PAPER · OPEN ACCESS

Incorporating background knowledge in symbolic regression using a computer algebra system

To cite this article: Charles Fox *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 025057

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 025057 https://doi.org/10.1088/2632-2153/ad4a1e

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
10 July 2023

REVISED
18 March 2024

ACCEPTED FOR PUBLICATION
10 May 2024

PUBLISHED
3 June 2024

PAPER

Incorporating background knowledge in symbolic regression using a computer algebra system

Charles Fox[2], **Neil D Tran**[1], **F Nikki Nacion**[1], **Samiha Sharlin**[1] and **Tyler R Josephson**[1,2,*]

[1] Department of Chemical, Biochemical, and Environmental Engineering, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, United States of America
[2] Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, United States of America
[*] Author to whom any correspondence should be addressed.

E-mail: tjo@umbc.edu, charfox1@umbc.edu, ntran3@umbc.edu, fnacion1@umbc.edu and s126@umbc.edu

Keywords: adsorption, symbolic regression, genetic algorithms, Bayesian

Supplementary material for this article is available online

Abstract

Symbolic regression (SR) can generate interpretable, concise expressions that fit a given dataset, allowing for more human understanding of the structure than black-box approaches. The addition of background knowledge (in the form of symbolic mathematical constraints) allows for the generation of expressions that are meaningful with respect to theory while also being consistent with data. We specifically examine the addition of constraints to traditional genetic algorithm (GA) based SR (PySR) as well as a Markov-chain Monte Carlo (MCMC) based Bayesian SR architecture (Bayesian Machine Scientist), and apply these to rediscovering adsorption equations from experimental, historical datasets. We find that, while hard constraints prevent GA and MCMC SR from searching, soft constraints can lead to improved performance both in terms of search effectiveness and model meaningfulness, with computational costs increasing by about an order of magnitude. If the constraints do not correlate well with the dataset or expected models, they can hinder the search of expressions. We find incorporating these constraints in Bayesian SR (as the Bayesian prior) is better than by modifying the fitness function in the GA.

1. Introduction

1.1. Symbolic regression (SR) for scientific discovery

SR generates mathematical expressions that are optimized for complexity and accuracy to a given dataset. Since John Koza pioneered the paradigm of programming by means of natural selection, many applications for SR in scientific discovery have emerged [1]. Unlike other applications of machine learning techniques, scientific research demands explanation and verification, both of which are made more feasible by the generation of human-interpretable mathematical models (as opposed to fitting a model with thousands of parameters) [2–4]. Furthermore, SR can be effective even with very small datasets (∼10 items) such as those produced by difficult or expensive experiments which are not easily repeated. The mathematical expressions produced by SR can easily be extrapolated to untested or otherwise unreachable domains within a dataset (such as extreme pressures or temperatures).

For decades, SR has discovered interesting models from data in many applications including inferring process models at the Dow Chemical Company [5], rainfall-runoff modeling [6], and rediscovering equations for double-pendulum motion [7]. SR has been applied across all scales of scientific investigation, including the atomistic (interatomic potentials [8]), macroscopic (computational fluid dynamics [9]), and cosmological (dark matter overdensity [10]) scales. Some techniques facilitate search through billions of candidate expressions, such as the space of nonlinear descriptors of material properties [11]. While most applications of SR in science focus on identifying empirical patterns in data, such 'data-only' approaches do not account for potential insights from background theory. In fact, some SR works emphasize their

capabilities of discovery 'without any prior knowledge about physics, kinematics, or geometry' [7]. Nonetheless, we posit that prior knowledge need not be discarded, and in this work, we explore how theory may be incorporated into SR to demonstrate machine learning in the context of background knowledge.

1.2. Incorporating background knowledge into SR

One particularly important step towards the effective use of SR in specific domains is the addition of prior knowledge. This step has the potential to take a general-purpose SR algorithm and use it to find novel models with physical meaning. For example, AI-DARWIN uses prior knowledge of chemical reaction mechanisms in the form of predefined functions that a genetic algorithm (GA) may use in its search of equation space, ensuring that each generated model is mechanistically meaningful [12]. This approach specifically encodes the prior knowledge in the form of functions available instead of limitations on functions generated [13]. A number of other approaches exist in the literature that narrow down the equation search space of SR for analyzing scientific data. One is the shape-constrained SR [14, 15], which incorporates constraints on function shape (such as partial derivatives and monotonicity) using an efficient application of integer arithmetic. Additionally, other variations of SR direct the search for unit correctness [16–18], conserve physical properties [17, 19–21], and guide using predefined forms derived from the dataset [19, 22–24]. All of these approaches build on SR based on GAs, the original and most popular technique [1, 13]. In contrast, Engle and Sahinidis developed a deterministic SR algorithm (using mixed-integer nonlinear programming) that constrains the equation space to functions that obey derivative constraints from theory, improving the quality of expressions for thermodynamic equations of state [25]. Another approach is the Bayesian machine scientist (BMS) [26]. BMS rigorously incorporates background knowledge in the form of a Bayesian prior on symbolic expressions; expressions are *a priori* more likely if their distribution of mathematical operators aligns with the distribution of operators in a corpus of prominent equations. However, their approach to the Bayesian prior does not incorporate meaning from particular scientific domains.

