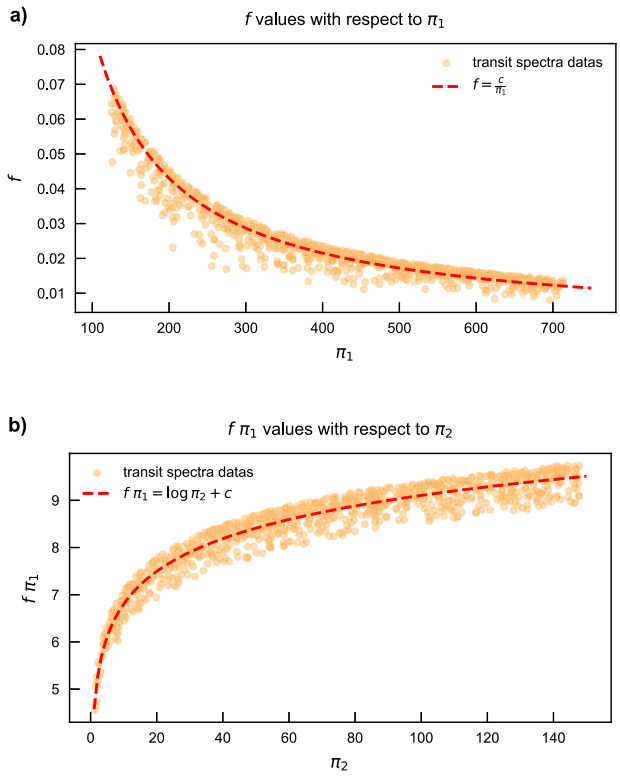

Fig. 5 | Empirical correlations in the transit spectra dataset. a Displays a scatter plot with π_1 on the x-axis and f on the y-axis, illustrating their relationship, while **b** shows a scatter plot with π_2 on the x-axis and $f\pi_1$ on the y-axis. These correlations, derived from the transit spectra[35], were used to establish the two priors for co-designing the analytical expression of f.

To further evaluate the versatility of Sym-Q with co-design, we applied it to derive an analytical expression for the transit radius factor f of a hot-Jupiter exoplanet using atmospheric parameters (π_i). Building on the work from Matchev et al.[2], we tested Sym-Q using a synthetic transit spectra dataset[35]. Applying the model without co-design, and running it autoregressively on the unprocessed data, we obtained a solution with a low MSE of 7.6×10^{-4}; however, the resulting equation had an R^2 value below 0.95, indicating a significant deviation from the ground truth. This discrepancy is primarily due to the differing variable ranges in the dataset $[10^{-2}, 10^{3}]$ compared to the training distribution of Sym-Q $[10^{-1}, 10^{1}]$. While rescaling could mitigate this issue, it would also broaden the range of constant values, complicating the online constant search.

Upon further examination of the data, we make the following observations. Firstly, we observed that:

f is inversely proportional to π_1 (Fig. 5a).

Based on this insight, we provide the prior $f = \frac{1}{\pi_1 + c} * (\cdot)$ to the model. This significantly improves the performance, resulting in an MSE of 4×10^{-6} and an R^2 of 0.974. We infer that the analytical expression of f takes the form:

$$f = \frac{1}{\pi_1 + c} * g(\pi_1, \pi_2) \tag{2}$$

where $\{\pi_i\}_{i>2}$ are superfluous. Secondly, we observe that:

$g \approx f \cdot \pi_1$ scales logarithmically with respect to π_2 (Fig. 5b).

With this in mind, we propose the prior $f = \frac{1}{\pi_1 + c} * \log(\pi_2 * (\cdot))$ and successfully retrieve the ground truth analytical expression:

$$f(\pi_1, \pi_2, \pi_{\text{extra}}) = f(\pi_1, \pi_2) = \frac{(c + log(c * sqrt(\pi_1) * \pi_2))}{(c + \pi_1)} \tag{3}$$

