Automated Scientific Discovery: From Equation Discovery to Autonomous Discovery Systems

Downloaded from: https://research.chalmers.se, 2026-09-05 02:06 UTC

Citation for the original published paper (version of record):

Kramer, S., Cerrato, M., Brugger, J. et al (2026). Automated Scientific Discovery: From Equation Discovery to Autonomous Discovery Systems. Machine Learning, 115(5). http://dx.doi.org/10.1007/s10994-025-06955-2

N.B. When citing this work, cite the original published paper.

research.chalmers.se offers the possibility of retrieving research publications produced at Chalmers University of Technology. It covers all kind of research output: articles, dissertations, conference papers, reports etc. since 2004. research.chalmers.se is administrated and maintained by Chalmers Library

(article starts on next page)

Machine Learning (2026) 115:109
https://doi.org/10.1007/s10994-025-06955-2

REVIEW

Automated Scientific Discovery: From Equation Discovery to Autonomous Discovery Systems

Stefan Kramer[1] · Mattia Cerrato[1] · Jannis Brugger[2] · Sašo Džeroski[3] · Ross D. King[4,5]

Received: 24 April 2025 / Revised: 8 September 2025 / Accepted: 12 December 2025
© The Author(s) 2025

Abstract
The paper surveys automated scientific discovery, from equation discovery and symbolic regression to autonomous discovery systems and agents. It discusses the individual approaches from a "big picture" perspective and in context, but also discusses open issues and recent topics like the various roles of deep neural networks in this area, aiding in the discovery of human-interpretable knowledge. Further, we will present closed-loop scientific discovery systems, starting with the pioneering work on the Adam system up to current efforts in fields from material science to astronomy. Finally, we will elaborate on autonomy from a machine learning perspective, but also in analogy to the autonomy levels in autonomous driving. The maximal level, level five, is defined to require no human intervention at all in the production of scientific knowledge. Achieving this is one step towards solving the Nobel Turing Grand Challenge to develop AI Scientists: AI systems capable of making Nobel-quality scientific discoveries highly autonomously at a level comparable, and possibly superior, to the best human scientists by 2050.

Keywords Scientific discovery · AI for science · Symbolic regression · Self-driving labs · AI scientist · Nobel Turing challenge

1 Introduction

The automated discovery of scientific knowledge has always been on the agenda of artificial intelligence research, and prominently so since the end of the 1970s (Langley, 1977; Langley et al., 1987). Scientific knowledge takes many forms: In many cases, the scientific process begins with collecting and classifying objects, and creating taxonomies of classes of objects. The more a scientific discipline advances, the more it tends to strive to describe the phenomena quantitatively, for better explanation and prediction. By far the most commonly used representation for describing systems of interest is in the form of mathematical equations, in particular differential equations. Thus, the automated discovery of equations

Editors: Riccardo Guidotti, Anna Monreale, Dino Pedreschi.

Extended author information available on the last page of the article

from data has been established as a family of methods within and partly outside artificial intelligence: it runs under the heading of equation discovery (Langley, 1977; Džeroski & Todorovski, 1993) as well as symbolic regression (Koza, 1994).

The goal in many application domains of equation discovery and symbolic regression is to learn a human-understandable model of the system dynamics in the form of (mostly ordinary) differential equations.[1] One important aspect of scientific discovery is that the resulting models need to be in principle interpretable.[2] Thus, the goal is not optimization (e.g., of properties in materials science or drug development), but to develop understanding.

An important part of the literature on automated scientific discovery (Langley et al., 1987; Li et al., 2021) discusses the topic from a cognitive science point of view (of what are or could be the reasoning processes leading to certain discoveries) and thus also a historical reconstruction of the processes. This is relevant, because today's AIs for scientific discovery also have to start from the same principles to enable discoveries in completely new application domains. While this can be viewed on the symbolic level only, many of today's approaches also consider the subsymbolic level to aid the process: Neural networks of various sorts can play a vital role in guiding the search, providing valuable information to the discovery agent, or turning low-level sensory information into high-level information that can be used for symbolic reasoning.

Finally, the question of autonomy of the discovery agents arises. While early systems assumed a table of input data is given by a human user, approaches with more autonomy on the side of the discovery agent are becoming more common. The approach became prominent with the development of the first robot scientist world-wide, Adam (King et al., 2009), that automated cycles of hypothesis generation and testing in the field of functional genomics. Meanwhile, the third generation of robot scientists is being developed. The degrees of autonomy of a discovery agent may range from completely passive, i.e., supervised learning, via active learning (Cohn et al., 1996) to reinforcement learning (Sutton & Barto, 2018).

Considering the above, this paper aims to give an overview of automated scientific discovery from a conceptual point of view, spanning the whole field from the generation of scientific knowledge, mainly in the form of equations, to automation and autonony in robot scientists or self-driving labs. It does not just enumerate approaches, but discusses central conceptual aspects and open issues that need to addressed in future systems. Particular attention is paid to the role of neural networks in the process: either for representation learning, for search in neural-guided equation discovery, or in neural operators, which abandon interpretability altogether. Discussing two main aspects of automated scientific discovery side by side in one paper, (i) the discovery of interpretable scientific knowledge in the form of equation on the one hand and (ii) automation/autonomy on the other (see Fig. 1), we identify a major research gap: systems that run autonomously, but are able to communicate results in formalisms used by scientists, so that interventions are possible, such as hints for search, the provision of goals and values, and the embedding of findings in bigger theories. There are few examples of such systems, but a relevant example is the pioneering work of Jan Zytkow, who coupled real electrochemistry experiments with the FAHRENHEIT system for equation discovery (Zytkow et al., 1990), and later proposed a robotic system for the rediscovery of Galileo's equation of objects rolling down an inclined plane, again

[1] The underlying data are most frequently temporal.

[2] If a model cannot be communicated to a community of researchers, it hardly qualifies as scientific, as communication is an indispensable part of the scientific endeavor.

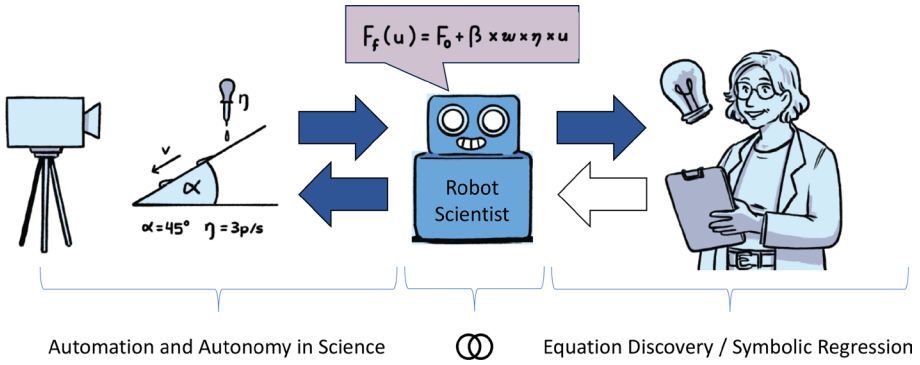


Fig. 1 Overview of the two realms of automated scientific discovery: (i) the discovery and communication of human-interpretable knowledge in a representation used by scientists in the field, e.g., equations (right-hand side) and (ii) autonomy and automation in science (left-hand slide). Approaches integrating both are currently rare

with the help of FAHRENHEIT, but already taking into account empirical error (Huang & Zytkow, 1997).

The paper is structured as follows: In Sect. 2, we will review equation discovery and symbolic regression from the beginnings to the current state of the art, with a list of open problems. In Sect. 3, we discuss the representations used in current scientific discovery and, in particular, how neural networks can be employed to learn representations for the discovery process and how dynamics can be learned directly by neural operators. The topic of Sect. 4 is closed-loop scientific discovery, with recent progress in the field. Sect. 5 discusses different levels of autonomy. An overview of causal discovery for scientific discovery and benchmarks and testbeds is given in Sect. 6, and Sect. 7, before we conclude in Sect. 8.

The survey paper is different from existing papers in many ways: Makke and Chawla (2024) presented a thorough survey of symbolic regression (SR) and equation discovery (ED). Our survey covers *both* SR/ED and automation/autonomy, so it is broader in scope. Also, it appears more conceptual and with a stronger focus on interesting open issues. Further, in the present paper the discussion of the various uses of neural networks appears both more extensive and deeper. In a recent study, Musslick et al. (2025) discuss primarily the limitations of autmated scientific discovery, with a focus on societal and ethical implications (e.g., the value alignment of robot scientists with human scientists). It discusses what should not be done, but also what potentially cannot be done. The latter is, of course, harder to argue, as it involves a forecast of the further progress of the field of artificial intelligence in general. Arguments likes the paradox of automation hold, others concerning the computational complexity of scientific discovery require more investigation. Another recent survey by Gao et al. (2024) focuses on life sciences exclusively and discusses everything in terms of agentic AI, which is both not our emphasis here. Two recent papers by Langley (2022, 2024) are both related, but at the same time different. The first of them (Langley, 2022) discusses the so far distinct notions of "agents of exploration" and "agents of discocery". Langley argues for a synthesis of the two, such that agents can both explore and discover in remote areas, like in space or in the deep sea. Although conceptually relevant (imagine a versatile scientific agent that explores a lab environment and discovers new concepts and laws along the way), the main thrust of the paper is clearly different. In the more recent

paper Langley (2024), Langley describes an integration effort different from the one shown above: In the paper, he envisages a tight integration and coupling of the various phases of scientific discovery, from the discovery of taxonomic knowledge via qualitative models to quantitative and causal models. It is argued that this integration is important, but has not been achieved before. We believe that, while interresting, this is of a different nature than the integration between the discovery of scientific, human-interpretable knowledge, and automation and autonomy in robot scientists or self-driving labs (see Fig. 1).

2 From BACON to Modern Equation Discovery and Symbolic Regression

2.1 History and Current Approaches

The first system for the discovery of equations based on data was BACON by Langley (1977). The first version of BACON was developed into a series of following systems, BACON.2 to BACON.5, with increasingly complex functionality (Langley et al., 1987). The basic philosophy behind the book by Langley et al. was that scientific discovery, even in its most intricate ways, is essentially problem solving. This even applies to the search for new problems, new representations, and new measurement devices. In the case of the BACON systems, the idea was applied to the discovery of equations.

BACON.1 to BACON.5 were implemented on the basis of PRISM, a system for the representation and inference of production rules. PRISM's production rules are meta-level discovery heuristics—they look like standard if–then rules, but instead of solving domain problems, they guide the search for scientific laws by telling the system which hypotheses to generate when particular data patterns appear. The actual search was a heuristic best-first search. PRISM was implemented in LISP.

The BACON systems relied on the observation of the correlation of pairs of variables, when everything else is being held constant (ceteris paribus). This is a strong assumption, as it will in many cases not be possible to control all other variables in an experiment. Also, interestingly, BACON has a flavor of active learning, since users are requested to record data, if they are not available yet. One interesting feature of BACON is the construction of new terms, e.g., ratios or products of existing terms, by production rules. In this way, it takes advantage of the structure of the search space, which is rarely ever attempted in current systems. Noise handling is achieved by some tolerance parameter, which establishes that a value of a variable (constructed or initially given) is constant, even though it varies within a certain range. BACON.2 to BACON.5 included advanced features for common divisors and symmetries, amongst others. Common divisors enable the discovery of ratios of quantities as ratios of integers, as commonly found in natural laws [see the book by Langley et al. (1987), p. 160]. Symmetries enable the discovery of symmetric equations like $n_1 \sin\theta_1 = n_2 \sin\theta_2$, one form of Snell's law [see Langley et al. (1987), p. 170–178].

Figure 2(a) shows the derivation of Kepler's third law $D^3/P^2 = k$ by a sequence of newly constructed terms, until a—more or less—constant value is found: The search first introduces *Term*1 as D/P and adds it to the table. Then it heuristically finds that the multiplication of *Term*1 with D may be beneficial, and adds this as $Term2 = D^2/P$ to the table. Finally, it multiplies *Term*1 and *Term*2 to obtain $Term3 = D^3/P^2$. In the final step, *Term*3

Moon	Distance (D)	Period (P)	Term1 (D/P)	Term2 (D^2/P)	Term3 (D^3/P^2)
A	5.67	1.769	3.203	18.153	58.15
B	8.67	3.571	2.427	21.036	51.06
C	14.00	7.155	1.957	27.395	53.61
D	24.67	16.689	1.478	36.459	53.89

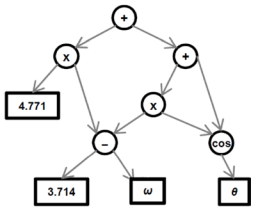
(a)

$$
\begin{array}{lcl|l|l}
E & \rightarrow & E+F & E-F & F \\
F & \rightarrow & E*T & F/T & T \\
T & \rightarrow & const[:0:25:70] & V & (E)
\end{array}
$$

(b)

$$E \rightarrow E+V\ [p] \mid V\ [1-p]$$
$$V \rightarrow x\ [q] \mid y\ [1-q],$$

(c)



(d)

Fig. 2 **a** BACON (Langley, 1977; Langley et al., 1987) **b** Example of context-free grammar guiding the search for equations in the Lagramge system (Todorovski & Džeroski, 1997) **c** A probabilistic context-free grammar as used in ProGED (Brence et al., 2021) **d** Symbolic regression (Schmidt & Lipson, 2009)

is deemed "constant" according to some tolerance parameter, derived from some historical measurement error, and the final law is obtained.

BACON rediscovered more than a dozen established scientific laws, however, it was not yet the time to apply it to complex real-world data and problems. The achievement was to define computational scientific discovery as a field, and equation discovery and symbolic regression as a task within that field. The book by Langley et al. (1987) is great summary of the state-of-the-art at that time and a treasure trove of ideas for solutions to problems that are still existent.