Checking the consistency of equations *after* the search is complete is also possible. Previously, we showed that generated expressions can be compared to rich background knowledge (expressed as axioms for the environment under study) by posing generated expressions as conjectures to an automated theorem prover (ATP) [27]. However, state-of-the-art ATPs are too slow to incorporate this logical check as symbolic expressions are generated and, therefore cannot be easily used to bias the search for equations in light of that background knowledge. Moreover, translating scientific theories into a computer-interpretable form is not straightforward.

We address these specific drawbacks by combining SR systems (both GA and Bayesian approaches) with a computer algebra system (CAS) that checks constraints as an equation search is conducted. This is similar to logic guided genetic algorithm (LGGA), which use 'auxiliary truths' (ATs) corresponding to datasets to weigh items in a dataset as well as augment it with more information [28]. LGGA follow an iterative approach of training an arbitrary GA with some dataset, augmenting that dataset with ATs, and training that algorithm again with more informative data. An important distinction between our work and LGGA is that the dataset is not altered in any way, and the addition of extra information is performed during the execution of the GA. We have also used parameter-specific constraints to control the degree of restriction placed on the optimization process, ensuring it remains stable.

1.3. Adsorption

Adsorption, the phenomenon in which molecules bind to a surface, enables chemical processes including carbon capture, humidity control, removal of harmful pollutants from water, and hydrogen production [29–32]. Models of adsorption enable prediction and design of engineered adsorption processes, and many have been proposed over the years (selected equations are shown in table 1) [33–36]. These models relate the amount adsorbed at equilibrium as a function of pressure or concentration and are commonly expressed as equations that are either empirical or derived from theory. For example, the Freundlich isotherm [37] is an empirical function designed to fit observed data, the Langmuir [38] and BET [39] isotherms are derived from physical models, and the Sips [40] isotherm is Langmuir-inspired with empirical terms added for fitting flexibility. We wonder, 'What kinds of models could be generated by a machine learning system, and what role can background knowledge play in the search for accurate and meaningful expressions?'

1.4. Thermodynamic constraints

We consider models to be more *meaningful* when they satisfy thermodynamic constraints on the functional forms appropriate for modeling these phenomena. That is, a random equation that fits data but does not

Table 1. Some well-known isotherms written as SR might find them, and their complexities. Complexity is defined as the number of operators and variables/constants in a given expression.

Isotherm	Literature expression	Symbolic regression form	SR complexity
Langmuir [38]	$\frac{q_{max}K_{eq}p}{1+K_{eq}p}$	$\frac{c_1 p}{c_2+p}$	7
Dual-Site Langmuir [38]	$\frac{q^a_{max}K^a_{eq}p}{1+K^a_{eq}p}+\frac{q^b_{max}K^b_{eq}p}{1+K^b_{eq}p}$	$\frac{c_1 p}{c_2+p}+\frac{c_3 p}{c_4+p}$	15
BET [39]	$\frac{v_m * c * (p/p_0)}{(1-p/p_0)*(1+(c-1)p/p_0)}$	$\frac{c_1 p}{p^2+c2p+c3}$	13
Freundlich [37]	$c_1 p^{\frac{1}{n}}$	$c_1 p^{c_2}$	5
Sips [40]	$\frac{c_1 p^{\frac{1}{n}}}{1+c_1 p^{\frac{1}{n}}}$	$\frac{p^{c2}}{c_1+p^{c2}}$	9

approach zero loading correctly, is less trustworthy outside the training data than an equation constrained to follow thermodynamics. We have identified three constraints relevant for single-component adsorption [27]:

$$\lim_{p\to 0} f(p) = 0 \tag{1}$$

$$\lim_{p\to 0} f'(p) < \infty \tag{2}$$

$$\forall p > 0 \qquad f'(p) \geqslant 0. \tag{3}$$

Constraint 1 ensures that, in the limit of zero pressure, all molecules must desorb, and loading cannot be negative. Constraint 2 requires that in the limit of zero pressure, the slope of the isotherm must be a positive finite constant. Talu and Myers show that, as pressure approaches zero, the slope of the adsorption isotherm equals the adsorption second virial coefficient B_{1S}, which characterizes the interaction between one molecule and the surface, and must be a finite positive number [41, 42]:

$$\lim_{p\to 0} \frac{\mathrm{d}f}{\mathrm{d}p} = \frac{B_{1S}}{RT} = c \tag{4}$$

Constraint 3 requires that loading does not decrease with increasing pressure (the isotherm is monotonically non-decreasing) for all ($\forall$) positive values of pressure. Note that this does not hold for mixture adsorption (in which competition plays a role), nor in BET adsorption, which exhibits a discontinuity at the saturation pressure, instead of a monotonic increase.