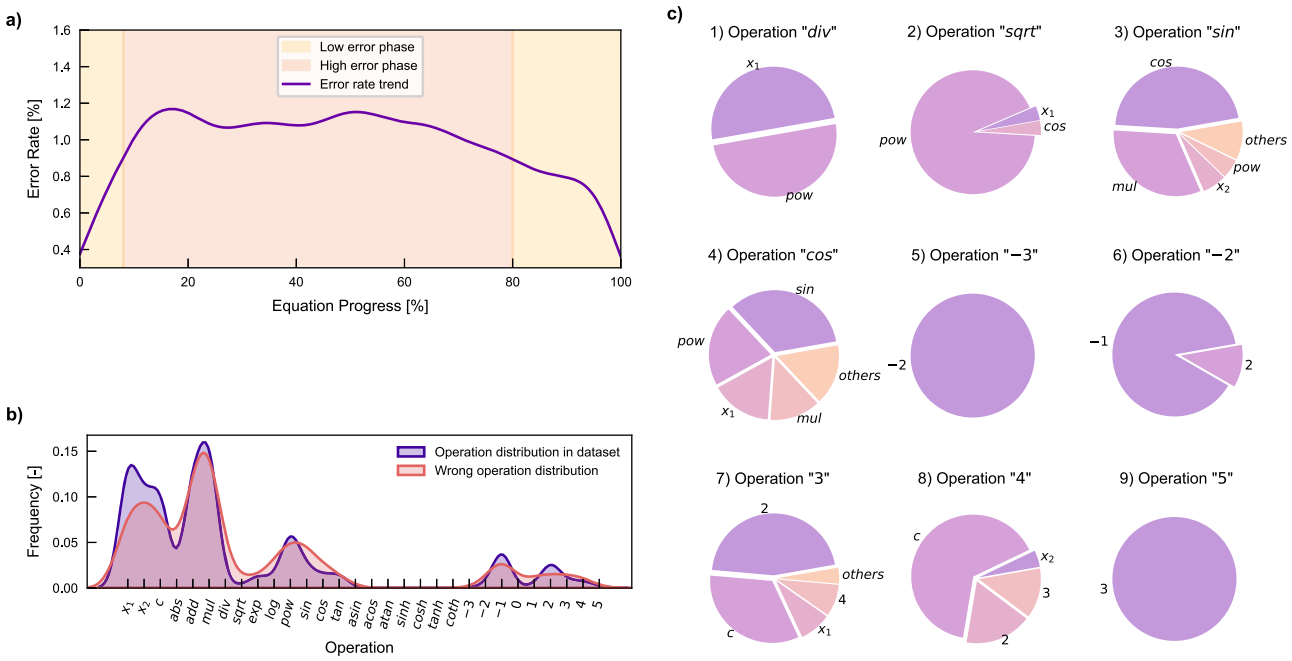

Fig. 6 | Error analysis on agent's step-wise decisions. a Error rate trend throughout the expression generation process. The *y*-axis displays error rates, and the *x*-axis represents the percentage of completion of the expression generation process, with 100% indicating a fully formed equation. A distinct pattern is evident: higher error rates occur in the middle stages of the decision-making process, while the initial and final phases exhibit lower error rates. **b** Comparative analysis of operation frequency and error incidence. This graph contrasts the frequency of operations in the training set (blue) with the incidence of incorrect actions (red) observed in the SSDNC dataset. The *y*-axis represents error rates, while the *x*-axis lists the various operations. A notable correlation between these distributions is observed, suggesting that the agent tends to favor operations that it encountered more frequently during training. **c** Distribution of incorrect decision choices. The pie charts illustrate the decisions made during instances of incorrect decision-making. Each segment of the chart corresponds to an incorrect choice in relation to a specific correct operation. The size of each segment represents the proportion of these incorrect choices, summing to a total of 1. The title above each chart identifies the correct operation targeted in each specific instance.

By incorporating astrophysical constants as additional task-specific priors into the BFGS optimization, the online search ultimately converges to the correct closed-form solution:

$$f(\pi_1, \pi_2, \pi_{\text{extra}}) = f(\pi_1, \pi_2) = \frac{\gamma_E + log(2\pi_2\sqrt{\pi\pi_1})}{\pi_1} \quad (4)$$

This iterative co-design process, supported by domain expert insights, allows Sym-Q to accurately recover the underlying physical equation by strategically guiding the model through informed priors and targeted refinements.

Detailed error analysis of the agent's decision-making

To thoroughly evaluate the agent's performance, we conducted a step-wise analysis of its decision-making process using the SSDNC dataset. This section presents key findings, including the error rate trend throughout the expression generation process, the distribution of incorrect selections, and the specific types of errors encountered. Step-wise accuracy was calculated by assessing whether the model's output decision sequence aligned with the expected decision sequence from the SSDNC dataset, based on the correct preceding expression tree encoding and the observed data point.

In addition to overall step accuracy, it is crucial to identify where the agent tends to make errors to thoroughly evaluate its behavior. To account for expressions of varying lengths, we normalized them by calculating their percentage of completion. Notably, the highest error rates occur during the middle stages of the process, as illustrated in Fig. 6a. A possible explanation for this trend is that initial decisions often involve straightforward binary operations such as addition (+) and multiplication (×), which typically follow a general strategy of expanding the expression tree early on. Conversely, decisions made towards the end of the process tend to be easier as the expression becomes more focused and specific.