The next generation of equation discovery systems was not restricted to keeping all but a pair of variables fixed, but was able to handle observational data. In the philosophy of science, "observational data" refers to data that are obtained by mere observation, without control variables or interventions in the system of interest. BACON still required controlled conditions under which simple laws could be rediscovered. The new generation was able to overcome this, but also to learn models of dynamical systems in the form of ordinary differential equations (ODEs). The Lagrange system (Džeroski & Todorovski, 1993) computes all derivatives up to a pre-defined order, then generates products of up to a maximum of variables, before it calculates a simple linear regression to generate a candidate equation. More recently, this approach has been taken up in the SINDY system (Brunton et al., 2016), which applies sparse (instead of simple) linear regression. In the meantime, the method has been extended to capture nonlinear dynamics by shallow recurrent decoder networks (SINDy-SHRED) (Gao et al., 2025).

The successor of Lagrange, named Lagramge (Todorovski & Džeroski, 1997), was a milestone in equation discovery, as it introduced the use of domain knowledge in addition to data: It was the first system to use a context-free grammar (CFG) to guide the search for systems of equations. Grammars are a way to use prior knowledge and let that knowledge guide the search for equations. For instance, one can specify that certain groups of terms have to occur together in an equation – so it is possible to provide meaningful building blocks of equations. Practically, CFGs for equation discovery have to be developed by computer scientists and domain experts together, because domain experts alone may not be familiar with grammars for formal languages. Lagramge generates all equations according to the grammar up to some user-defined depth limit, and then fits them as Lagrange did. Using grammars, Lagramge was able to solve problems that the predecessor Lagrange

was not able to solve, for instance, the problem of two poles on a cart. More importantly, Lagramge was successfully applied to ecological modeling, in particular to phytoplankton modeling (Todorovski et al., 1998). An example CFG for Lagramge is shown in Figure 1(b). Lagramge GSAT (Ganzert et al., 2010) improves Lagramge by a bundle of modifications: first, it uses a search procedure similar to GSAT (random restart hillclimbing) to randomize search; further, it employs a one-step look-ahead and a momentum to make the search less erratic. The application was to develop a model of the mechanically ventilated lung. Washio and Motoda (1997) further improved the methods by also taking into account units and scale types as constraints. Dimensional units are also considered for use in ProGED (Brence et al., 2021), which is based on the idea of using probabilistic CFGs to represent the search space and sample from it. An example is given in Fig. 2(c), where both the rules and the probabilities associated with the rules (p and q) are shown. These probabilities can be fixed, but can also be learned from corpora of equations (Chaushevska et al., 2022). Sampling candidate equations from probabilistic CFGs enables easy parallelization: batches of sampled equations can be distributed to nodes and evaluated in an embarrassingly parallel way.

Symbolic regression, a development parallel to the development of equation discovery, was originally based on genetic programming (GP): the term was introduced by Koza (1994). Typical systems of symbolic regression work on an operator tree or DAG representation of equations (see Fig. 2(d)). These trees are modified by a set of possible operations, such as crossover between subtrees of two trees (equations), mutations, substitutions of variables by constants, or, vice versa, substitutions of constants by variables. Schmidt and Lipson (2009) used symbolic regression to discover natural laws from measured data. Symbolic regression approaches were used early on to discover ODEs (Džeroski & Petrovski, 1994) and used ideas from grammar-based genetic programming to consider domain-specific knowledge, paving the way for systems that use both data and domain knowledge in equation discovery, such as Lagramge, Lagramge2.0 (Todorovski & Džeroski, 2006), IPM (Bridewell et al., 2008) and Prob-MoT (Čerepnalkoski et al., 2012). The last three use process-based formalism to represent models and domain knowledge.

The Bayesian machine scientist (BMS) (Guimerà et al., 2020) establishes the plausibility of models using explicit approximations to the exact marginal posterior over models and establishes its prior expectations about models by learning from a large empirical set of mathematical expressions. The space of equations is explored via Markov Chain Monte Carlo (MCMC), with specific moves for mathematical expression sampling.

PySR by Cranmer (2023) is a fast, effective and popular approach to symbolic regression. It is based on genetic programming and outputs one solution per complexity class (from simple to complex equations). PySR is frequently found to be well-performing in practice. It has a Python front-end and delegates numerical computations to Julia. Using Julia "under the hood" and heuristics to avoid redundancy, it is able to explore a large number of candidates in a relatively short period of time, giving it a competitive advantage in many situations. In the meantime, version 1.4 of PySR is available with template expressions and version 1.5 with mini-batching, which further improves practical applicability.

Recent work by Boris Krämer and collaborators (Bychkov et al. 2024) has advanced the use of quadratic models for data-driven discovery of dynamical systems governed by partial differential equations (PDEs). In particular, they explore transformations of nonlinear PDEs into quadratic form, which enables the application of structure-preserving reduced-order modeling and symbolic regression techniques. The approach facilitates the use of quadratic

latent variable models that retain interpretability and allow for efficient training on noisy and sparse data. The usefulness of the approach has been demonstrated in areas such as fluid dynamics and plasma modeling.

Symbolic regression and equation discovery are currently limited to systems with only few variables. Jiang and Xue (2023) address this problem by identifying control variables, which can be varied to discover the dynamics of a system in "controlled experiments" step by step. The approach is still based on genetic programming. A precondition of its use is evidently the existence of such variables, which is not always the case. In practical applications and real systems, the set of control variables is not equal to the set of variables that should appear in an equation. Thus, that mapping has to be learned first. Nevertheless, the idea of actively using control variables to reduce complexity is valuable and could be a key to making ED/SR practically applicable to large and complex sytems.

In recent years, a new field of research has emerged that focuses on how neural networks can be used in equation discovery. To provide an overview, we divide the work into three categories. The categories are: (i) NNs as a supporting module in the equation discovery system (EDS), (ii) NNs as the main component of the EDS, and finally, (iii) foundation models as EDS. We discuss the three categories in consecutive order.

AI Feynman 2.0 (Udrescu et al., 2020) is a recent symbolic regression approach that aims to improve its predecessor (a) by structuring the search space by building the equation in meaningful increments and (b) making it more noise-tolerant. The first goal is achieved by graph modularity, i.e., constructing the equations by so-called graph modules. It should be noted that, in doing so, it is one of the few approaches that takes advantage of the structure of the search space (instead of just brute-force search, sampling or "blind" randomized traversal). The second goal is achieved by employing an MDL-inspired evaluation function instead of the RMSE. This function is called MEDL in Feynman 2.0. Using MEDL, effective pruning can be developed, because the complexity of the equation can be balanced against its error. Lusch et al. (2018) apply an auto-encoder structure to find a coordination transformation for a differential equation that maps the nonlinear original problem to linear embedding. Following the idea of an autencoder, Mežnar et al. (2023) embed equations in a low-dimensional latent space and use this smooth latent space to suggest new equations based on genetic programming. Mundhenk et al. (2021) use a Recurrent Neural Network (RNN) to seed a genetic programming algorithm, and the genetic algorithm results are used to train the RNN. While the work discussed so far uses a subsymbolic component to simplify the original problems, the following articles use neural networks as main component.

Deep Symbolic Regression (DSR) (Petersen et al., 2021) addresses the problem of GP approaches with finding solutions for larger problems. It employs a recurrent neural network to build an equation tree step by step. As the objective function (of fitting a low-error equation) is not differentiable, a reinforcement learning approach is proposed. More specifically, DSR employs a risk-seeking policy gradient, which maximizes the best-case performance, not the average-case performance. NeSymReS (Biggio et al., 2021), SymbolicGPT (Valipour et al., 2021), and E2E (Kamienny et al., 2022) use a transformer-based architecture to predict the equation on a token level. The main difference is the embedding architecture of the data set. MGMT (Brugger et al., 2025) compares different embedding methods and shows their influence on the prior of the guiding network. Additionally, the work shows that supervised learning is beneficial compared to reinforcement learning for the architectures considered. TPSR (Shojaee et al., 2023) and DGSR-MCTS (Kamienny et al., 2023)

combine a transformer-based architecture with a Monte Carlo Tree Search (MCTS). In the second paper, the network suggests how to mutate the current equation. Another approach is to train a specialized end-to-end differentiable network and parse it after the training with gradient descent to an equation. $EQL^{\div}$ (Sahoo et al., 2018) or Kolmogorov Arnold networks (KAN) (Liu et al., 2024) are examples for this approach.

Large language models (LLMs) have also impacted the field of equation discovery. Foundation models such as GPT-4 have the advantage that after the initial learning, they only need to be adapted to the equation discovery domain through fine-tuning or prompt design. In addition, they have been shown to retain background knowledge from the initial training, and the user can add domain knowledge through prompts. In-Context Symbolic Regression (ICSR) (Merler et al., 2024) and the system by Sharlin et al. (Sharlin & Josephson, 2024) employ a foundation model to produce initial equations. These equations are then tested on the data set. The fitness score and other measures, such as complexity, are calculated externally and then fed back to the model with the task of refining the solutions. LLM-SR (Shojaee et al., 2024) follows the same idea, but represents equations as programs and uses comments in the program to make the discovered equation better understandable. Meyerson et al. (2023) use a foundation model to perform genetic programming (mutation, crossover, etc.) through prompts. Foundation models are also used for specific types of dynamical systems (Berghaus et al., 2024) and imputation in such contexts (Seifner et al. 2025), but without giving explicit representations in the form of equations. Overall, foundation model-based equation discovery systems show promising results, but the extent to which the initial training influences the test results has not yet been sufficiently investigated, as the standard benchmarks (see below) have been included in the initial training.

2.2 Open Problems

In equation discovery and symbolic regression, some significant open problems require further research. References and summaries of useful partial solutions can be found in Tables 1 and 2.

- It remains hard to exploit structure in the space of equations to guide the search to promising parts of the search space. Opportunities for pruning would also be helpful. The early BACON systems (Langley et al., 1987) structured the space by introducing additional, more complex terms for further use in equations found later. AI Feynman 2.0 (Udrescu et al., 2020) introduced the notion of *graph modularity* to decompose complex functions into simpler ones. RED (residuals for equation discovery) (Brugger et al., 2025) reformulates the symbolic regression problem by transforming the expression tree such that only the residuals of the current expression need to be explained. Other approaches use predefined patterns to dynamically weight parts of the search space in reinforcement learning (Xu et al., 2024) or actively query the data generation process for informative data points (Jin et al., 2023) or employ control variables to disentangle the dependencies (Jiang & Xue, 2023, 2024).
- Most approaches struggle with a dimensionality of the problem higher than a very small number of variables. Earlier methods employed classical preprocessing and dimensionality reduction techniques such as PCA (Zhong et al., 2021) and feature selection (Chen et al., 2017; Al-Helali et al., 2020, 2024; Neshatian & Varn, 2017) to address the

Table 1 Open problems of symbolic regression by category (part I) and existing approaches to address them

Category	Refs.	System Name/Comments
Structuring the Search Space		
Term Generation	Langley et al. (1987)	BACON
Graph Modules	Udrescu et al. (2020)	AI Feynman 2.0, graph modularity
Residual-Based	Brugger et al. (2025)	Residuals for Equation Discovery (RED)
		rewrites expression tree to solve for residuals
Predefined Patterns	Xu et al. (2024)	policy-driven structuring by predefined patterns
Active Learning	Jin et al. (2023)	queries system where expressions disagree
Control Variables	Jiang and Xue (2023)	experiments with control variables for SR
	Jiang and Xue (2024)	optimized experiments with control variables
Handling High Dimensionality		
Classical Preprocessing	Zhong et al. (2021)	PCA-based
Feature Selection (Hard or Soft Removal)	Chen et al. (2017)	wrapper, inner and outer loop of GPs
	Al-Helali et al. (2020)	feature and instance selection for imputation
	Al-Helali et al. (2024)	feature impact on model level
	Neshatian and Varn (2017)	detection of "feature bundles", GP-based
Attribution Methods	Wang et al. (2025)	Shapley values
Weighting Primitives	Huang et al. (2020)	adaptive weighting of primitives
and Productions	Ali et al. (2022)	GP-based, ranking of productions of grammar
Problem Reformulation	Kahlmeyer et al. (2025)	variable substitutions to reduce search space
Overfitting Avoidance and Regularization		
Evolutionary Complexity Control	Sambo et al. (2021)	evaluation time as a measure of complexity
Statistical and	Kubalik et al. (2016)	LASSO-based
Bayesian Regularization	Brunton et al. (2016)	SINDy, sparse regression (not LASSO)
	Bomarito et al. (2022)	Bayesian model selection for reduced bloating
Reducing Constants	de Franca and Kronberger (2023)	rewriting equations to reduce constants to be fit
Integrated Denoising	Sun et al. (2025)	integrating noise gating module into RL for SR

Table 2 Open problems of symbolic regression by category (part II) and existing approaches to address them

Category	Refs.	System Name/Comments
Incorporating Prior Knowledge and User Interaction		
Grammars and Process Models	Džeroski and Todorovski (1993)	Lagrange
	Todorovski and Džeroski (1997)	Lagramge
	Brence et al. (2021)	probabilistic context-free grammars
	Bridewell et al. (2008)	IPM
Units,	Washio and Motoda (1997)	consideration of scale types and identities
Dimensional	Udrescu et al. (2020)	Buckingham Π theorem for dimensionless variables
Analysis/	Villar et al. (2023)	encodings for dimensionless machine learning
Constraints	Brence et al. (2023)	attribute grammars to ensure consistency
Library	Schmidt and Lipson (2009)	mentions learning of function libraries
Learning	Ellis et al. (2018)	EC^2, neurally guided program synthesis
	Grayeli et al. (2024)	concepts also represented as text for LLMs
Logical	Cornelio et al. (2023)	AI-Descartes, consistency with logical axioms
Constraints	Prieschl et al. (2019)	ontologies for feature construction
Shape	Haider et al. (2023)	shape constraints (e.g., monotonicity or convexity)
Constraints	Haider et al. (2022)	optimistic vs. pessimistic shape-constrained SR
Physics-Aware SR	Kubalík et al. (2021)	generation of constraint samples, GP-based approach
	Kubalík et al. (2023)	generation of constraint samples, NN-based approach
	Bendinelli et al. (2023)	neural symbolic regression with hypothesis (NSRwH)
		from hypothesized properties to conditioning tensors
	Fox et al. (2024)	checking thermodynamic constraints by SymPy
		adjusting score functions for PySR resp. BMS
Human-in-the-Loop	Tian et al. (2025)	interactive symbolic regression
SR and Visualization	Kim et al. (2020)	for DSR, visualizes: rewards, complexity
		possibility of diagnostics
	Crochepierre et al. (2022)	user chooses preferred expression / suggests expression

problem. Also, attribution methods from explainable AI (Wang et al., 2025) have been proposed. More advanced methods learn to weight the primitives that can be used in an equation (Huang et al., 2020) or to rank the production rules of a grammar (Ali et al., 2022). An elegant approach aims to discover suitable variable substitutions (Kahlmeyer et al., 2025) to split the problem into subproblems of lower dimensionality, which are easier to solve.