1.5. PySR: SR using GAs

PySR, Python for SR, is a Python library that uses a GA for SR [43]. PySR is a Python wrapper that calls a Julia library, SymbolicRegression.jl (SR.jl), for numerical performance. Due to the nature of the modifications needed to the algorithm for this work, the base Julia library was used, but all added functionality should be inherited by the Python wrapper library as well.

The basic premise is that one or more populations of models move towards more optimal solutions via random mutations. At each generation, some members of a population are removed based on their fitness, age, or some other criteria (PySR replaces the oldest members). Beneficial solutions are encouraged by having more optimal members of a population mutate and reproduce.

Changes include mutating a single constant or operator, simplifying the expression, or performing crossover between two expressions (figures 1 and 2). PySR uses multiple populations in a method similar to the island methodology [44]. This aims to allow for specialization by separately evolving unique populations, occasionally allowing some members to move between them to share that specialization. Specifically, PySR implements the so-called hall of fame (HOF), which is a Pareto front built from the best members across each population. After a number of generations, each population submits its top 10 best members (based on score), which are then compared and pared down via Pareto front. Expressions that remain in the HOF are used for future mutations in each of the populations.

1.6. Bayesian symbolic regression (BSR)

The BMS by Guimera *et al* [26] approaches SR from a Bayesian perspective. BSR frames the search for accurate, concise, and informed models as sampling the marginal posterior distribution of symbolic models with respect to a prior and fit to a dataset. Markov chain Monte Carlo (MC) is used to generate new expression trees (figure 3), which are accepted or rejected based on their likelihood. The authors define three MC moves: node replacement, root addition/removal, and elementary tree replacement, which together enable the construction of expression trees while maintaining detailed balance, ensuring proper sampling of the posterior.

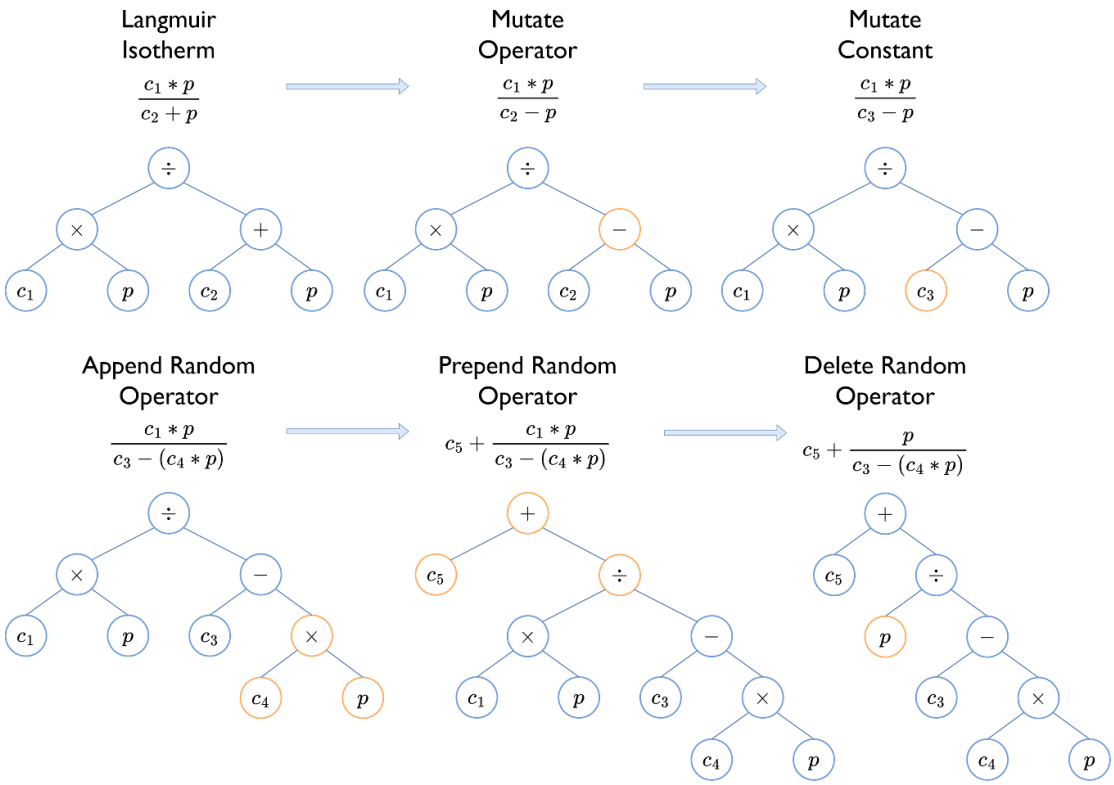

Figure 1. All mutations (except for random tree generation and simplification) in PySR in succession (read from left to right, top to bottom). Changes from each previous expression tree are shown in orange.