Our analysis indicates that the frequency of errors closely aligns with the distribution of operations within the training dataset. Specifically, the agent tends to favor operations that it encountered more frequently during training, as clearly depicted in Fig. 6b. This observation suggests a potential imbalance in the design of the training dataset, which should be carefully considered and addressed in future work.

Our observations indicate that the agent often struggles with determining exact constant values. While it correctly identifies the need to include a constant, the chosen value is often incorrect, as illustrated in Fig. 6c.5–c.9. This issue arises because the model does not perform coefficient fitting for constants, instead treating each discrete value as a separate action. Additionally, the agent commonly confuses $\sin(\cdot)$ with $\cos(\cdot)$ functions, likely due to their similar mathematical properties, as illustrated in Fig. 6c.3 and c.4. Another notable observation is the frequent confusion between the square root operation ($\sqrt{\cdot}$) and the power operation, as illustrated in Fig. 6c.7. This confusion is understandable, given that the square root is equivalent to raising a number to the power of $\frac{1}{2}$. Similarly, the division operation (÷) is often mistaken for the power operation, as division is mathematically equivalent to raising a number to the power of -1, as shown in Fig. 6c.1. These patterns suggest that the agent understands the effects of these operations on mathematical relationships.

Discussion

The study presents an advanced paradigm for large-scale symbolic regression by explicitly framing it as a sequential decision-making task and addressing it with the proposed Sym-Q algorithm. Sym-Q not only

excels at discovering symbolic expressions from scratch but also overcomes a key limitation of traditional symbolic regression models: their less effectiveness and efficiency in incorporating human priors due to limited generalization and extrapolation capabilities, particularly when faced with fixed or OOD equation structures.

To address this challenge, Sym-Q integrates a co-design mechanism that enables more effective interactive refinement of expressions. This innovative approach allows domain experts to provide partially defined expression trees, fostering real-time collaboration with the model. By dynamically modifying generated nodes or providing prior information, experts can more effectively guide the agent to develop mathematical expressions that accurately capture underlying problem dynamics and adhere to established physical laws, particularly when partial domain knowledge is available. This co-design mechanism effectively integrates expert insights and user hypotheses, enhancing both the interpretability and scientific relevance of the resulting expressions.

A key strength of Sym-Q lies in its versatility, as it can be combined with various types of encoders, from advanced transformer architectures to simpler RNNs, for processing expression trees—differentiating it from prior works. By leveraging RL, Sym-Q avoids the exposure bias by guiding the model step-by-step through the equation construction process. Rather than memorizing token sequences, the agent learns to generate expressions dynamically based on observations, effectively alleviating the train-test mismatch inherent in teacher-forced models. This structured learning approach allows Sym-Q to generalize better and adapt to unseen equations more effectively. To ensure a meaningful and fair evaluation, we compare Sym-Q against three SOTA transformer-based supervised learning models that, like Sym-Q, do not incorporate additional online search mechanisms. Extensive benchmark evaluations demonstrate that Sym-Q performs comparably to other SOTA algorithms in fitting accuracy and recovery rates across most benchmark datasets used for pretrained large-scale symbolic regression models.

This innovative framework is particularly well-suited for co-design, as evidenced by the demonstrated effectiveness of Sym-Q's co-design mechanism in the evaluation experiments. By integrating domain knowledge, Sym-Q effectively tackles complex scenarios, such as recovering drift terms in the Feynman dataset and deriving analytical expressions from synthetic transit spectra. This integration leads to notable improvements in both recovery rates and the coefficient of determination. Our experiments show that as more domain expertise is incorporated, the co-design approach not only enhances performance metrics but also aligns the model's outputs with underlying physical laws and patterns. Compared to NeSymReS, Sym-Q's co-design mechanism consistently delivers superior results, showcasing more reliable improvements across varying levels of ground truth availability and effectively leveraging partial information.

Beyond its immediate performance gains, Sym-Q's co-design capabilities unlock significant advancements in symbolic regression, particularly in scenarios involving OOD equation structures. By effectively leveraging partial prior knowledge and dynamically adapting to unfamiliar equation forms, Sym-Q demonstrates robust generalization and adaptability, making it a powerful tool for tackling complex, real-world problems.

A promising direction for future research involves extending Sym-Q to tackle more complex types of expressions, such as Ordinary Differential Equations and Partial Differential Equations. Additionally, further enhancements to the co-design framework could explore more sophisticated ways of integrating expert knowledge, potentially incorporating comprehensive constraints or hierarchical guidance to further refine the discovery process. While we have validated Sym-Q's scalability from two to three variables, extending the model to higher-dimensional spaces remains an important direction for future research.