- Overfitting avoidance and regularization: The syntactic complexity of an equation does not necessarily correspond to the complexity of the function in the feature space. Current approaches to the avoidance of overfitting are: Complexity control built into the evolutionary search itself (Sambo et al., 2021), equating the elapsed time with complexity, and statistical and Bayesian regularization (Kubalik et al., 2016; Brunton et al., 2016; Bomarito et al., 2022). More recent approaches use rewriting techniques to reduce the number of constants to be fit (de Franca & Kronberger, 2023) (much like the reduc-

tion of variables discussed above (Kahlmeyer et al., 2025)) or a noise gating module that is tightly integrated with a reinforcement learning agent (Sun et al., 2025). However, more and more meaningful ways to approximate or bound complexity would be helpful.

- Relating discovered equations to existing theory or making the equations consistent with it remains a big challenge. Consistency with current theory can be ensured by user-specified grammars (Džeroski & Todorovski, 1993; Todorovski & Džeroski, 1997; Brence et al., 2021) or process models (Bridewell et al., 2008). Proper scale types and dimensions can be enfored algorithmically, e.g., via dimensional analysis (Udrescu et al., 2020; Villar et al., 2023) (e.g., applying the Buckingham Π theorem) or annotated grammars (Brence et al., 2023). Library learning is way to make the system constantly improving, by collecting more and more useful building blocks of equations (Schmidt & Lipson, 2009; Ellis et al., 2018; Grayeli et al., 2024). Logical (Cornelio et al., 2023; Prieschl et al., 2019) and shape (Haider et al., 2022, 2023) constraints are useful to restrict what is being learned based on domain knowledge. Physics-aware approaches either generate so-called constraint samples that are used in the fitting process (Kubalík et al., 2021, 2023) or integrate prior knowledge more directly, either to condition the neural networks (Bendinelli et al., 2023) or in the loss function (Fox et al., 2024). Several approaches enable or employ a human-in-the-loop approach to integrate user feedback (Tian et al., 2025; Kim et al., 2020; Crochepierre et al., 2022). Ultimately, it is not clear whether or how an "understanding of the physical meaning" of variables can be achieved.
- At least in the case of differential equations, fitting the model is the most expensive part. Ways of stopping the fitting process if it turns out to be unpromising would save a lot of computation time. Some systems like PySR employ heuristics to abandon equations that are unpromising early on, but it is usually considered an implementation detail and not the topic of algorithmic development.
- Equations are "brittle": properties of differential equations can change dramatically with only little syntactic modifications. Minor changes can lead to no solutions, one solution, or many solutions. Ironically, research on symbolic regression usually does not care whether the discovered differential equations have a solution—the focus is on the fitting of an explanatory model.
- For the approaches based on foundation models, it is unclear how the results can generalize to new, previously unseen problems. Data provenance is an issue: It is unclear whether the models have seen some of the equations before in training. Many of the approaches are based on embeddings of datasets. It is, at this point, not clear, what the best way is to embed a dataset for a foundation model for symbolic regression.

3 Representation Learning in Scientific Discovery

3.1 Representation Learning of the Input

The standard representation of data for scientific discovery is tabular data (see, e.g., also the tables in the book by Langley et al. (1987) and Figure 1(a)). However, recent years have seen a surge of papers that use neural networks as an intermediate representation to aid in the discovery of models.

One notable example is the work of (Cranmer et al., 2020), who proposed Graph Neural Networks (GNNs) as an intermediate representation. GNNs were used to learn about the interaction of objects, in terms of, for example, forces that act upon each other. Classical examples include n-body problems or, more specifically, orbital mechanics—the motion of planets and other larger objects in our solar system. The nodes in the graph represent the objects, which are annotated by feature vectors representing the properties of the objects. The edges in the graph represent the interactions between the objects and are annotated by properties that partially depend on those of the objects. For example, one may consider the masses of planets as properties of the nodes, and the distance and gravitational force between the objects as properties of the edges. When learning GNNs, typically, so-called node models ϕ_v are updated depending on the edge models ϕ_e of neighboring edges and, alternatingly, the edge models ϕ_e are updated based on the node models ϕ_n of the nodes that the edges connect. Update steps are frequently framed as message passing, and pooling functions aggregate input from multiple edges connected to one node. GNNs usually can be trained end-to-end, but are not guaranteed to converge. For more technical details, we have to refer to the original paper (Cranmer et al., 2020), especially the architecture on p. 3 of that article (Fig. 3).

In the application domain that was given as an example, orbital mechanics, the input to the system are (x, y, z) coordinates of the Sun, all planets, and all moons with a mass above 10^{18} kg. Data from 1980 to 2013 were used with time intervals of 30 min each, with the first 30 years for training and the last 3 years for validation.

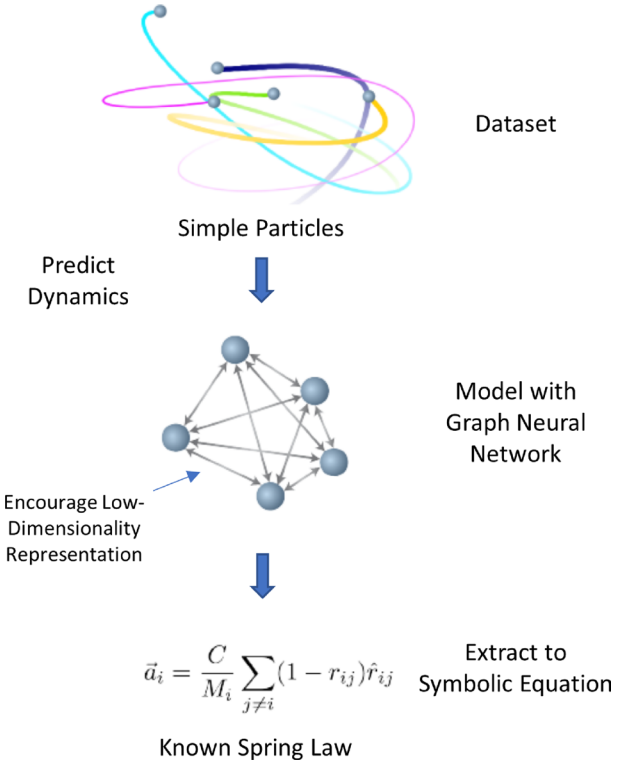


Fig. 3 Workflow of Cranmer et al. (2020): GNNs are used in the intermediate step to support or enable the learning process

(Garcon et al., 2022) proposed a method to predict known physical parameters and discover new ones from oscillating time series (Fig. 4). The method is trained on a large set of synthetic time series. The latent parameters used to generate the monochromatic sine waves are the carrier frequency, F_c, and phase ϕ (which is mapped for technical reasons to two separate parameters, $\sin(\phi)$ and $\cos(\phi)$), in addition to the coherence time τ. The AM and FM sine waves are generated by adding a modulation function to the carrier. The modulation function's latent parameters are the modulation frequency F_m and amplitude I_m. Noise is linearly added to the pure signals by sampling the Gaussian distribution. AM/FM signals with minimum I_m reduce to decaying monochromatic sine waves and reach 100% modulation with maximum I_m. These latent parameter ranges are wide enough such that they would encompass many foreseeable real-world signals. Figure 3 shows the neural network architecture used to predict the latent parameters, with an autoencoder-type subnetwork to

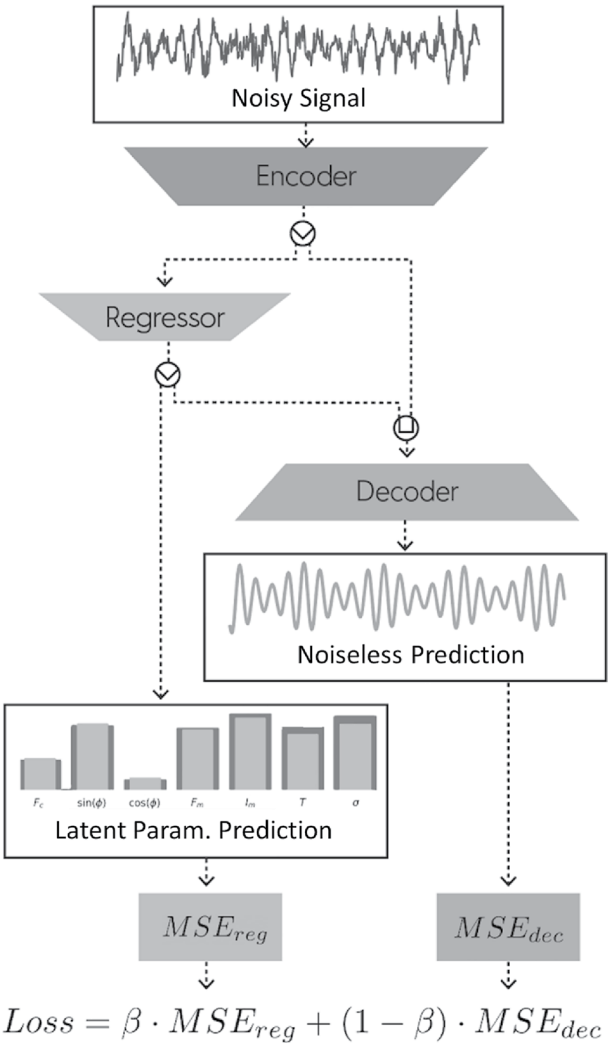


Fig. 4 Neural network architecture of model that extracts known and unknown physical parameters from oscillating time series Garcon et al. (2022)

support the prediction. The method can be used to discover new parameters (not just predict known ones) and reconstruct equations producing input time series.

The situation is clearly more complex when the observations are given as videos instead of tabular data. Chen et al. (2022) presented a solution based on what they call neural state variables. Neural state variables are essentially latent variables. The current state-of-the-approach to computing latent variables is to define an autoencoder with a bottleneck layer of the right dimension. The dimension should be large enough to allow faithful reconstruction by the decoder, but small enough so that the latent variables are non-redundant. The goal of the proposed method is to have the number of dimensions (i.e., the number of neural state variables) as close as possible to the degrees of freedom of the observations in the videos. In technical terms, the number of dimensions should be close to the so-called intrinsic dimension (ID), which is the minimum number of independent variables needed to fully describe the state of a dynamical system. Various methods from manifold learning, for instance the one by Levina and Bickel (2004), are known to efficiently calculate an estimate of the intrinsic dimension. It would be tempting to calculate the intrinsic dimension for the videos and then use it as the bottleneck size of an autoencoder to come up with the latent variables. However, practically, information becomes blurry at much larger bottleneck sizes than the ID already. Therefore, Chen et al. take a two-step approach and define two autoencoders, one regular and one that maps the latent variables of the first to further ID latent variables. These are the neural state variables that can be used for downstream analysis. The approach has not yet been made explainable for scientific discovery.

Summing up the above, neural networks are used in this domain for

- making the data sparse in the sense of removing small to negligible interactions (Cranmer et al., 2020),
- a change of representation (e.g., from coordinates to distances depending on some variables (Cranmer et al., 2020)),
- data augmentation (to sample arbitrarily large data from the neural network and also smoothen the data in that way (Cranmer et al., 2020; Li et al., 2021)),
- the prediction of important parameters to be used in equations directly (Garcon et al., 2022), and
- extracting latent variables from low-level input representations (e.g., neural state variables from videos (Chen et al., 2022)).

3.2 Representation Learning of the Dynamics

Neural operators (Kovachki et al., 2023) can learn to map the current state of a system to the next state. This can be done for systems that evolve over space or time and especially for systems for which partial differential equations (PDEs) are too difficult to solve. Neural operators are, however, not restricted to mapping from one state to the next over time: They can learn general functional mappings between various types of inputs and outputs, e.g., inititial conditions to solutions or, even more generally, function-to-function mappings [like DeepONet (Lu et al., 2021) or Fourier Neural Operators (Li et al., 2021)]. The latter learn mappings between functions, not just states over time, for instance, they can map a boundary condition (a function) to a solution (another function), which might involve non-temporal variables. Advantages are, amongst others, speed and flexibility (they are not hard

to apply from one problem to the next). Neural operators like DeepONet or Fourier Neural Operators are, like other neural networks, black-box models.

3.3 Representation Learning of Chemical and Biological Objects

No representation learning section for the sciences would be complete without mentioning efforts to learn latent representations of chemical or biological entities, like proteins and small molecules. We discuss three well-known examples for proteins and small molecules that focus on latent representations with good predictive performance down-stream, without caring for explainability.