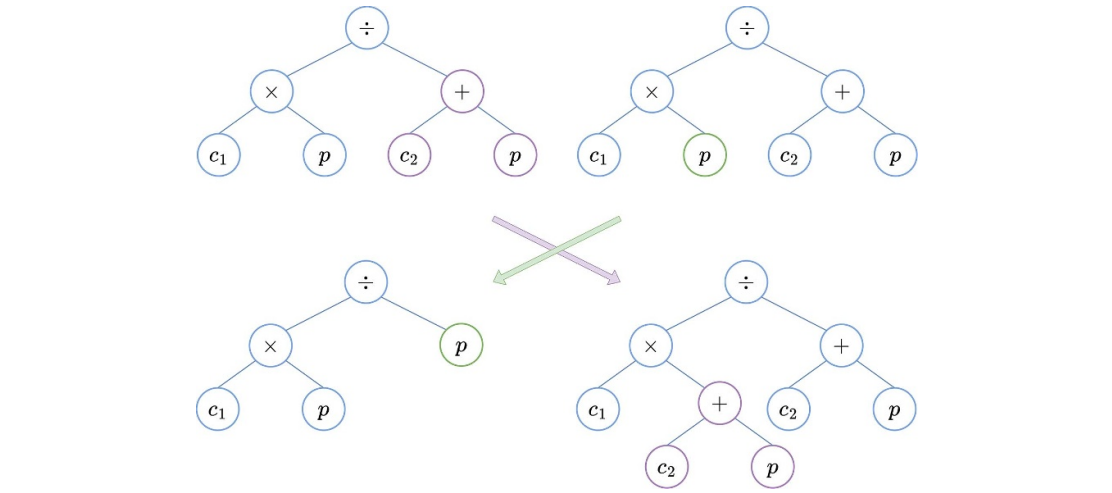

Figure 2. An example of the crossover mutation between two expression trees.

Specifically, the probability of some model given some data is defined as:

$$p(f_i|D) = \frac{1}{Z}\int_{\Theta_i} \mathrm{d}\theta_i\, p(D|f_i,\theta_i)\, p(\theta_i|f_i)\, p(f_i) = \frac{\exp\left[-\mathcal{L}(f_i)\right]}{Z} \tag{5}$$

where Z is the probability of the dataset $p(D)$, Θ_i is the space of possible values for parameters θ_i and $\mathcal{L}$ is the description length of the model.

A central idea in BSR is the inclusion of a prior to emphasize expressions that are *a priori* more likely than others, regardless of the data. Guimera *et al* based their prior on a corpus of 4080 mathematical expressions collected from Wikipedia (from the 'list of scientific equations named after people'), and assigned the prior

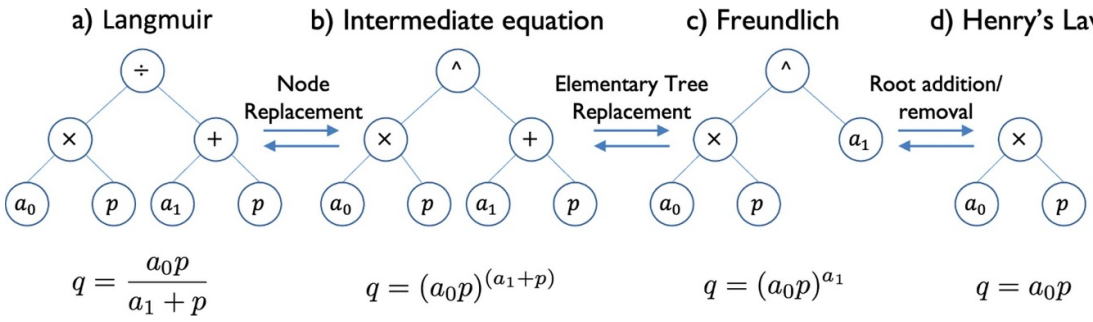

Figure 3. Illustrating the moves available to the BMS algorithm, as applied to adsorption equations. In contrast to the mutations available in PySR, these transformations satisfy detailed balance [26].

likelihood of a function $p(f_i)$ (and its 'energy' EP) using the counts of each unique operator (n_o) in the corpus, by fitting parameters α and β like so:

$$\mathrm{EP} = -\log\left(p\left(f_i\right)\right) = \sum_{o \in O}\left[\alpha_o n_o\left(f_i\right) + \beta_o n_o^2\left(f_i\right)\right] \tag{6}$$

While this method leads to a distribution of expressions that resembles the corpus when run with no data, $p(f_i)$ can also be set to a constant value so that there is no bias based on operators present in the search process. For our problem, we crafted a prior especially for adsorption thermodynamics (see details in Methods).