Methods

Overview

We propose Sym-Q, an RL-based method designed to tackle large-scale symbolic regression problems, featuring a collaborative co-design mechanism (see Fig. 1). Leveraging our proposed offline RL training approach, Sym-Q facilitates interactive symbolic regression that integrates iteratively domain expert knowledge. As illustrated in Fig. 1b, Sym-Q effectively learns from both optimal demonstrations and sub-optimal reward signals, enhancing its adaptability and performance in complex regression tasks.

To establish a robust co-design symbolic regression framework, we conceptualize the generation of an expression tree as a Markov Decision Process (MDP). In this framework, the agent is trained to interpret observations and sequentially expand and refine the expression tree (Fig. 1b). The *state* is defined by the current expression tree and the observed data at each step, while actions involve selecting mathematical expressions to populate the nodes of the expression tree (Fig. 1c). The effectiveness of these actions is quantified using the coefficient of determination (R^2)[27] as the reward metric, providing a clear measure of model accuracy. The agent sequentially replaces placeholders and expands the expression tree until the process concludes. Expression trees are generated using a beam search, repeated 128 times as in prior works[3,29], resulting in 128 distinct skeletons. The final expression is derived by applying the Broyden-Fletcher-Goldfarb-Shanno (BFGS)[36] optimization on the constant terms in the expressions. The expression with the lowest fitting error is selected as the agent's final answer.

This formulation as a sequential decision process enriches the co-design mechanism by enabling domain experts to inject explicit prior knowledge based on the problem's context. Such expertise guides the model's decisions, allowing for dynamic refinement and optimization of the expression tree. Moreover, operations can be adjusted or corrected at any step based on current expert insights, ensuring that the expressions generated are both precise and pertinent.

Expression tree

An expression tree, as defined by Hopcroft et al.[37], is a hierarchical structure where internal nodes represent mathematical operations, such as addition, subtraction, and logarithm, while leaf nodes denote constants or variables. This tree-based representation is particularly advantageous for sequential decision-making processes, as it allows for the incremental construction and refinement of mathematical expressions. Illustrated in Fig. 1a, the expression tree facilitates the systematic assembly and analysis of complex mathematical expressions, especially when their exact forms are initially unknown. In this work, we use one-hot encoding to represent tokens within the expression tree, aligning with traditional sequence encoding methods that typically employ embedding layers. This encoding choice does not introduce significant differences in performance or functionality compared to other methods.

Symbolic regression as a sequential decision-making task

Symbolic regression involves searching for a sequence of operations $a_{1:T} = a_1, a_2, \ldots, a_T$ to construct an expression tree that accurately fits observational data. In this context, each operation a_t represents a mathematical function, either a unary or binary operation within a predefined search space at decision step t. Previous large-scale transformer-based models[3,29] address symbolic regression similarly to machine translation. These models use feature extractors to process input data points and utilize a transformer to decode these features into expression skeletons. In this setup, the model's next-token predictions during training are conditioned on ground truth tokens, resulting in exposure bias. This approach causes the model to become overly reliant on teacher-forcing supervision, creating a significant discrepancy between the training and inference phases. As a result, the

model often struggles during real-world predictions, where ground truth tokens are unavailable, leading to error accumulation.

In contrast, our approach utilizes offline RL, eliminating the reliance on teacher-forcing supervision at the decoder level. Instead, the model iteratively uses its own predictions as inputs during decoding, ensuring that the training process closely aligns with real inference conditions. This approach also prevents direct gradient back-propagation across multiple tokens, promoting more stable and realistic model updates[20–22]. Our study conceptualizes symbolic regression as an MDP. An MDP is characterized by the tuple (S, A, r, P, ρ), where S denotes the set of states reflecting the current situation, A represents the set of possible actions, $r(s, a)$ is the reward function, $P(s'|s, a)$ defines the state transition probabilities, and $\rho(s)$ is the initial state distribution.