ESM-1b (Rives et al., 2021) is a large masked-LM trained on 250 M sequences. Its embeddings capture multi-scale biological signal and deliver strong downstream prediction (e.g., mutational effects, secondary structure) without requiring multiple sequence alignments (MSAs). For ProtTrans (Elnaggar et al., 2022), a family of Transformers (BERT/T5/XLNet/Transformer-XL) was pretrained on up to 393B amino acids. The released checkpoints (e.g., ProtT5) are broadly adopted in benchmarks and applications. Finally, UniRep (Alley et al., 2019) is an mLSTM trained on 24 M UniRef50 sequences. The fixed-length sequence embeddings enable competitive prediction of structure and function and enable low-N protein engineering after light task-specific tuning.

D-MPNN (Yang et al., 2019) established a strong supervised baseline via directed bond message passing and a massive public/industrial benchmark suite. GROVER (Rong et al., 2020) scaled self-supervised pretraining to 10 M molecules with a graph Transformer/MPNN hybrid, delivering consistent fine-tuning gains. Graphormer (Ying et al., 2021) showed that Transformers with the right structural encodings can reach state-of-the-art performance on graph/molecular leaderboards.

3.4 Open Problems

The biggest open problem of representation learning for scientific discovery may be to obtain latent representations that are interpretable: It is currently not well understood how learned representations can be aligned with representations that can interpreted by humans or related to existing theories. An overview of proposed solutions is shown in Table 3:

- While neural operators can find accurate approximations to the solution of a PDE, understanding how they arrived at that solution is not straightforward. A few articles outline a progression toward human-comprehensible latent dynamics: Neural-operator surrogates (Zhou et al., 2025) cast latent variables as physical parameters, making system responses easy to visualize and sensitivity sweeps computationally cheap. Invariant-constrained models (Zhang et al., 2023) shape latent geometry so trajectories lie on conserved manifolds, allowing flows to be visualized and robustness tested under perturbations. Latent-dynamics approaches (Wang et al., 2024) then learn reduced state coordinates with physically valid transitions, producing transition graphs for checking stability and sensitivity. Finally, aligning latent spaces with human concepts (Zhang & Lipson, 2024) emphasizes parameters, invariants, and states as natural bases, ensuring that learned representations can be both visualized and interrogated in ways that foster scientific insight.

Table 3 Open problems of representation learning for scientific discovery

Category	Refs.	System Name/Comments
Explainable Latent	Zhou et al. (2025)	neural operators as surrogates for simulations
Representations	Zhang et al. (2023)	PIANO, learns invariants as latent
for Neural Operators		structures, by cropping input regions
	Wang et al. (2024)	discretized latent states and constraints on state transitions
	Zhang and Lipson (2024)	alignment with human-interpretable latent structures
Explainable Latent	Du et al. (2022)	VAEs with monotonicity constraints
Representations for	Kirchoff et al. (2024)	SALSA, based on contrastive loss
Small Molecules	Haddad et al. (2025)	RL to ensure smoothness in latent space

- Computational biologists are just beginning to adopt the same "interpretable latent" mindset that physics-inspired symbolic regression has already embraced. Current protein latent spaces are mostly only post-hoc interpretable. The situation is different for small molecules, for which good solutions for interpretable latent spaces exist: The variational autoencoder by Du et al. (2022) was made to fulfill monotonicity constraints from the application domain. The SALSA system by Kirchoff et al. (2024) builds on contrastive loss to achieve latent representations that preserve important structural and physico-chemical properties. Haddad et al. (2025) recently proposed a reinforcement learning approach to ensure smoothness in latent space.

4 Closed-Loop Scientific Discovery

4.1 Main Concepts, History and Advantages

The cutting edge of applying AI to science are "AI Scientists" (aka "Robot Scientists", "Self-driving Labs", "Autonomous Discovery systems", "Machine Scientists", etc.). These AI systems area capable of the closed-loop automation of scientific research. AI Scientists were named in 2025 by Nature as the "number one technology to watch" (Eisenstein, 2025). AI Scientists automatically originate hypotheses to explain observations (abduction/induction), devise experiments to test these hypotheses (deduction), physically run the experiments using laboratory robotics, analyze and interpret the results to change the probability of hypotheses, and then repeat the cycle. In other words, they aim to automate all or parts of the scientific method, as shown (simplistically) in Fig. 5. As the experiments are conceived and executed automatically by computer, it is possible to completely capture and digitally curate all aspects of the scientific process, making science more reproducible (King et al., 2009).

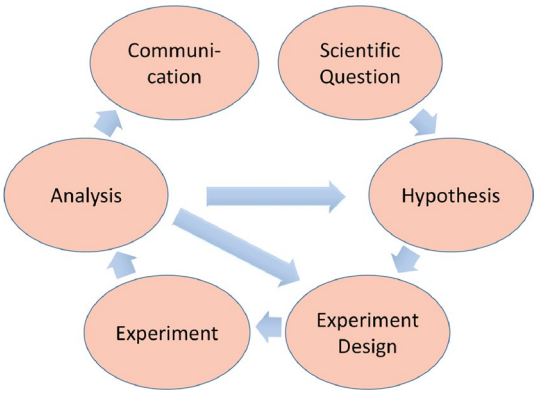


Fig. 5 Six steps of the scientific process

Fig. 6 The robot scientist Eve

The first contribution describing a largely autonomous system which discovered new knowledge was due to Ross D. King and his group (King et al., 2009), who developed the Adam robot scientist. Adam identified 6 genes encoding orphan enzymes in yeast (*Saccharomyces cerevisiae*), i.e., enzymes which catalyze reactions occurring in yeast for which the encoding genes were not known at the time. The system was provided with a freezer, liquid handlers, plate readers, robot arms, and further actuators, enabling yeast cultivation experiments lasting as long as 5 days. Yeast growth was measured via optical sensors. On the software side, Adam was provided with an extensive Prolog knowledge base describing known facts about yeast metabolism. Hypotheses were formed by abduction, enabled by a combination of bioinformatic software and databases, after which an experiment planning module was responsible for selecting metabolites to be inserted in the yeast's growth medium.

Another successful example of an AI Scientist is Eve (see Fig. 6). Originally developed for high-throughput drug screening (Sparkes et al., 2021), the system was then instrumental in discovering that several existing drugs could be repurposed to prevent tropical diseases (Williams et al., 2015). Most prominently, it found that an anti-cancer compound (TNP-470) could be employed against the parasite *Plasmodium vivax*, whose bite is the most frequent cause of recurring malaria. The system is able to hypothesize and test quantitative

structure–activity relationships (QSARs) via a combination of active learning and Gaussian process regression (GPR). GPR is employed to learn a QSAR f mapping the characteristics of compounds to a response variable indicating the strength of the biological activity; then, the obtained function f is employed as a noisy oracle to select K compounds out of a pool of possible candidates. Exploration and exploitation is balanced. This two-step process may be repeated until enough candidates are obtained. Following Adam and Eve, the third generation of robot scientists is being developed.

AI Scientists have a number of relevant advantages, besides being able to discover new knowledge in a way that may be less biased than a human scientist:

- Efficiency: AI Scientists are increasing the productivity of science. They can work cheaper, faster, more accurately, and longer than humans (Williams et al., 2015). They can also be easily multiplied.
- Reproducibility: Biomedical science is facing a "reproducibility crisis". AI Scientists have the potential to ameliorate this problem, as they describe experiments in far greater detail and semantic clarity than human scientists, and robots execute experimental protocols more accurately than human scientists (Roper et al., 2022).
- Robustness: The Covid-19 pandemic clearly demonstrated the vital importance of biomedical research and the critical need to maintain research continuity (Burger et al., 2020). AI Scientists should be able to keep up research and a research infrastructure even in times of crisis.

However, also disadvantages have been noted in the literature (Musslick et al., 2025): Issues like the automation paradox and questions concerning research evaluation and value alignment need to be addressed. Besides, it is unclear which role negative results will play in the new research landscape of AI Scientists.

In the meantime, AI Scientists are increasingly being applied to multiple scientific domains (ranging from quantum mechanics to astronomy, from chemistry to medicine). For an overview of example systems, see Table 4.

4.2 Open Problems

Three of the current main limitations of AI Scientists are (i) the design of novel experiments, (ii) integration with laboratory robotics, and (iii) the formation of completely new hypotheses and theories.

The central task that faces every experimental scientist is the design of novel experiments to test a hypothesis. The abstract problem is given (1) a hypothesis, and (2) a set of laboratory equipment, output (3) a protocol to test the hypothesis using the equipment. Relatively little AI research has focussed on this aspect of automating science. Note that this task is different in kind from the task of traditional "experimental design", it also different from deciding, from a set of given experiments, the most efficient (in terms of time/money) to test a set of hypotheses. In all the existing AI Scientists systems that we are aware of the types of experiment that can be executed are limited to a small stereotypical set. For the design of novel experiments to be possible it will be necessary to formalise general scientific knowledge, as well as knowledge about the functionality of laboratory equipment, and experimental protocols. It is also necessary to develop inference and planning engines to generate the

Table 4 Robot Scientists by Discipline, Name, and Country

Discipline	Name	Country
Drug Discovery	Eve	Sweden
Drug Discovery	Recursion	US
Drug Discovery	Lilly Life Sciences Studio lab	US
Drug Discovery	XtalPi	China
Chemistry	UK Centre for Rapid Online Analysis of Reactions	UK
Chemistry	roboRXN at IBM	Switzerland
Chemistry	phactor™	US
Chemistry	Pharmacy on Demand (PoD)	US
Chemistry	Molecule Maker Institute	US
Chemistry	AI-Chemist	China
Chemistry	A self-optimizing reactor	US
Chemistry	Chemputer	UK
Chemistry	Lapkin Group	UK
Chemistry	RoboChem	Netherlands
Materials	Kebotix	US
Materials	Autonomous Research System (ARES)	US
Materials	Robot Chemist	UK
Materials	Acceleration Consortium	Canada
Materials	Brookhaven	US
Materials	SARA	US
Materials	AI-Chemist	China
Materials	A-Lab	US
Materials	Matterhorn	UK
Materials	ARC – Exciton Science	Australia
Materials	Gormley	US
Catalysis	RealCat	France
Catalysis	SwissCAT+	Switzerland
Metallurgy	ACCMET	EU
Materials	BIG-MAP	EU
Cell Biology	Labdroids	Japan
Cell Biology	Murphy Lab	US
Mechanical Eng	Creative Machines Lab	US
Protein Design	Molcure	Japan
Protein Design	LabGenius	UK
Systems Biology	Genesis	Sweden
Materials/Biology	Argonne Autonomous Discovery	US
Quantum Physics	MELVIN	Germany
Medicine	Automation Science	Singapore

new experiments, as well as to develop compilers to translate generated experiments into executable protocols on specific laboratory automation.

Historically, laboratory automation has been driven by the desire to run large numbers of related laboratory experiments, especially in the pharmaceutical and clinical analysis industries. It is now a thriving multibillion dollar industry (King et al., 2023). The first use of AI to control laboratory equipment was probably that of Zytkow et al. (1990) (see above). The

technology of laboratory automation is steadily advancing, and robots can now carry out most (but not all) of the tasks that humans can do in the laboratory. Such laboratory automation is increasing the productivity of science as robots can work cheaper, faster, more accurately, and for longer (24/7) than humans, they can also be more easily increased/reduced in number. Laboratory automation still has many limitations. Robots typically today operate in protective boxes and are hard to program by bench scientists; logistics tasks are generally performed by lab technicians and scientists, with humans tending the robots for consumables; laboratory automation is expensive in capital to build and maintain—requiring specialised staff. Research in laboratory automation has been largely divorced from AI robotics research—which has mainly focused on the problem of mobile robots. Almost all laboratory robots are fixed in place, although there is growing interest in mobile robot assistants (Burger et al., 2020).

Hypothesis formation needs to be supported by a variety of AI and ML methods, from knowledge representation to active learning and reinforcement learning. The creation of a whole new theory, with theoretical terms and new measurement devices, is at least one level of complexity harder and has not been addressed yet at all.

5 Autonomy

One key aspect of AI Scientists is their degree of autonomy. One approach to measuring autonomy is to use the classification of degrees of autonomy in self-driving cars (King et al., 2023). The approach taken here is similar, Table 5 describes five levels of autonomy.

Beyond levels of autonomy are levels of skill. All human drivers are autonomous, but very few are skillful enough drivers to win a Formula 1 race. Among human scientists there are also levels of skill, with few human scientists being skillful enough to win a Nobel

Table 5 Six levels of autonomy in scientific discovery analogously to autonomy levels in autonomous driving

Level	Summary	Narrative	Example
0	No automation	Traditional human science before the advent of computers	–
1	Machine assistance	The use of computers to automate an aspect of science, e.g. analysing data	Most current applications of ML
2	Partial Automation	An important aspect of the discovery cycle is fully automated	AlphaFold 2, Real-time weather forecasting
3	Conditional Automation	Closed-loop automation. The full cycle of discovery is automated in a restricted domain	See Table 1
4	High Automation	Closed-loop automation. Multiple scientific domains. Limited ability to set its own goals	No existing system
5	Full Automation	All aspects of science are automated and no human intervention is required	No existing system

prize. AI scientists are improving in autonomy and skill. Extrapolating this trend, it is likely that advances in technology and our understanding of science will drive the development of ever-smarter AI Scientists. The Physics Nobel Frank Wilczek said that "in 100 years' time the best physicist will be a machine" (Wilczek & Devine, 2006). In February 2020 a workshop was held in London to kick-off the Nobel Turing Grand Challenge to develop: AI systems capable of making Nobel-quality scientific discoveries highly autonomously at a level comparable, and possibly superior, to the best human scientists by 2050 (Kitano, 2021). Achieving the Nobel Turing Grand Challenge would clearly transform the world.