An important subtlety of the BSR methodology is that while there is no explicit Pareto front that the algorithm seeks to optimize, both the loss and complexity of each expression contribute to the energy. That is, the algorithm should try to move in a direction that optimizes both the accuracy and simplicity of the expressions discovered, while not directly representing that direction through a Pareto front.

2. Methods

2.1. Checking thermodynamic constraints

Three constraint checking functions for the thermodynamic constraints described in section 1.4) were developed using the Python library SymPy, an open-source CAS [45]. Each function returns either true or false, depending on if its constraint is met or not (if a time limit is exceeded, the constraint is returned as false). For both PySR and BSR, we found that hard constraints (rejecting every expression that fails any constraint) severely hinder the search process, cutting off intermediate expressions between better expressions that may also pass the constraints. Consequently, we impose these as 'soft' constraints, penalizing expressions for constraint violation, without outright rejecting them. References [14, 15] also found soft constraints to be more effective than hard constraints. This approach (as implemented in PySR) is detailed in algorithm 1 (in SI).

Constraints 1 and 2 could be checked using SymPy's limit and derivative functionality, but Constraint 3 was more challenging. Though SymPy can check if an expression is strictly increasing in a given range, the check for monotonicity returns false if any change in curvature (critical point) exists for the expression—thus preventing functions such as x^3 from being considered monotonically non-decreasing. To allow for zero slope, we implemented a custom monotonic non-decreasing check function (see algorithm 2 in SI). Instead of just checking the slope in one range, it checks the ranges between all critical points (as well as to the start and end of the original range in question).

We hypothesize that the 'equation space' explored by SR includes accurate, but not thermodynamically consistent expressions that can be rejected through the incorporation of background knowledge, guiding the search to more theory-informed expressions.

The data used for our experiments come from the original papers which proposed those isotherm models. Because we aim to generate expressions consistent with background theory and rediscover literature expressions, our *test is the output expression* [46]. We thus use the entire dataset in each SR run; we do not split the data into training and testing sets. Propensity for overfitting may still be analyzed through the Pareto front plots, where overfit models have similar accuracy to their simpler counterparts. For those aiming to develop expressions without a known target expression, accuracy with respect to unseen data would be important, and cross-validation can be used to assess model performance [7, 26].

2.2. PySR modifications

In PySR, each member in a population has a score to be minimized, which combines the loss and complexity (defined by total nodes in the expression tree). When a thermodynamic constraint is violated, we multiply the loss function by a penalty, raising the score and making the expression less fit. This allows any number of constraints to be checked in any order (as multiplication is commutative), and confers larger penalties to expressions that violate multiple constraints,

$$\text{Loss: } L = \ell_2^R * \prod_{i=1,2,3} c_i^{\delta_i} \text{ where } \delta_i = \left\{ \begin{array}{ll} 1 & \text{if constraint } i \text{ failed} \\ 0 & \text{if constraint } i \text{ passed} \end{array} \right\} \tag{7}$$

$$\text{Member Score: } S = L + n_{\text{nodes}} * c_l. \tag{8}$$

The above equations detail how the loss and score are calculated in PySR. ℓ_2^R is the L2 norm, c_i is the penalty for constraint i, δ_i indicates if constraint i is passed and c_l is the penalty for the length / complexity of an expression. For this work, all constraint penalties for PySR were set to 1.3 so each constraint increases the loss by 30%.

PySR also has the option to take any operators defined in Julia or Python, including custom user-defined operators. For this work only the operators $+$, $-$, $*$ and $\div$ were used to manage the size of the search space. Expressions written in their canonical form may use other operators such as exponents but these are only due to simplification of generated expressions.

2.3. BMS modifications

The prior used in the BMS code by Guimera *et al* [26] incorporates 'background knowledge' in its equation search by considering mathematical operation frequency among named equations in Wikipedia. The authors found this to be helpful for searching for general scientific equations, but we aim to color this background knowledge according to our domain of inquiry. We consider the thermodynamic constraints described above to be our 'prior knowledge,' and construct the following expression:

$$\text{EP} = \sum_{o \in O} \left[c_{\text{ops}} n_o \left(f_i \right) \right] + \sum_{i=1,2} c_i * \delta_i \text{ where } \delta_i = \left\{ \begin{array}{ll} 1 & \text{if constraint } i \text{ failed} \\ 0 & \text{if constraint } i \text{ passed} \end{array} \right\} \tag{9}$$

where c_{ops} is the constraint penalty for operators (analogous to the parsimony parameter in PySR), and n_o is the count of each operator in expression f_i. This expression directly replaces equation (6), changing the prior distribution. Constraint penalties of 10 and 5 were used for the first and second constraints. Note that we checked all three constraints with PySR, and only the first two constraints with BSR (omitting the monotonic non-decreasing check).