At each decision step t, the agent selects an action a_t based on the policy $\pi_\theta(a_t|s_t)$, where s_t represents the current state. This state encompasses both the observed data points (X, y) and the encoded structure of the current expression tree, derived from previous actions $a_{1:t-1}$. The action a_t determines the next operation in the sequence, thereby incrementally building the symbolic expression $a_{1:T} = a_1, a_2, \ldots, a_T$ as a trajectory of the MDP, denoted by $\tau = s_1, a_1, s_2, a_2, \ldots, s_T, a_T$. In our model, rewards are assigned exclusively at the termination state, with all intermediate states receiving a reward of 0. We utilize the coefficient of determination R^2 as the reward metric, providing a clear measure of model accuracy. Under this MDP formulation, the objective of the RL-based agent is to discover a policy π that maximizes the cumulative reward $J(\pi)$:

$$J(\pi) = \mathbb{E}_{\tau\sim\rho_\pi} \sum_{t=0}^{\infty} r(s_t, a_t). \tag{5}$$

Symbolic regression targets only optimal solutions, with all sub-optimal solutions receiving significantly lower rewards. To address the challenges of sparse reward and a large search space inherent in symbolic regression, we employ offline RL, a learning paradigm that merges the ability of supervised learning to leverage existing data with RL's capacity to optimize arbitrary rewards and leverage temporal compositionality[38–41].

Under the offline RL framework, the objective remains the optimization of the function specified in Eq. (5). However, because the agent cannot interact directly with the environment, it must adopt a different approach to learn from the static dataset. By relying on a predetermined, static dataset of transitions denoted as $D = (s_t^i, a_t^i, s_{t+1}^i, r_t^i)$, offline RL circumvents the inefficiencies of online exploration and mitigates the impact of sparse reward signals. By leveraging this approach, Sym-Q efficiently utilizes existing data to train the agent, overcoming the limitations of online exploration and reward sparsity.

Symbolic Q-network

Sym-Q is an offline RL algorithm specifically designed to address large-scale symbolic regression problems by enabling sequential decision-making and efficient step-wise updates. Sym-Q comprises three integral modules:

1. Point set encoder $E_\phi^{points}(X, y)$ which transforms the point sets $(X, y) = ((x_1^1, x_1^2, y_1), (x_2^1, x_2^2, y_2), \ldots, (x_n^1, x_n^2, y_n))$ associated with each equation into a latent space, resulting in a fixed-size latent representation $z^p \in \mathbb{R}^{1\times K_p}$, where K_p denotes the dimensionality of the latent variable;
2. Expression tree encoder $E_\psi^{tree}(M_t)$ which maps the current tree's one-hot encoded matrix M_t into another fixed-size latent representation $z^t \in \mathbb{R}^{1\times K_t}$, where K_t indicates the dimensionality of the variable;
3. Q-network $Q_\theta(s_t)$ which calculates the Q-value for each potential operation, given the combined latent state $s_t = (z^p, z^t)$.

This architecture enables Sym-Q to address symbolic regression in a more efficient way. The comprehensive framework of Sym-Q is depicted in Fig. 1. For further details on the neural network architectures and hyperparameters used, please refer to Supplementary Material. Additionally, we conducted experiments using various combinations of encoders, including Transformer and RNN architectures, paired with different loss functions such as cross-entropy and Classical mean squared error for Q-Learning. Our findings demonstrate that Sym-Q is the first large-scale symbolic regression model capable of processing sequence data without depending on a transformer-based architecture. This advanced approach not only offers a more lightweight and modular framework but also underscores the potential of RL-based methods for tackling complex symbolic regression tasks. For further details, refer to Supplementary Material.

Efficient conservative offline Q-learning for symbolic regression

In training the Sym-Q through offline RL, a significant challenge is the overestimation of values due to distributional shifts between the dataset and the learned policy. To address this, we adopt Conservative Q-learning (CQL)[40], which minimizes the values of state-action pairs outside the training data distribution, while simultaneously maximizing the values within this distribution. Given the sparse reward structure of symbolic regression, where rewards are typically given only at the completion of each trajectory and most sub-optimal trajectories receive lower rewards, we have adapted CQL to this context. Our modified version of CQL is customized for symbolic regression, aiming to effectively leverage optimal offline data in this domain:

$$J(\theta, \psi) = -\mathbb{E}_{(s,a^i)\sim\mathcal{D}}\left[\log\left(\frac{e^{Q_{\theta,\psi}(s,a^i)}}{\sum_{j=1}^{m} e^{Q_{\theta,\psi}(s,a^j)}}\right)\right], \tag{6}$$

where $Q_{\theta,\psi}$ represents the network and its weight parameters, including those beyond the points encoder. D represents the training dataset, a^i is the action demonstrated at state s, and $a^j | j \neq i, 1 \leq j \leq m$ are the non-demonstrated actions, regarded as sub-optimal.

Building on the relationship between CQL loss and cross-entropy loss, our proposed objective function represents the log probability of the optimal action. This mirrors the emphasis of cross-entropy on identifying the correct class. The inclusion of the softmax function in the denominator serves to normalize the Q values across all possible actions, effectively converting them into a probability distribution.