6 Causal Discovery for Scientific Discovery

Causal discovery algorithms aim to infer causal relationships from data, providing a computational approach to what scientists do in forming and testing theories. Over the past two decades, there have been significant theoretical advances in causal discovery, ranging from constraint-based and score-based methods to functional model approaches and, more recently, deep learning techniques. These advances not only improve technical performance but also illuminate how algorithmic discovery parallels the scientific method of theory formation, model evaluation, and iterative refinement. In this section, we survey these developments, highlighting at least four major paradigms: constraint-based, score-based, functional causal models, and deep learning approaches, borrowing from the taxonomy established by Zheng et al. (2024). We also discuss how they converge with or diverge from frameworks of scientific discovery discussed elsewhere in this paper.

6.1 Constraint-Based Methods

Early work by Spirtes et al. (2000) and Pearl (2009) established the foundations of constraint-based causal discovery, leveraging graphical models and conditional independence. In a causal Bayesian network, the structure (a directed acyclic graph, DAG) entails specific conditional independence relations. Constraint-based algorithms like the PC algorithm (Spirtes et al., 2000) exploit this by systematically testing conditional independencies in data: if a hypothesized causal graph predicts that two variables should be independent given some condition and the test contradicts this, the hypothesis is falsified, akin to rejecting a scientific theory when its prediction fails. Under the assumption of causal Markov and faithfulness (i.e., all and only the independencies implied by the true causal graph are present in the data-generating distribution), the PC algorithm can recover the graph's Markov equivalence class (i.e. the set of DAGs indistinguishable by independence tests). Colombo and Maathuis (2014) introduced an order-independent version of PC to make results invariant to the (arbitrary) variable ordering. These constraint-based approaches map well to the scientific process of theory testing. Each conditional independence test is akin to an experiment probing a model's implication. Nonetheless the reliance on the faithfulness assumption has been controversial. This assumption has been analyzed in depth by Zhang and Spirtes (2016), who identified its role in making causal structure identifiable, algorithms tractable, and sample requirements reasonable. Violations of faithfulness (or of causal sufficiency, when unmeasured confounders exist) can mislead constraint-based methods, underscoring that real scientific data may not always cooperate with these idealized assumptions. Earlier

on, philosophers like Freedman and Humphreys (1999) critiqued causal discovery efforts on such grounds, arguing that inferring causation from mere observation was "premature at best" and that algorithmic outputs might not reliably correspond to true causal mechanisms in realistic samples.

6.2 Scoring Methods

An alternative to using local independence tests is to formulate causal discovery as a search for the best-fitting causal model according to a scoring criterion, typically balancing goodness-of-fit with model complexity. This approach aligns with scientific model selection by judging theories on how well they explain data. Pioneering work by Chickering (2002) introduced the Greedy Equivalence Search (GES) algorithm, which incrementally adds or removes edges to maximize a score (e.g., Bayesian information criterion). Similar measures have been employed in recent scientific discovery testbeds (Gandhi et al., 2025). Subsequent research provided exact or improved search strategies: Xiang and Kim (2013) applied A^* search to find globally optimal networks with an ℓ_1-regularized score, and Scanagatta et al. (2015) developed techniques to learn Bayesian networks with thousands of variables efficiently, illustrating the scalability of score-based discovery. A key insight of score-based methods is that they too ultimately rely on the data encoding of causal constraints (through the score), but rather than individually testing constraints, they evaluate entire model candidates. This is analogous to scientists considering multiple competing theories and selecting the one with the best empirical support and theoretical parsimony. The connection to theory evaluation is direct: a numerical score stands in for "evidence for the theory".

6.3 Functional-Causal Methods

A major theoretical breakthrough in causal discovery was the realization that by making structural assumptions on the functional form of causal mechanisms, one can achieve identifiability of the full causal DAG, not just its equivalence class. In other words, additional assumptions can break the symmetry and allow algorithms to distinguish cause from effect even in two-variable cases where vanilla methods cannot. An early example is the Linear Non-Gaussian Acyclic Model (LiNGAM) proposed by Shimizu et al. (2006). By assuming the data are generated by a linear structural equation model with non-Gaussian (independent) noise, LiNGAM enabled the recovery of the exact causal ordering and directed graph via independent component analysis techniques. This was a marked theoretical advance: Shimizu et al. (2006) showed that under non-Gaussian distributions one can rigorously identify who causes whom in a way that is impossible under purely Gaussian assumptions (where only undirected associations can be found). The idea that independence of noise and input carries identifying information was extended to nonlinear relationships. Hoyer et al. (2009) and later Peters et al. (2014) developed methods for nonlinear additive noise models (ANMs), assuming each child variable is a smooth function of its parents plus an independent noise term. Remarkably, if the data truly follow an additive noise structure, the directed acyclic graph is identifiable from observational data alone. This means that, in principle, one can discover the exact causal model (not just an equivalence class) because any wrong direction of an arrow would yield a different distribution than observed. Other functional approach advances include the Causal Additive Model (CAM) of Bühlmann et al. (2014),

which integrated high-dimensional variable selection with causal ordering under additive models. These works illustrate the interplay between strong modeling assumptions and causal discovery: by inserting additional theoretical assumptions (akin to background scientific knowledge or laws), the algorithms may tackle more general problems and settings.

Another notable development merging domain knowledge with data-driven discovery is the use of invariance across environments. Peters et al. (2016) introduced *Invariant Causal Prediction* (ICP), which leverages data from multiple experimental or environmental contexts. The idea is that a correct causal model for some target variable will have a certain subset of predictors whose relationship remains invariant across all environments (e.g., across different experimental conditions or subsets of data), whereas spurious associations will vary. This approach highlighted a principle akin to robustness in scientific theories: a law-like causal relation should hold universally (invariantly), not just in one dataset. Such work ties algorithmic causal discovery back to philosophical notions of natural laws and stability: it operationalizes the idea that if X directly causes Y, then interventions or changes in other factors (that do not directly affect Y) should not destabilize the X-Y relationship. In contrast, a non-causal correlation would likely break under new conditions.

6.4 Deep Learning and Causal Discovery

More recently, researchers have begun to employ continuous optimization and deep learning to tackle the combinatorial complexity of causal discovery. Traditional constraint-based and score-based searches have super-exponential worst-case complexity, motivating new strategies. A landmark contribution was the NOTEARS algorithm by Zheng et al. (2018), which introduced a smooth characterization of acyclicity. By embedding the DAG constraint into a differentiable function and optimizing a least-squares reconstruction loss with gradient-based methods, Zheng et al. (2018) showed it is possible to perform causal structure learning via standard machine learning frameworks (e.g., PyTorch), treating it like training a neural network but with an added continuous constraint ensuring no cycles. This approach was significant theoretically because it reframed structure search as a single-shot optimization problem, proving that under certain assumptions one can recover the exact DAG by solving a penalized regression. Building on this, Yu et al. (2019) proposed DAG-GNN, which uses a variational autoencoder (VAE) with a graph neural network to model complex nonlinear relationships while enforcing the DAG constraint, thus coupling deep learning's function approximation power with causal graph learning. Lachapelle et al. (2020) introduced a gradient-based method (GraN-DAG) that parameterizes neural networks in a way that inherently respects the DAG constraint during training, improving on the flexibility and accuracy of NOTEARS for nonlinear data. Meanwhile, Zheng et al. (2020) extended the differentiable paradigm to nonparametric models, relaxing assumptions of linearity by using kernels or generalized score functions, and demonstrated recovery of causal graphs in cases with no simple parametric relationship. In parallel, other researchers have explored reinforcement learning (RL) and combinatorial optimization techniques guided by deep networks. For instance, Zhu et al. (2020) treated the search for an optimal causal graph as a sequential decision problem, training an RL agent to add or remove edges in a graph in a way that maximizes a reward based on BIC score. This approach, and related ones using attention mechanisms to traverse search spaces, show improved performance on certain complex benchmarks, effectively learning search heuristics that would be hard to hand-craft.

There is considerable convergence between formal causal discovery techniques and scientific discovery. Both emphasize testable implications (an algorithm's use of data to accept/reject a model is akin to an experiment testing a theory's prediction), model selection (both scientists and algorithms prefer simpler, well-fitting models, Occam's razor), and iterative refinement (algorithms like GES or notears refine models stepwise, much as scientists refine theories). In this sense, causal discovery research has formalized aspects of theory formation: e.g., an equivalence class of graphs represents underdetermination of theory by evidence, and additional assumptions (like non-Gaussianity or invariance) play the role of new theoretical insights. We conclude there might be significant research to be undertaken at the intersection of causal discovery and automated scientific discovery, as the fields appear to have some complementarity: In practice, most autonomous discovery platforms either stop at finding predictive correlations/equations without guaranteeing they are truly causal, or they require substantial human input to interpret and test any learned models in a causal sense. Likewise, most causal discovery studies stop at producing a graph or simple functional relationships and do not continue on to automatically propose new laws or theories in a broader scientific context.

7 Evaluation and Testbeds

The evaluation of an autonomous discovery system is intrinsically tied to the levels of autonomy displayed by the methodology at hand and which steps of the scientific process are to be automatized and the level of autonomy being evaluated (Fig. 5 and Table 5). Equation discovery methods may help in automating the analysis of experiments by providing human-readable knowledge, while systems with physical actuators may be evaluated in their ability to execute experimental protocols. Thus, evaluation methodologies and benchmarks in the area have different characteristics in terms of supervision, data modalities, scope and open-endedness. We define these properties in the following, and give a table of existing methods for evaluation in Table 6.

Supervision. Supervision refers to the nature of the ground truth or reward signals provided to the autonomous discovery system during training and evaluation. Depending on the degree of autonomy assessed, supervision may range from explicit labels or predefined objectives to feedback signals [rewards in the reinforcement learning sense (Sutton & Barto, 2018)]. The type and quantity of supervision significantly affect the evaluation outcome, as they directly influence the system's capability to navigate scientific exploration autonomously.

Data Modalities. Data modalities encompass the types and formats of data available for evaluation, such as pixel-based images, textual descriptions, numerical tables, or structured representations of experimental observations. The choice of modality greatly impacts the complexity and applicability of autonomous systems, as certain data forms inherently require more sophisticated methods for interpretation, abstraction, and knowledge extraction (see Sect. 3). Evaluating systems across diverse data modalities helps in understanding their flexibility, generalizability, and robustness in real-world scientific scenarios.

Scope. Scope defines which specific phases of the scientific discovery process the evaluation benchmark addresses. This includes one or more of the six distinct steps: scientific

Table 6 Benchmark Categorization by Evaluation Properties. In the **Scope** column, we take D = experimental Design, E = Experimental Execution, H = Hypothesis formation, A = Analysis of results, Q = research Question formation

Benchmark	Supervision	Data Modalities	Scope	Open-endedness
Nguyen Uy et al. (2011)	Equation	Tabular	A	No
Feynman Udrescu and Tegmark (2020)	Equation	Tabular	A	No
ScienceWorld Wang et al. (2022)	Reward	Text	D, E, A	No
DiscoveryWorld Jansen (2024)	Reward	Text, images	All	Some
ChemGymRL Beeler et al. (2024)	Reward	Tabular	E, A	No
DiscoveryBench Majumder et al. (2024)	Rewards, LLM judge	Tabular, text	H, A	No
BoxingGym Gandhi et al. (2025)	Rewards	Tabular, textual	H, D, E	No
ScienceGym Cerrato et al. (2024)	Rewards	Tabular, Images	H, D, E, A	No
Matbench Dunn et al. (2020)	Supervised	Tabular	H, A	No
Open Catalyst Chanussot (2021)	Labels	Tabular, Graph	H, A	No
SCP-116K Lu et al. (2025)	Supervised	Textual	Q, H, A	No
The Well Ohana et al. (2024)	Equation	Tabular	Q, H, E, A	Yes

question formulation, hypothesis generation, experimental design, execution of experiments, data analysis and communication.

Open-endedness. Open-endedness characterizes whether the benchmark or evaluation method includes previously unexplained data, phenomena lacking known mathematical descriptions, or allows the formulation of novel scientific questions. An open-ended benchmark gives the opportunity to showcase the capabilities of a discovery system not only in replicating existing knowledge but in discovering genuinely novel information.

We now move to introducing benchmark and testbeds while discussing their potential in the autonomous discovery setting. We will not offer here an exhaustive survey of symbolic regression benchmarks.

7.1 Available Benchmarks

Nguyen dataset by Uy et al. (2011) is a widely-used collection of symbolic regression problems introduced specifically to evaluate genetic programming (GP) methods. It consists of synthetic mathematical equations designed with varying complexity and structure, aiming to assess the ability of GP algorithms to accurately recover symbolic expressions from numerical data. Each task provides numerical input–output pairs generated from known symbolic formulas. This benchmark primarily evaluates one-shot analysis of already collected experimental data and has been extensively employed in the testing of symbolic regression methods.

Feynman by Udrescu and Tegmark (2020) provides a comprehensive symbolic regression benchmark inspired by fundamental physics equations from the *Feynman Lectures on Physics*. This dataset includes 120 symbolic regression tasks covering a diverse range of physics phenomena, from classical mechanics to electromagnetism.

Matbench by Dunn et al. (2020) is a supervised machine-learning benchmark containing 13 prediction tasks related to materials science. The dataset consists of structured data representing chemical formulas and crystalline structures, with tasks that involve predicting material properties such as band gap or elastic moduli: thus, it focuses on applications in materials science. While each task is narrowly defined with a fixed prediction goal, collectively, a transfer learning approach may be employed to also evaluate the generalization capabilities of discovery algorithms across materials science domains.

SCP-116K by Lu et al. (2025) is a large-scale textual dataset comprising problem-solution pairs extracted from higher education science textbooks and other academic sources, totaling 116,000 entries. It is designed primarily for supervised training and evaluation of models on scientific reasoning, question answering, and hypothesis generation from textual data. While each problem-solution pair is relatively constrained in scope, the dataset's scale and diversity across scientific disciplines provides opportunities for broader generalization and transfer learning evaluation.