2.4. Simulation details

Each BSR simulation consists of 50 000 steps at the single computational temperature $T = 1$, with penalties for violating constraints 1 (c_1) and 2 (c_2) being 10 and 5, respectively, and complexity penalty (c_{ops}) of 2. Each PySR simulation was run for 50 iterations with 8 runs in parallel. Parsimony and crossover were kept at their respective default values of 0.0032 and 0. All available data was used when running each SR algorithm.

3. Results

3.1. Datasets

To examine the effects of adding constraints to SR during a search, four experimental adsorption datasets were identified: adsorption of nitrogen and methane on mica [38], adsorption of isobutane in silicalite, [47] and adsorption of nitrogen on Fe–Al_20_3 catalyst [39]. The first and second datasets come from the landmark paper introducing the Langmuir isotherm model [38]. This model assumes there are discrete loading 'sites' that do not interact with each other, and that each site can either be occupied or not. The isobutane dataset is well-described by a dual-site Langmuir model which has two unique types of sites [48]. The fourth dataset (referred to as the BET dataset) was used by the authors of BET theory to support their model for multilayer adsorption. These data and the respective target model fits are shown in figure 4.

3.2. Langmuir datasets

The main results of this work are shown in two plot types. The left column contains Pareto fronts which show the best expressions based on complexity and accuracy. In these, the horizontal axis shows increasing complexity (defined as the total number of nodes in an expression tree), and the vertical axis shows loss, which is logarithmically scaled so the trend of the Pareto front is more apparent. The best expression at each

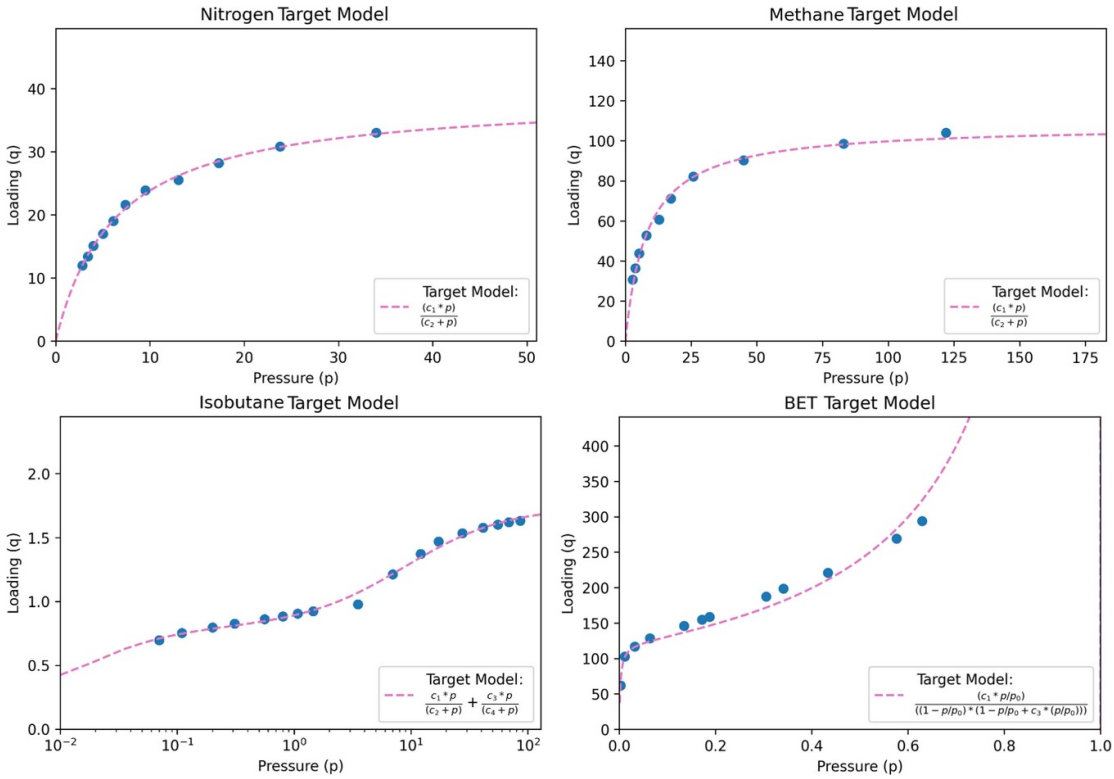

Figure 4. Each dataset and corresponding target model with constants fit using SciPy. Isobutane is shown with log scaled pressure so that the two separate curves are visible. BET is shown with pressure increasing to 1 so the asymptote is visible.

complexity is taken from each of 8 runs (gray curves), with the overall Pareto front shown in orange. The target expression for each dataset is also shown in the form it would likely be expressed by SR, along with loss found after fitting constants. The right column of each figure shows the dataset and select expressions from the overall Pareto front for that test. Only some expressions are shown so plots remain readable and because expressions longer than the target model are usually overfit and overlay the target model expression too closely for distinction. The target model is plotted with a dotted line so that expressions with similar accuracy can still be seen. Plotting the generated expressions on the data helps to illustrate how they may or may not follow the thermodynamic constraints and how similar they are to the target model.