By adopting this objective function, our approach more closely aligns with the principles of traditional supervised learning. This alignment makes the method more intuitive and straightforward to implement, particularly in scenarios where the correct action at each state is well-defined, such as in symbolic regression.

Supervised contrastive learning for point encoders

Following a similar idea from Li et al.[3], Dong et al.[42], we implemented the supervised contrastive loss for expressions and data points that share the same skeleton. This approach uses the skeleton of expressions as category labels to enrich supervisory information. More specifically, expressions and their corresponding latent points encoding z_i^p, which belong to the same skeleton, are grouped together in the embedding space. At the same time, we ensure that the latent points encoding z_j^p from different skeletons are distanced from each other. The supervised contrastive loss is defined as:

$$J(\phi) = \sum_{i=1}^{N} \frac{-1}{|\mathcal{P}(i)|} \sum_{p\in\mathcal{P}(i)} \log \frac{\exp(E_\phi^{points}(X,y)^i \cdot E_\phi^{points}(X,y)^p/\tau)}{\sum_{j=1}^{N} 1_{[j\neq i]} \exp(E_\phi^{points}(X,y)^i \cdot E_\phi^{points}(X,y)^j/\tau)} \tag{7}$$

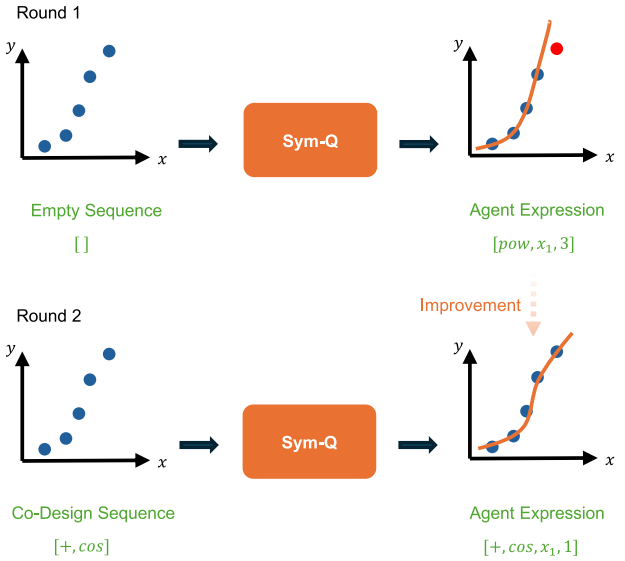

Fig. 7 | Iterative enhancement through Co-Design in Sym-Q. This diagram illustrates the iterative refinement process within Sym-Q when initial solutions are sub-optimal.

where $\mathcal{P}(i)$ represents the set of indices for all positives in the multi-viewed batch distinct from i, and $|\mathcal{P}(i)|$ is its cardinality; τ is an adjustable scalar temperature parameter controlling class separation; N represents the mini-batch size; $z_i^p = E_\phi^{points}(X,y)^i$ is the points embedding of the sample i in a batch, $z_p^p = E_\phi^{points}(X,y)^p$ is the points embedding of a positive sample sharing the same skeleton as sample i. The overall loss is then given by:

$$J(\phi,\theta,\psi) = \alpha_1 J(\theta,\psi) + \alpha_2 J(\phi), \tag{8}$$

where $\alpha_{1,2}$ are scalar weighting hyperparameters. More details about the parameter settings and network architectures can be found in Supplementary Material.

Co-design mechanism

A central feature of the proposed architecture is the innovative co-design mechanism, which integrates users' domain knowledge with the learned knowledge of the model, allowing for interaction at any decision step.

This collaborative approach leverages the computational power and pattern recognition capabilities of an RL agent while incorporating the contextual knowledge and insights from human experts. By involving domain experts, the generated mathematical expressions become more accurate, relevant, and meaningful. This synergy between human expertise and machine learning ensures that the model is grounded in the true underlying physical processes and dynamics. Consequently, the collaboration facilitates more effective problem-solving, leading to impactful and meaningful solutions. This dynamic interplay between human and machine intelligence is crucial for addressing the limitations of purely automated methods, fostering a deeper understanding of complex data relationships, and enhancing the model's adaptability to diverse and evolving problem contexts.

This co-design approach can be conceptualized as an environment within an RL framework. At each interaction step, the agent accesses both the point sets and the current expression tree. The expression tree, or parts of it, can originate from three potential sources: (1) decisions made by Sym-Q, (2) hypotheses or modifications introduced through direct human intervention, and (3) partial decision sequences suggested by other symbolic regression models. In this work, we focus only on the first two types of decisions as prior information to the Sym-Q agent. The co-design mechanism enables domain experts to integrate explicit prior knowledge tailored to the problem's context. Their expertise guides the model's decisions, enabling dynamic refinement and optimization of the expression tree. Additionally, operations can be modified or corrected at any stage based on domain expert insights, ensuring that the generated expressions are both accurate and relevant.