The Well by Ohana et al. (2024) is a comprehensive collection of physics simulation datasets, explicitly constructed for machine learning model training and benchmarking in physics-informed learning. It contains diverse simulation data spanning fluid dynamics, astrophysics, plasma physics, and more. These simulations allow evaluation of models' abilities in hypothesis generation, scientific analysis, and predictive modeling in physics. The current baseline analysis proposed by Ohana et al. (2024) rather focuses on predictive capabilities of neural operator algorithms, which appears to leave opportunities for researchers in the space of automated discovery.

ScienceWorld by Wang et al. (2022) is a publicly available reinforcement learning environment designed to evaluate an AI agent's capacity for grounded scientific reasoning in a simulated laboratory context. The benchmark contains 30 interactive text-based tasks, such as converting substances between states of matter. Evaluation relies on binary task completion within a budget of simulator steps. Compared to other testbeds, this environment thus evaluates experimental execution capabilities of an agent, albeit in an idealized, text-based format.

DiscoveryWorld by Jansen (2024) is an open-source, highly interactive environment designed to benchmark complete cycles of scientific discovery, including hypothesis generation, experimental design, execution, and analysis. The general setting is akin to a 2D

role-playing game to be played on a grid. It provides agents with quests, subquests and various tasks to be completed to make progress.

ChemGymRL by Beeler et al. (2024) provides a suite of customizable, publicly accessible reinforcement learning environments simulating chemistry laboratory experiments. Each virtual "bench" simulates distinct chemical procedures such as synthesis or titration. Agents receive structured numeric data representing chemical states and perform sequential lab actions. In terms of the scientific process, this library appears to emphasize experimental design and execution with reward signals. Extensions to e.g. new chemical reactions are possible and encouraged by the authors.

DiscoveryBench by Majumder et al. (2024) is a publicly accessible benchmark focusing on data-driven scientific discovery tasks using multimodal data (tabular data and textual descriptions). It comprises over a thousand real-world and synthetic tasks spanning various scientific domains. Evaluation of agent-generated hypotheses is performed using LLMs, which may allow for some open-endedness in the tasks considered. DiscoveryBench primarily targets hypothesis generation and data analysis, but experimental execution and planning is also evaluated: agents are supposed to plan out the code necessary to analyze the data and come to conclusions, which may transfer to e.g. lab actuators controlled by software.

BoxingGym by Gandhi et al. (2025) provides publicly available, interactive simulation environments for benchmarking autonomous experimental design and scientific model discovery. The benchmark covers multiple scientific domains through generative probabilistic models. Evaluation metrics include expected information gain for experimental quality and predictive power of agent-generated scientific models. The environment is numeric and textual in data modalities.

Science-Gym by Cerrato et al. (2024) is a publicly released Gym-compatible benchmark designed to evaluate autonomous equation discovery in simulated physical and epidemiological environments. Agents interactively select experimental parameters to generate data, subsequently performing symbolic regression to derive underlying scientific equations. Evaluation assesses the symbolic accuracy of discovered equations, providing a structured setting emphasizing experimental execution and analytical reasoning.

Open Catalyst 2020 (OC20) by Chanussot (2021) provides a large-scale benchmark for catalysis research, encompassing over a million atomic structure relaxations generated via density functional theory (DFT) calculations. It offers structured atomic 3D data for supervised machine learning tasks aimed at predicting energies and molecular interactions relevant to catalytic processes. OC20 primarily evaluates data-driven analysis and indirectly supports hypothesis-driven experimental design, particularly aiding in computational screening of catalytic materials. While individually each task has a fixed objective, its expansive dataset encourages robust and generalizable modeling approaches.

8 Conclusion

This paper is an attempt at giving a survey of research on automated scientific discovery, from discovering equations to autonomous discovery systems or agents. In doing so, it takes a broad perspective on the topic, which is necessary to understand the individual efforts in context. The article covers the beginnings of the fields to very recent approaches, under-

standing that we still have a long way of putting everything together to create human-level autonomous scientists. Human-level autonomous scientists should, ultimately, be able to produce whole new theories, along with theoretical terms and measurement devices, which can be communicated to humans and interpreted in the light of other, existing theories. At this point, autonomous discovery systems are focused primarily on "closing the loop" and lab automation, and not so much on generating human-interpretable knowledge, like (differential) equations. Vice versa, computational approaches to scientific discovery, e.g., for equation discovery and symbolic regression, do not have the "embodiment" in autonomous systems in their focus yet. Ultimately, these currently disparate efforts have to grow together. Finally, it should be noted that artificial intelligence has a role also in so far unexplored areas, like the design of experiments, where much of human ingenuity is currently still needed.

Author Contributions S.K. conceptualized the survey and the overall structure. M.C. contributed to most sections, esp. about testbeds and experimentation. J.B. contributed expertise around equation discovery and symbolic regression, and refined the text in other sections. S.D. contributed to the historical perspective on the field and oversaw the development. R.K. contributed about autonomy and current developments and oversaw the development.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no Conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

Al-Helali, B., Chen, Q., Xue, B., & Zhang, M. (2020). GP-based feature selection and weighted KNN-based instance selection for symbolic regression with incomplete data. In *2020 IEEE symposium series on computational intelligence, SSCI 2020, Canberra, Australia, December 1–4*, 2020, pp. 905–912. IEEE Press, Piscataway, NJ. https://doi.org/10.1109/SSCI47803.2020.9308382

Al-Helali, B., Chen, Q., Xue, B., & Zhang, M. (2024). Genetic programming for feature selection based on feature removal impact in high-dimensional symbolic regression. *IEEE Transactions on Emerging Topics in Computational Intelligence, 8*(3), 2269–2282. https://doi.org/10.1109/TETCI.2024.3369407

Ali, M.S., Kshirsagar, M., Naredo, E., & Ryan, C. (2022). Automated grammar-based feature selection in symbolic regression. In *Proceedings of the 2022 Genetic and Evolutionary Computation Conference (GECCO)*, pp. 902–910. https://doi.org/10.1145/3512290.3528852

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., & Church, G. M. (2019). Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods, 16*(12), 1315–1322. https://doi.org/10.1038/s41592-019-0598-1

Beeler, C., Subramanian, S. G., Sprague, K., Baula, M., Chatti, N., Dawit, A., Li, X., Paquin, N., Shahen, M., Yang, Z., Bellinger, C., Crowley, M., & Tamblyn, I. (2024). ChemGymRL: A customizable interactive framework for reinforcement learning for digital chemistry. *Digital Discovery, 3*, 742–758. https://doi.org/10.1039/D3DD00183K

Bendinelli, T., Biggio, L., & Kamienny, P. (2023). Controllable neural symbolic regression. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (eds.) International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA. Proceedings of Machine Learning Research, 202, 2063–2077.

Berghaus, D., Cvejoski, K., Seifner, P., Ojeda, C., & Sanchez, R. (2024). Foundation inference models for markov jump processes. In: Advances in Neural Information Processing Systems (NeurIPS). https://papers.nips.cc/paper_files/paper/2024/hash/e9df36b21ff4ee211a8b71ee8b7e9f57-Abstract-Conference.html

Biggio, L., Bendinelli, T., Neitz, A., Lucchi, A., & Parascandolo, G. (2021). Neural symbolic regression that scales. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18–24 July 2021, Virtual Event. Proceedings of Machine Learning Research, vol. 139, pp. 936–945. Virtual event. http://proceedings.mlr.press/v139/biggio21a.html

Bomarito, G.F., Leser, P.E., Strauss, N.C.M., Garbrecht, K.M., & Hochhalter, J.D. (2022). Bayesian model selection for reducing bloat and overfitting in genetic programming for symbolic regression. In *Proceedings of the 2022 genetic and evolutionary computation conference companion (GECCO)*, pp. 526–529.https://doi.org/10.1145/3520304.3528899.

Brence, J., Dzeroski, S., & Todorovski, L. (2023). Dimensionally-consistent equation discovery through probabilistic attribute grammars. *Information Sciences, 632*, 742–756. https://doi.org/10.1016/j.ins.2023.03.073

Brence, J., Todorovski, L., & Džeroski, S. (2021). Probabilistic grammars for equation discovery. *Knowledge Based Systems, 224*, Article 107077.

Bridewell, W., Langley, P., Todorovski, L., & Džeroski, S. (2008). Inductive process modeling. *Machine Learning, 71*, 1–32.

Brugger, J., Cerrato, M., Richter, D., Derstroff, C., Maninger, D., Mezini, M., & Kramer, S. (2025). Neural-guided equation discovery. https://arxiv.org/abs/2503.16953

Brugger, J., Pfanschilling, V., Richter, D., Mezini, M., & Kramer, S. (2025). Prompting neural-guided equation discovery based on residuals. In *Proceedings of the 28th international conference on discovery science 2025*. Accepted for publication.

Brunton, S. L., Proctor, J. L., & Kutz, J. N. (2016). Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *PNAS, 113*, 3932–3937.

Bühlmann, P., Peters, J., & Ernest, J. (2014). CAM: Causal additive models, high-dimensional order search and penalized regression. *Annals of Statistics, 42*(6), 2526–2556.

Burger, B., Maffettone, P. M., Gusev, V. V., Aitchison, C. M., Bai, Y., Wang, X., Li, X., Alston, B. M., Li, B., Clowes, R., Rankin, N., Harris, B., Sprick, R. S., & Cooper, A. I. (2020). A mobile robotic chemist. *Nature, 583*(7815), 237–241. https://doi.org/10.1038/s41586-020-2442-2

Bychkov, A., Issan, I., Pogudin, G., & Krämer, B. (2024). Exact and optimal quadratization of nonlinear finite-dimensional non-autonomous dynamical systems. *SIAM Journal on Applied Dynamical Systems, 23*(1), 982–1016.

Čerepnalkoski, D., Taškova, K., Todorovski, L., Atanasova, N., & Džeroski, S. (2012). The influence of parameter fitting methods on model structure selection in automated modeling of aquatic ecosystems. *Ecological Modelling, 45*, 136–165.

Cerrato, M., Schmitt, N., Baur, L., Finkelstein, E., Jukic, S., Münzel, L., Paul, F.P., Pfannes, P., Rohr, B., Schellenberg, J., Wolf, P., & Kramer, S. (2024). Science-Gym: A simple testbed for ai-driven scientific discovery. In: Proceedings of the 26th International Conference on Discovery Science (DS). Lecture Notes in Computer Science, vol. 15243, pp. 229–243. Springer, Pisa, Italy. https://doi.org/10.1007/978-3-031-78977-9_15

Chanussot, Lea. (2021). The open catalyst 2020 (oc20) dataset and community challenges. *ACS Catalysis, 11*(10), 6059–6072.

Chaushevska, M., Todorovski, L., Brence, J., & Džeroski, S. (2022). Learning the probabilities in probabilistic context-free grammars for arithmetical expressions from equation corpora. In *Proceedings of the Slovenian conference on artificial intelligence*.

Chen, B., Huang, K., Raghupathi, S., Chandratreya, I., Du, Q., & Lipson, H. (2022). Automated discovery of fundamental variables hidden in experimental data. *Nature Computational Science, 2*, 433–442.

Chen, Q., Zhang, M., & Xue, B. (2017). Feature selection to improve generalization of genetic programming for high-dimensional symbolic regression. *IEEE Transactions on Evolutionary Computation, 21*(5), 792–806. https://doi.org/10.1109/TEVC.2017.2683489