Figure 5 shows the results from both SR algorithms with constraints on and off on the Langmuir nitrogen dataset. The first and second rows show BSR and PySR respectively with constraints off and clearly show that BSR finds the target model while PySR does not. The expression that defines the corner at complexity 7 in the BSR Pareto front plot (figure 5(a)) is indistinguishable from the target model (both written mathematically and drawn on the data) when viewed in the isotherm plot (figure 5(b)). The BSR plot (figure 5(a)) has a much larger variance in terms of best Pareto fronts across 8 runs (as shown by the grey lines) than PySR, but this may indicate longer time needed for the algorithm to converge. The corresponding isotherm plots (the right column) show how expressions fit the data better as they become more complex, following the general trend of the Pareto fronts. These plots also show how some expressions can fit the data reasonably well while violating the constraints from theory, as is the case in the plot for PySR (figure 5(d)). In fact, only 2.2% of expressions generated by PySR (without enforcing constraints) pass the first constraint and only 33% pass the second constraint (table 2). Without constraints enforced, BSR finds more consistent expressions than PySR, with 37% of its expressions passing the first constraint and 67% passing the second (on the Langmuir datasets). A similar pattern can be seen for the other datasets examined. This improved consistency can be attributed to BSR's tendency to generate higher-quality distributions of equations, as observed by Guimera [26]. While the datasets are relatively clean, they do contain some noise. When models are overfit to this noise, there is a higher likelihood of violating constraints, such as introducing a discontinuity.

When the thermodynamic constraints are enabled, the effect is clearly shown in the Pareto fronts (bottom two rows). Both SR methods find the target model and achieve the same or similar accuracy (accuracy is less for the same expression when the constants were not optimized as thoroughly in the search). Datasets that are well represented by the Langmuir isotherm show the effects of the constraints well because it is typically very accurate as well as being concise. The isotherm plots show, as before, how the expressions