Our proposed co-design process, illustrated in Fig. 7, begins with the agent attempting to construct an expression starting from an empty sequence. This initial attempt often results in an expression that only partially fits the data points, revealing a misalignment between the agent's interpretation of the data structure and an accurate solution. In the next step, we employ a co-design mechanism to guide the agent toward a more accurate representation. To expand the expression tree, we introduce a binary operator as the initial token, followed by a specific operator that better captures the shape of the data points. This step leverages human intuition to correct the agent's initial misjudgments, enhancing its problem-solving capabilities. By integrating insights into the agent's behavior with domain expertise, our co-design process becomes both realistic and reliable. This integration significantly improves the agent's ability to generate accurate expressions, making it more effective in addressing complex problems.

Data availability

The data generated in this study for training and testing have been deposited in the Zenodo database under accession code https://doi.org/10.5281/zenodo.15105239.

Code availability

Our reproducible code is available at https://github.com/EPFL-IMOS/Sym-Q.

References

1. Wadekar, D. et al. Augmenting astrophysical scaling relations with machine learning: application to reducing the SZ flux-mass scatter. *arXiv preprint arXiv:2201.01305* (2022).
2. Matchev, K. T., Matcheva, K. & Roman, A. Analytical modeling of exoplanet transit spectroscopy with dimensional analysis and symbolic regression. *Astrophys. J.* **930**, 33 (2022).
3. Li, W. et al. Transformer-based model for symbolic regression via joint supervised learning. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=ULzyv9M1j5 (2023).
4. Virgolin, M. & Pissis, S. P. Symbolic Regression is NP-hard. *Trans. Mach. Learn. Res.* https://openreview.net/forum?id=LTiaPxqe2e (2022).
5. Forrest, S. Genetic algorithms: principles of natural selection applied to computation. *Science* **261**, 872–878 (1993).
6. Schmidt, M. & Lipson, H. Distilling free-form natural laws from experimental data. *Science* **324**, 81–85 (2009).
7. Błądek, I. & Krawiec, K. Solving symbolic regression problems with formal constraints. In *Proceedings of the Genetic and Evolutionary Computation Conference* 977–984 (Association for Computing Machinery, 2019).
8. Cranmer, M. Interpretable machine learning for science with PySR and symbolic regression. jl. *arXiv preprint arXiv:2305.01582* (2023).
9. Petersen, B. K. et al. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *International Conference on Learning Representations* https://openreview.net/forum?id=m5Qsh0kBQG (2021).
10. Mundhenk, T. N. et al. Symbolic regression via neural-guided genetic programming population seeding. In *Proceedings of the 35th International Conference on Neural Information Processing Systems* 24912–24923 (2021).