Chickering, D. M. (2002). Optimal structure identification with greedy search. *Journal of Machine Learning Research, 3*, 507–554.
Cohn, D. A., Ghahramani, Z., & Jordan, M. I. (1996). Active learning with statistical models. *Journal of Artificial Intelligence Research, 4*, 129–145.
Colombo, D., & Maathuis, M. H. (2014). Order-independent constraint-based causal structure learning. *Journal of Machine Learning Research, 15*, 3921–3962.
Cornelio, C., Dash, S., Austel, V., Josephson, T.R., Goncalves, J., Clarkson, K.L., Megiddo, N., Khadir, B.E., & Horesh, L. (2023). Combining data and theory for derivable scientific discovery with AI-Descartes. Nature Communications **14**. https://doi.org/10.1038/s41467-023-37236-y
Cranmer, M. D. (2023). Interpretable machine learning for science with PySR and SymbolicRegression.jl. CoRR **abs/2305.01582**. arXiv:2305.01582
Cranmer, M.D., Sanchez-Gonzalez, A., Battaglia, P.W., Xu, R., Cranmer, K., Spergel, D.N., & Ho, S. (2020). Discovering symbolic models from deep learning with inductive biases. In *Advances in Neural Information Processing Systems* 33. https://dl.acm.org/doi/10.5555/3495724.3497186
Crochepierre, L., Boudjeloud-Assala, L., & Barbesant, V. (2022). Interactive reinforcement learning for symbolic regression from multi-format human-preference feedbacks. In: Raedt, L. D. (Ed.) *International joint conference on artificial intelligence*, pp. 5900–5903. https://doi.org/10.24963/IJCAI.2022/849
Du, Y., Guo, X., Shehu, A., & Zhao, L. (2022). Interpretable molecular graph generation via monotonic constraints, pp. 73–81. https://doi.org/10.1137/1.9781611977172.9
Dunn, A., Wang, Q., Ganose, A., Dopp, D., & Jain, A. (2020). Benchmarking materials property prediction methods: The matbench test set and automatminer reference algorithm. *npj Computational Materials, 6*, 138.
Džeroski, S., & Petrovski, I. (1994). Discovering dynamics with genetic programming. In *Proceedings of the seventh European conference on machine learning*, pp. 347–350. Springer, Berlin.
Džeroski, S., & Todorovski, L. (1993). Discovering dynamics. In *Proceedings of the tenth international conference on machine learning*, pp. 97–103. Morgan Kaufmann, Amherst, MA, USA.
Eisenstein, M. (2025). Self-driving laboratories, advanced immunotherapies and five more technologies to watch in 2025. *Nature, 637*, 1008–1011. https://doi.org/10.1038/d41586-025-00075-6
Ellis, K., Morales, L., Sable-Meyer, M., Solar-Lezama, A., & Tenenbaum, J.B. (2018). Library learning for neurally-guided Bayesian program induction. In *Advances in Neural Information Processing Systems*, 31 (NeurIPS 2018), 31.
Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2022). ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 44*(10), 7112–7127. https://doi.org/10.1109/TPAMI.2021.3095381
Fox, C., Tran, N. D., Nacion, F. N., Sharlin, S., & Josephson, T. R. (2024). Incorporating background knowledge in symbolic regression using a computer algebra system. *Machine Learning: Science and Technology, 5*, Article 025057. https://doi.org/10.1088/2632-2153/ad4a1e
Franca, F. O., & Kronberger, G. (2023). Reducing overparameterization of symbolic regression models with equality saturation. In *Proceedings of the 2023 genetic and evolutionary computation conference (GECCO)*, pp. 1064–1072. https://doi.org/10.1145/3583131.3590346
Freedman, D., & Humphreys, P. (1999). Are there algorithms that discover causal structure? *Synthese, 121*(1), 29–54.
Gandhi, K., Li, M.Y., Goodyear, L., Li, L., Bhaskar, A., Zaman, M., & Goodman, N.D. (2025). *BoxingGym: Benchmarking progress in automated experimental design and model discovery*. Project page with environments: https://github.com/kanishkg/boxing-gym
Ganzert, S., Guttmann, J., Steinmann, D., & Kramer, S. (2010). Equation discovery for model identification in respiratory mechanics of the mechanically ventilated human lung. In *Proceedings of the 13th international conference on discovery science (DS 2010)*, pp. 296–310. Springer, Berlin.
Gao, M. L., Williams, J. P., & Kutz, J. N. (2025). Sparse identification of nonlinear dynamics and Koopman operators with Shallow Recurrent Decoder Networks. https://arxiv.org/abs/2501.13329.
Gao, S., Fang, A., Huang, Y., Giunchiglia, V., Noori, A., Schwarz, J. R., Ektefaie, Y., Kondic, J., & Zitnik, M. (2024). Empowering biomedical discovery with AI agents. *Cell, 187*(22), 6125–6151.
Garcon, A., Vexler, J., Budker, D., & Kramer, S. (2022). Deep neural networks to recover unknown physical parameters from oscillating time series. *PLoS ONE, 17*(5), 0268439.
Grayeli, A., Sehgal, A., Costilla-Reyes, O., Cranmer, M.D., & Chaudhuri, S. (2024). Symbolic regression with a learned concept library. In *Advances in neural information processing systems (NeurIPS)*.
Guimerà, R., Reichardt, I., Aguilar-Mogas, A., Massucci, F. A., Miranda, M., Pallarès, J., & Sales-Pardo, M. (2020). A Bayesian machine scientist to aid in the solution of challenging scientific problems. *Science Advances, 6*, 6971.

Haddad, R., Litsa, E. E., Liu, Z., Burkhardt, D., & Bhisetti, G. (2025). Targeted molecular generation with latent reinforcement learning. *Scientific Reports, 15*, 15202. https://doi.org/10.1038/s41598-025-99785-0

Haider, C., Franca, F.O., Kronberger, G., & Burlacu, B. (2022). Comparing optimistic and pessimistic constraint evaluation in shape-constrained symbolic regression. In *Proceedings of the 2022 genetic and evolutionary computation conference (GECCO)*, pp. 938–945. https://doi.org/10.1145/3512290.3528714

Haider, C., Franca, F. O., Burlacu, B., & Kronberger, G. (2023). Shape-constrained multi-objective genetic programming for symbolic regression. *Applied Soft Computing, 132*, Article 109855. https://doi.org/10.1016/j.asoc.2022.109855

Hoyer, P. O., Janzing, D., Mooij, J. M., Peters, J., & Schölkopf, B. (2009). Nonlinear causal discovery with additive noise models. *Advances in Neural Information Processing Systems (NeurIPS), 21*, 689–696.

Huang, K.-M., & Zytkow, J. M. (1997). Discovering empirical equations from robot-collected data. In *Proceedings of the 10th international symposium on foundations of intelligent systems (ISMIS 1997)*, pp. 287–297. Springer, Charlotte, North Carolina, USA.

Huang, Z., Zhong, J., Feng, L., Mei, Y., & Cai, W. (2020). A fast parallel genetic programming framework with adaptively weighted primitives for symbolic regression. *Soft Computing, 24*(10), 7523–7539. https://doi.org/10.1007/s00500-019-04379-4

Jansen, P. E. A. (2024). DiscoveryWorld: A virtual environment for developing and evaluating automated scientific discovery agents. In *Advances in neural information processing systems*, vol. 37.

Jiang, N., & Xue, Y. (2023). Symbolic regression via control variable genetic programming. In *Proceedings of the European conference on machine learning and principles and practice of knowledge discovery in databases (ECML PKDD 2023)*, pp. 178–195. Springer, Berlin. https://doi.org/10.1007/978-3-031-43421-1_11

Jiang, N., & Xue, Y. (2024). Racing control variable genetic programming for symbolic regression. In Wooldridge, M.J., Dy, J.G., Natarajan, S. (Eds.) *Thirty-eighth AAAI conference on artificial intelligence, AAAI 2024, thirty-sixth conference on innovative applications of artificial intelligence, IAAI 2024, fourteenth symposium on educational advances in artificial intelligence, EAAI 2014, February 20–27*, 2024, Vancouver, Canada, pp. 12901–12909. AAAI Press, Washington, DC. https://doi.org/10.1609/AAAI.V38I11.29187

Jin, P., Huang, D., Zhang, R., Hu, X., Nan, Z., Du, Z., Guo, Q., & Chen, Y. (2023). Online symbolic regression with informative query. In: Williams, B., Chen, Y., Neville, J. (Eds.) *Thirty-seventh AAAI conference on artificial intelligence*, AAAI, pp. 5122–5130. AAAI Press, Washington, DC. https://doi.org/10.1609/AAAI.V37I4.25641

Kahlmeyer, P., Fischer, M., & Giesen, J. (2025). Dimension reduction for symbolic regression. In *AAAI*, pp. 17707–17714. https://doi.org/10.1609/aaai.v39i17.33947

Kamienny, P., Lample, G., Lamprier, S., & Virgolin, M. (2023). Deep generative symbolic regression with Monte-Carlo-Tree-Search. In: Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., Scarlett, J. (Eds.) *International conference on machine learning, ICML 2023*, 23–29 July 2023. Proceedings of Machine Learning Research, vol. 202, pp. 15655–15668.

Kamienny, P., d'Ascoli, S., Lample, G., & Charton, F. (2022). End-to-end symbolic regression with transformers. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in neural information processing systems 35: Annual conference on neural information processing systems 2022, NeurIPS 2022, November 28–December 9, 2022*. New Orleans: LA, USA.

Kim, J. T., Kim, S., & Petersen, B. K. (2020). An interactive visualization platform for deep symbolic regression. In *International joint conference on artificial intelligence*, IJCAI, pp. 5261–5263. https://doi.org/10.24963/IJCAI.2020/763

King, R., Peter, O., & Courtney, P. (2023). Robot scientists: From Adam to Eve to Genesis. In T. Science & O. Innovation (Eds.), *Artificial intelligence in science: Challenges, opportunities and the future of research* (pp. 127–138). Paris: OECD Publishing.

King, R. D., Rowland, J., Oliver, S. G., Young, M., Aubrey, W., Byrne, E., Liakata, M., Markham, M., Pir, P., Soldatova, L. N., Sparkes, A., Whelan, K. E., & Clare, A. (2009). The automation of science. *Science, 324*(5923), 85–89.

Kirchoff, K. E., Maxfield, T., Tropsha, A., & Gomez, S. M. (2024). SALSA: Semantically-aware latent space autoencoder. *Proceedings of the AAAI Conference on Artificial Intelligence, 38*(12), 13211–13219. https://doi.org/10.1609/aaai.v38i12.29221

Kitano, H. (2021). Nobel Turing challenge: Creating the engine for scientific discovery. npj Systems Biology and Applications **7**(29).

Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A. M., & Anandkumar, A. (2023). Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research, 24*, 1–97.

Koza, J. R. (1994). Genetic programming as a means for programming computers by natural selection. *Statistics and Computing, 4*, 87–112.

Kubalik, J., Alibekov, E., Zegklitz, J., & Babuska, R. (2016). Hybrid single node genetic programming for symbolic regression. In Transactions on computational collective intelligence XXIV, lecture notes in computer science, pp. 61–82. https://doi.org/10.1007/978-3-662-53525-7_4

Kubalík, J., Derner, E., & Babuska, R. (2021). Multi-objective symbolic regression for physics-aware dynamic modeling. *Expert Systems with Applications, 182*, Article 115210. https://doi.org/10.1016/j.eswa.2021.115210

Kubalík, J., Derner, E., & Babuska, R. (2023). Toward physically plausible data-driven models: A novel neural network approach to symbolic regression. *IEEE Access, 11*, 61481–61501. https://doi.org/10.1109/ACCESS.2023.3287397

Lachapelle, S., Brouillard, P., Deleu, T., & Lacoste-Julien, S. (2020). Gradient-based neural DAG learning. In *International conference on learning representations (ICLR)*.

Langley, P. (1977). BACON: A production system that discovers empirical laws. In *Proceedings of the 5th international joint conference on artificial intelligence (IJCAI 1977)*, p. 344.

Langley, P. (2024). Integrated systems for computational scientific discovery. In *Proceedings of the thirty-eighth AAAI conference on artificial intelligence (AAAI-24)*, pp. 22598–22606. AAAI Press, Washington, DC.

Langley, P. (2022). Agents of exploration and discovery. *AI Magazine, 42*(4), 72–82.

Langley, P. W., Simon, H. A., Bradshaw, G., & Zytkow, J. M. (1987). *Scientific discovery: Computational explorations of the creative process*. Cambridge, MA: MIT Press.

Levina, E., & Bickel, P.J. (2004). Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems* 17, pp. 777–784.

Li, Z., Ji, J., & Zhang, Y. (2021). From Kepler to Newton: Explainable AI for science. arXiv preprint. arXiv:2111.12210

Li, Z., Kovachki, N.B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A.M., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. In *Proceedings of the 9th international conference on learning representations (ICLR 2021)*.

Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Soljačić, M., Hou, T. Y., & Tegmark, M. (2024). KAN: Kolmogorov–Arnold networks. arXiv preprint arXiv:2404.19756.

Lu, D., Tan, X., Xu, R., Yao, T., Qu, C., Chu, W., Xu, Y., & Qi, Y. (2025). *SCP-116K: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain*. https://arxiv.org/abs/2501.15587

Lu, L., Jin, P., Pang, G., Zhang, Z., & Karniadakis, G. E. (2021). Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence, 3*(3), 218–229.

Lusch, B., Kutz, J. N., & Brunton, S. L. (2018). Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications, 9*(1), 4950.

Majumder, B.P., Surana, H., Agarwal, D., Dalvi Mishra, B., Meena, A., Prakhar, A., Vora, T., Khot, T., Sabharwal, A., & Clark, P. (2024). DiscoveryBench: Towards data-driven discovery with large language models. Dataset and code available on GitHub (https://github.com/allenai/discoverybench) and HuggingFace.

Makke, N., & Chawla, S. (2024). Interpretable scientific discovery with symbolic regression: A reviews. *Artificial Intelligence Review, 57*(2).

Merler, M., Haitsiukevich, K., Dainese, N., & Marttinen, P. (2024). In-Context Symbolic Regression: Leveraging Large Language Models for Function Discovery. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop), pp. 589–606. Association for Computational Linguistics, Bangkok, Thailand. https://doi.org/10.18653/v1/2024.acl-srw.49 . https://aclanthology.org/2024.acl-srw.49 Accessed 2025-03-17

Meyerson, E., Nelson, M. J., Bradley, H., Gaier, A., Moradi, A., Hoover, A. K., & Lehman, J. (2023). Language Model Crossover: Variation through Few-Shot Prompting. arXiv. Version Number: 3. https://doi.org/10.48550/ARXIV.2302.12170. https://arxiv.org/abs/2302.12170 Accessed 2025-03-27.

Mežnar, S., Džeroski, S., & Todorovski, L. (2023). Efficient generator of mathematical expressions for symbolic regression. *Machine Learning, 112*(11), 4563–4596.

Mundhenk, T. N., Landajuela, M., Glatt, R., Santiago, C. P., Faissol, D. M., & Petersen, B. K. (2021). Symbolic regression via neural-guided genetic programming population seeding. arXiv preprint arXiv:2111.00053.

Musslick, S., Bartlett, L.K., Chandramouli, S.H., Dubova, M., Gobet, F., Griffiths, T.L., Hullman, J., King, R.D., Kutz, J.N., Lucas, C.G., Mahesh, S., Pestilli, F., Sloman, S.J., & Holmes, W.R. (2025). Automating the practice of science: Opportunities, challenges, and implications. PNAS **122**(5).

Neshatian, K., & Varn, L. (2017). On the existence of feature bundles and their effect on symbolic regression algorithms. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pp. 2974–2981. https://doi.org/10.1109/SSCI.2017.8285419.

Ohana, R., McCabe, M., Meyer, L.T., Morel, R., Agocs, F.J., Beneitez, M., Berger, M., Burkhart, B., Dalziel, S.B., Fielding, D.B., Fortunato, D., Goldberg, J.A., Hirashima, K., Jiang, Y.-F., Kerswell, R., Maddu, S., Miller, J.M., Mukhopadhyay, P., Nixon, S.S., Shen, J., Watteaux, R., Blancard, B.R.-S., Rozet, F., Parker, L.H., Cranmer, M., & Ho, S. (2024). The Well: a large-scale collection of diverse physics simulations for machine learning. In: The Thirty-eight Conference on neural information processing systems datasets and benchmarks track. https://openreview.net/forum?id=00Sx577BT3

Pearl, J. (2009). *Causality: Models, reasoning, and inference* (2nd ed.). New York: Cambridge University Press.

Peters, J., Bühlmann, P., & Meinshausen, N. (2016). Causal inference by using invariant prediction: Identification and confidence intervals. *Journal of the Royal Statistical Society: Series B, 78*(5), 947–1012. https://doi.org/10.1111/rssb.12167

Petersen, B.K., Larma, M.L., Mundhenk, T.N., Santiago, C.P., Kim, S.K., & Kim, J.T. (2021). Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. In *Proceedings of the 9th international conference on learning representations (ICLR 2021)*.

Peters, J., Mooij, J. M., Janzing, D., & Schölkopf, B. (2014). Causal discovery with continuous additive noise models. *Journal of Machine Learning Research, 15*(1), 2009–2053. https://doi.org/10.5555/2627435.2670315

Prieschl, S., Girardi, D., & Kronberger, G. (2019). Using ontologies to express prior knowledge for genetic programming. In *3rd International cross-domain conference for machine learning and knowledge extraction*, pp. 362–376. https://doi.org/10.1007/978-3-030-29726-8_23

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2021). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences, 118*(15), 2016239118. https://doi.org/10.1073/pnas.2016239118

Rong, Y., Bian, Y., Xu, T., Xie, W., Wei, Y., Huang, W., & Huang, J. (2020). Self-supervised graph transformer on large-scale molecular data. In *Advances in neural information processing systems, vol. 33*. https://proceedings.neurips.cc/paper/2020/hash/94aef38441efa3380a3bed3faf1f9d5d-Abstract.html.

Roper, K., Abdel-Rehim, A., Hubbard, S., Carpenter, M., Rzhetsky, A., Soldatova, L. N., & King, R. D. (2022). Testing the reproducibility and robustness of the cancer biology literature by robot. *Journal of the Royal Society Interface, 19*(189), 20210821. https://doi.org/10.1098/rsif.2021.0821

Sahoo, S., Lampert, C., & Martius, G. (2018). Learning Equations for Extrapolation and Control. In *Proceedings of the 35th international conference on machine learning*, Stockholm, Sweden, pp. 4442–4450. ISSN: 2640-3498. https://proceedings.mlr.press/v80/sahoo18a.html. Accessed 2024-02-19

Sambo, A. S., Azad, R. M. A., Kovalchuk, Y., Indramohan, V. P., & Shah, H. (2021). Evolving simple and accurate symbolic regression models via asynchronous parallel computing. *Applied Soft Computing, 104*, Article 107198. https://doi.org/10.1016/j.asoc.2021.107198

Scanagatta, M., Campos, C. P., Corani, G., & Zaffalon, M. (2015). Learning bayesian networks with thousands of variables. *Advances in Neural Information Processing Systems (NeurIPS), 28*, 1855–1863.

Schmidt, M., & Lipson, H. (2009). Distilling free-form natural laws from experimental data. *Science, 324*(5923), 81–85. https://doi.org/10.1126/science.1165893

Seifner, P., Cvejoski, K., Körner, A., & Sanchez, R. (2025). Zero-shot imputation with foundation inference models for dynamical systems. In *Proceedings of the international conference on learning representations (ICLR)*. To appear. https://openreview.net/forum?id=NPSZ7V1CCY

Sharlin, S., & Josephson, T. R. (2024). In context learning and reasoning for symbolic regression with large language models. arXiv. Version Number: 2. https://doi.org/10.48550/ARXIV.2410.17448. https://arxiv.org/abs/2410.17448 Accessed 2025-03-17

Shimizu, S., Hoyer, P. O., Hyvärinen, A., & Kerminen, A. (2006). A linear non-Gaussian acyclic model for causal discovery. *Journal of Machine Learning Research, 7*, 2003–2030.

Shojaee, P., Meidani, K., Farimani, A. B., & Reddy, C. K. (2023). Transformer-based planning for symbolic regression https://doi.org/10.48550/ARXIV.2303.06833. Publisher: arXiv Version Number: 4. Accessed 2023-08-09

Shojaee, P., Meidani, K., Gupta, S., Farimani, A.B., & Reddy, C.K. (2024). LLM-SR: Scientific equation discovery via programming with large language models. arXiv. Version Number: 2. https://doi.org/10.48550/ARXIV.2404.18400. https://arxiv.org/abs/2404.18400 Accessed 2024-08-05

Sparkes, A., Aubrey, W., Byrne, E., Clare, A., Khan, M.N., Liakata, M., Markham, M., Rowland, J., Soldatova, L.N., Whelan, K.E., Young, M., & King, R.D. (2021). Towards robot scientists for autonomous scientific discovery. Automated Experimentation **2**(1).

Spirtes, P., Glymour, C., & Scheines, R. (2000). *Causation, prediction, and search* (2nd ed.). Cambridge, MA: MIT Press.

Sun, C., Shen, S., Tao, W., Xue, D., & Zhou, Z. (2025). *Noise-resilient symbolic regression with dynamic gating reinforcement learning*, *39*, 20690–20698. https://doi.org/10.1609/aaai.v39i19.34280

Sutton, R., & Barto, A. (2018). *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

Tian, Y., Zhou, W., Viscione, M., Dong, H., Kammer, D. S., & Fink, O. (2025). Interactive symbolic regression with co-design mechanism through offline reinforcement learning. *Nature Communications, 16*, 3930. https://doi.org/10.1038/s41467-025-59288-y

Todorovski, L., & Džeroski, S. (1997). Declarative bias in equation discovery. In *Proceedings of the fourteenth international conference on machine learning*, pp. 376–384. Morgan Kaufmann, San Francisco, CA.

Todorovski, L., & Džeroski, S. (2006). Integrating knowledge-driven and data-driven approaches to modeling. *Ecological Modelling, 194*, 3–13.

Todorovski, L., Dzeroski, S., & Kompare, B. (1998). Modelling and prediction of phytoplankton growth with equation discovery. *Ecological Modelling, 113*(1–3), 71–81. https://doi.org/10.1016/S0304-3800(98)00135-5

Udrescu, S.-M., Tan, A., Feng, J., Neto, O., Wu, T., & Tegmark, M. (2020). AI Feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In *Advances in neural information processing systems*, 33.

Udrescu, S.-M., & Tegmark, M. (2020). AI Feynman: A physics-inspired method for symbolic regression. *Science Advances, 6*(16), 2631.

Uy, N. Q., Hoai, N. X., O'Neill, M., McKay, R. I., & Galván-López, E. (2011). Semantically-based crossover in genetic programming: Application to real-valued symbolic regression. *Genetic Programming and Evolvable Machines, 12*, 91–119.

Valipour, M., You, B., Panju, M., & Ghodsi, A. (2021). SymbolicGPT: A generative transformer model for symbolic regression. arXiv. arXiv:2106.14131 [cs] version: 1. https://doi.org/10.48550/arXiv.2106.14131.

Villar, S., Yao, W., Hogg, D. W., Blum-Smith, B., & Dumitrascu, B. (2023). Dimensionless machine learning: Imposing exact units equivariance. *Journal of Machine Learning Research, 24*, 109.

Wang, R., Jansen, P., Côté, M.-A., & Ammanabrolu, P. (2022). ScienceWorld: Is your Agent Smarter than a 5th Grader?. https://arxiv.org/abs/2203.07540

Wang, C., Chen, Q., Xue, B., & Zhang, M. (2025). Improving generalization of genetic programming for high-dimensional symbolic regression with Shapley value based feature selection. *Data Science and Engineering, 10*(2), 196–211. https://doi.org/10.1007/s41019-024-00270-x

Wang, D., Wang, Y., Evans, L., & Tiwary, P. (2024). From latent dynamics to meaningful representations. *Journal of Chemical Theory and Computation, 20*(9), 3503–3513. https://doi.org/10.1021/acs.jctc.4c00249

Washio, T., & Motoda, H. (1997). Discovering admissible models of complex systems based on scale-types and identity constraints. In *Proceedings of the fifteenth international joint conference on artificial intelligence (IJCAI 1997)*, pp. 810–819.

Wilczek, F., & Devine, B. (2006). *Fantastic realities: 49 Mind journeys and a trip to stockholm*. Singapore: World Scientific.

Williams, K., Bilsland, E., Sparkes, A., Aubrey, W., Young, M., Soldatova, L. N., De Grave, K., Ramon, J., Clare, M., Sirawaraporn, W., Oliver, S. G., & King, R. D. (2015). Cheaper faster drug development validated by the repositioning of drugs against neglected tropical diseases. *Journal of the Royal Society Interface, 12*(104), 20141289. https://doi.org/10.1098/rsif.2014.1289

Xiang, J., & Kim, S. (2013). A* Lasso for learning a sparse Bayesian network structure for continuous variables. *Advances in Neural Information Processing Systems (NeurIPS), 26*, 1466–1474.

Xu, Y., Liu, Y., & Sun, H. (2024). Reinforcement symbolic regression machine. In *International conference on learning representations, ICLR*.

Yang, K., Swanson, K., Jin, W., Coley, C. W., Eiden, P., Gao, H., Guzman-Perez, A., Hopper, T., Kelley, B., Mathea, M., Palmer, A., Settels, V., Jaakkola, T. S., Jensen, K. F., & Barzilay, R. (2019). Analyzing learned molecular representations for property prediction. *Journal of Chemical Information and Modeling, 59*(8), 3370–3388. https://doi.org/10.1021/acs.jcim.9b00237

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., & Liu, T.-Y. (2021). Do transformers really perform badly for graph representation? In *Advances in neural information processing systems, vol. 34*. https://proceedings.neurips.cc/paper/2021/hash/f1c1592588411002af340cbaedd6fc33-Abstract.html

Yu, Y., Chen, J., Gao, T., & Yu, M. (2019). DAG-GNN: DAG structure learning with graph neural networks. In *Proceedings of the 36th international conference on machine learning (ICML)*. Proceedings of Machine Learning Research, pp. 7154–7163.

Zhang, K., & Lipson, H. (2024). Aligning ai-driven discovery with human intuition. arXiv:2410.07397 [cs.LG]

Zhang, R., Meng, Q., & Ma, Z.-M. (2023). Deciphering and integrating invariants for neural operator learning with various physical mechanisms. *National Science Review, 11*(4), 336. https://doi.org/10.1093/nsr/nwad336. Published online: 29 Dec 2023.

Zhang, J., & Spirtes, P. (2016). The three faces of faithfulness. *Synthese, 193*, 1011–1027. https://doi.org/10.1007/S11229-015-0673-9

Zheng, X., Dan, C., Aragam, B., Ravikumar, P., & Xing, E.P. (2020). Learning sparse nonparametric DAGs. In S. Chiappa, R. Calandra, (Eds.) *The 23rd international conference on artificial intelligence and statistics, AISTATS 2020, 26–28 August 2020*, Online [Palermo, Sicily, Italy]. Proceedings of Machine Learning Research, vol. 108, pp. 3414–3425.

Zheng, X., Aragam, B., Ravikumar, P. K., & Xing, E. P. (2018). DAGs with no tears: Continuous optimization for structure learning. *Advances in Neural Information Processing Systems (NeurIPS), 31*, 9472–9483.

Zheng, Y., Huang, B., Chen, W., Ramsey, J., Gong, M., Cai, R., Shimizu, S., Spirtes, P., & Zhang, K. (2024). Causal-learn: Causal discovery in Python. *Journal of Machine Learning Research, 25*(60), 1–8.

Zhong, L., Zhong, J., & Lu, C. (2021). A comparative analysis of dimensionality reduction methods for genetic programming to solve high-dimensional symbolic regression problems. In *2021 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 476–483. https://doi.org/10.1109/SMC52423.2021.9658595

Zhou, M., Song, H., Ye, W., Wang, W., & Lai, Z. (2025). Parameter estimation of structural dynamics with neural operators enabled surrogate modeling. Mechanical Systems and Signal Processing 237, 112914. https://doi.org/10.1016/j.ymssp.2025.112914

Zhu, S., Ng, I., & Chen, Z. (2020). Causal discovery with reinforcement learning. In *International conference on learning representations (ICLR)*. https://openreview.net/forum?id=S1g2skStPB

Zytkow, J.M., Zhu, J., & Hussam, A. (1990). Automated discovery in a chemistry laboratory. In *Proceedings of the 8th National Conference on artificial intelligence (AAAI 1990)*, pp. 889–894. AAAI Press/MIT Press, Washington, DC.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Stefan Kramer[1] · Mattia Cerrato[1] · Jannis Brugger[2] · Sašo Džeroski[3] · Ross D. King[4,5]

✉ Stefan Kramer
kramer@informatik.uni-mainz.de

Mattia Cerrato
mcerrato@uni-mainz.de

Jannis Brugger
jannis.brugger@tu-darmstadt.de

Sašo Džeroski
Saso.Dzeroski@ijs.si

Ross D. King
rk663@cam.ac.uk

[1] Institute of Computer Science, Johannes Gutenberg University Mainz, Saarstrasse 21, 55116 Mainz, Germany

[2] hessian.AI, TU Darmstadt, Karolinenpl. 5, 64289 Darmstadt, Germany

[3] Dept. of Knowledge Technologies, Jozef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia

[4] Data Science and AI, Chalmers University of Technology, Chalmersgatan 4, 41296 Göteborg, Sweden

[5] Department of Chemical Engineering and Biotechnology, University of Cambridge, Philippa Fawcett Drive, Cambridge West CB3 0AS, UK