11. Vastl, M. et al. Symformer: End-to-end symbolic regression using transformer-based architecture. *IEEE Access* **12**, 37840–37849 (2024).
12. Kamienny, P.-A., d'Ascoli, S., Lample, G. & Charton, F. End-to-end symbolic regression with transformers. *Adv. Neural Inf. Process. Syst.* **35**, 10269–10281 (2022).
13. Meidani, K. et al. SNIP: Bridging mathematical symbolic and numeric realms with unified pre-training. In *NeurIPS 2023 AI for Science Workshop* https://openreview.net/forum?id=Nn43zREWvX (2023).
14. Li, Y. et al. MMSR: symbolic regression is a multi-modal information fusion task. *Inf. Fusion* **114**, 102681 (2025).
15. Li, Y. et al. Discovering mathematical formulas from data via GPT-guided Monte Carlo tree search. *Expert Syst. Appl.* **281**, 127591 (2025).
16. Li, Y. et al. Generative pre-trained transformer for symbolic regression base in-context reinforcement learning. *arXiv preprint arXiv:2404.06330* (2024).
17. Holt, S. et al. Deep generative symbolic regression. In *The Eleventh International Conference on Learning Representations* https://openreview.net/forum?id=o7koEEMA1bR (2023).
18. Landajuela, M. et al. A unified framework for deep symbolic regression. *Adv. Neural Inf. Process. Syst.* **35**, 33985–33998 (2022).
19. Shojaee, P., Meidani, K., Barati Farimani, A. & Reddy, C. Transformer-based planning for symbolic regression. *Adv. Neural Inf. Process. Syst.* **36**, 45907–45919 (2023).
20. Bahdanau, D. et al. An actor-critic algorithm for sequence prediction. In *International Conference on Learning Representations* https://openreview.net/forum?id=SJDaqqveg (2017).
21. Xu, Y. et al. Rethinking exposure bias in language modeling. *arXiv preprint arXiv:1910.11235* (2019).
22. Tsai, C.-P. & Lee, H.-Y. Order-free learning alleviating exposure bias in multi-label classification. *Proc. AAAI Conf. Artif. Intell.* **34**, 6038–6045 (2020).
23. Pang, R. Y. & He, H. Text generation by learning from demonstrations. In *International Conference on Learning Representations* https://openreview.net/forum?id=RovX-uQ1Hua (2021).
24. Hao, Y., Liu, Y. & Mou, L. Teacher forcing recovers reward functions for text generation. *Adv. Neural Inf. Process. Syst.* **35**, 12594–12607 (2022).
25. Bendinelli, T., Biggio, L. & Kamienny, P.-A. Controllable neural symbolic regression. In *International Conference on Machine Learning* 2063–2077 (2023).
26. Lample, G. & Charton, F. Deep learning for symbolic mathematics. In *International Conference on Learning Representations* https://openreview.net/forum?id=S1eZYeHFDS (2020).
27. Glantz, S. A., Slinker, B. K. & Neilands, T. B. Primer of Applied Regression and Analysis of Variance. 3rd edn (McGraw-Hill Education, 2017).
28. Valipour, M., You, B., Panju, M. & Ghodsi, A. Symbolicgpt: a generative transformer model for symbolic regression. *arXiv preprint arXiv:2106.14131* (2021).
29. Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A. & Parascandolo, G. Neural symbolic regression that scales. In *Proc. International Conference on Machine Learning* 936–945 (PMLR, 2021).
30. Uy, N. Q., Hoai, N. X., O'Neill, M., McKay, R. I. & Galván-López, E. Semantically-based crossover in genetic programming: application to real-valued symbolic regression. *Genet. Program. Evol. Mach.* **12**, 91–119 (2011).
31. Keijzer, M. Improving symbolic regression with interval arithmetic and linear scaling. In *Proc. European Conference on Genetic Programming* 70–82 (Springer, 2003).
32. Krawiec, K. & Pawlak, T. Approximating geometric crossover by semantic backpropagation. In *Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation* 941–948 (Association for Computing Machinery, 2013).
33. Udrescu, S.-M. & Tegmark, M. AI Feynman: a physics-inspired method for symbolic regression. *Sci. Adv.* **6**, eaay2631 (2020).
34. Koza, J. R. Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**, 87–112 (1994).
35. Márquez-Neila, P., Fisher, C., Sznitman, R. & Heng, K. Supervised machine learning for analysing spectra of exoplanetary atmospheres. *Nat. Astron.* **2**, 719–724 (2018).
36. Fletcher, R. *Practical Methods of Optimization* (John Wiley & Sons, 2000).
37. Hopcroft, J. E., Motwani, R. & Ullman, J. D. Automata theory, languages, and computation. *Int. Ed.* **24**, 171–183 (2006).
38. Levine, S., Kumar, A., Tucker, G. & Fu, J. Offline reinforcement learning: tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643* (2020).
39. Kostrikov, I., Nair, A. & Levine, S. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations* https://openreview.net/forum?id=68n2s9ZJWF8 (2022).
40. Kumar, A., Zhou, A., Tucker, G. & Levine, S. Conservative Q-learning for offline reinforcement learning. *Adv. Neural Inf. Process. Syst.* **33**, 1179–1191 (2020).
41. Janner, M., Li, Q. & Levine, S. Offline reinforcement learning as one big sequence modeling problem. *Adv. Neural Inf. Process. Syst.* **34**, 1273–1286 (2021).
42. Dong, H., Nejjar, I., Sun, H., Chatzi, E. & Fink, O. SimMMDG: a simple and effective framework for multi-modal domain generalization. In *Thirty-seventh Conference on Neural Information Processing Systems* https://openreview.net/forum?id=RiSMijlsLT (2023).

Acknowledgements

The contributions of Y.T. were funded by the ETH grant no. ETH-12 21-1. The contributions of W.Z. and M.V. were funded by the Hasler Foundation no. 24019.

Author contributions

Y.T., W.Z., M.V., and O.F. contributed to conceptualization, formal analysis, and investigation. Y.T. and W.Z. developed the methodology. Y.T., W.Z., M.V., and D.H. contributed to experiments, programming. O.F. and D.K. supervised the project. All authors contributed to the writing of the paper.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-025-59288-y.

Correspondence and requests for materials should be addressed to Olga Fink.

Peer review information *Nature Communications* thanks Mojtaba Valipour and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.