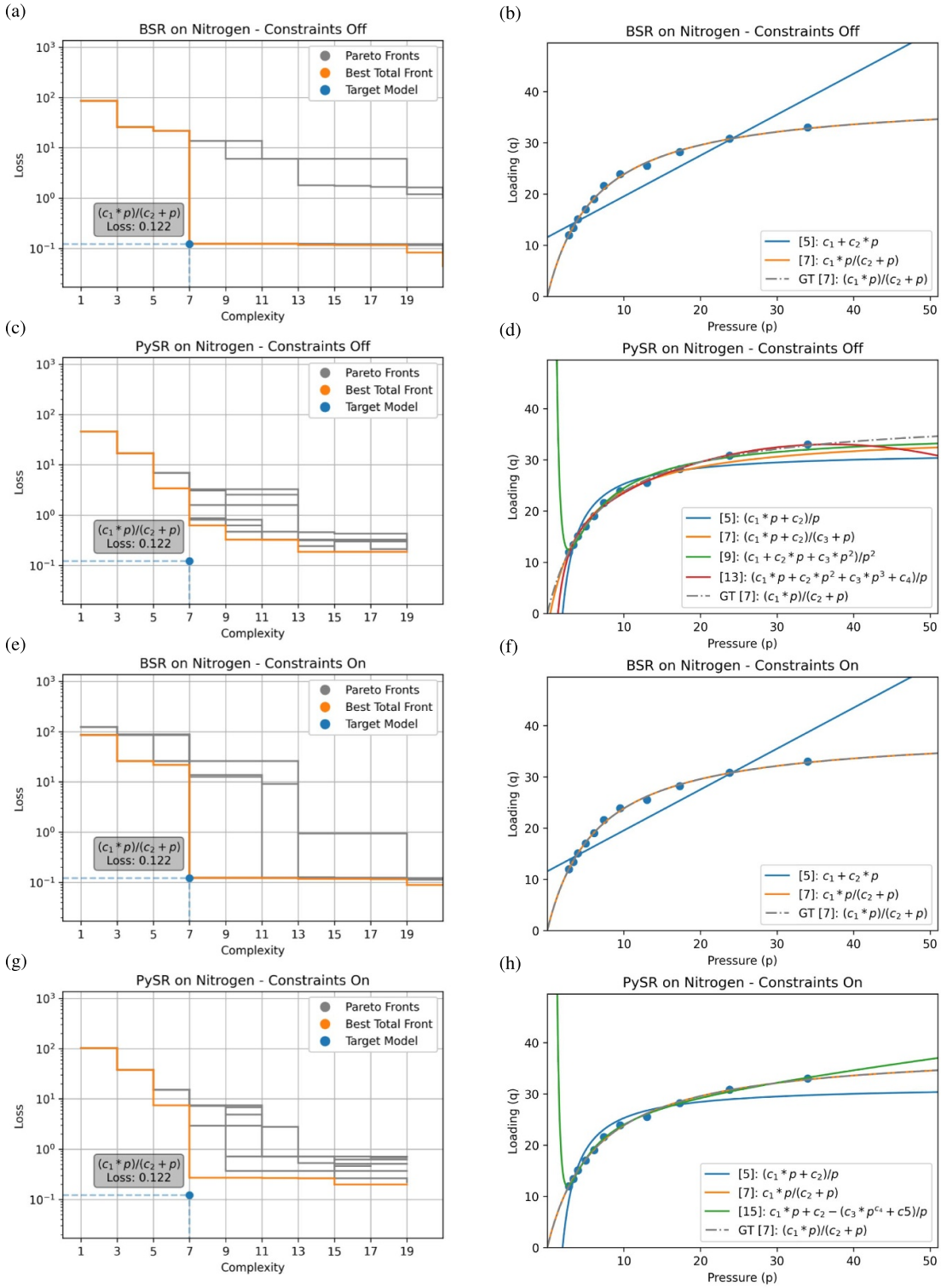

Figure 5. BSR and PySR on the Nitrogen dataset. The left column shows combined Pareto fronts across 8 runs and the right column shows interesting isotherms found at the defining corners of those Pareto fronts. The constraints are disabled in the top four subplots and enabled in the bottom four. The rows alternate between BSR and PySR.

fit the data better as they become more complex but showing anything beyond a complexity of 7 is redundant as the target model is discovered and matches the pre-fit target model almost perfectly. The trend of slightly more variation across BSR runs also continues here to some extent and the variation across PySR runs appears roughly similar to with constraints disabled. Importantly, PySR sees a 5x increase in expressions passing the first constraint (though still only 10%) and a marginal improvement across the other two constraints (8% and 13%). The change is more stark in BSR where twice as many expressions now pass the

Table 2. Percentage of expressions generated passing each of the three constraints. Results are shown across both SR methods, all datasets and with constraints active and disabled. It is important to note that while the results can be compared, constraint 3 was not used when running BSR, so any expressions which do satisfy that constraint result from only the first two constraints.

Dataset	Constraints active	BSR C1	BSR C2	BSR C3*	PySR C1	PySR C2	PySR C3
Nitrogen	False	37%	67%	0.46%	2.2%	33%	46%
Nitrogen	True	72%	73%	19%	10%	41%	59%
Methane	False	33%	51%	0.51%	4.1%	48%	61%
Methane	True	59%	59%	2.5%	5.7%	54%	62%
Isobutane	False	24%	46%	0.45%	3.4%	65%	68%
Isobutane	True	36%	36%	1.3%	5.5%	56%	58%
BET	False	16%	92%	0.09%	6.2%	35%	79%
BET	True	85%	92%	2.4%	4.7%	30%	81%

first constraint (up to 72%) and a significant portion pass the third constraint (up to 19% from 0.46%) even though it was not included in the Bayesian prior.

While the results are mostly similar for the methane dataset, there are some important differences. Like with the nitrogen dataset, BSR finds the target model without constraints enabled while PySR does not. This is apparent in the Pareto fronts (figures 6(a) and (c)). In this case, PySR finds an expression with complexity 9 with more accuracy than the target model, though with an extra constant in the numerator, it violates the thermodynamic constraints (figure 6(d)). Imposing the constraints penalized the loss for this expression relative to the target model, but not enough to overcome the increased accuracy (figure 6(h)). As with nitrogen, BSR does a better job of finding expressions that pass the constraints, even when they are not enabled, as it finds 33% passing the first and 51% passing the second (where PySR finds 4.1% and 48% respectively).

3.3. Isobutane dataset

Unlike the methane and nitrogen datasets (figures 5 and 6) which are best modeled by the Langmuir isotherm, the isobutane dataset (figure 7) is best modeled by the dual-site Langmuir isotherm, which has twice the complexity. Despite this significant complexity, the dual-site Langmuir isotherm is not significantly more accurate than many expressions shorter than it. This is best seen in figures 7(c) and (g) which show the Pareto fronts for PySR with constraints off and on respectively. In both plots, expressions with half the complexity reach almost the same accuracy, creating a plateau from complexity 7 onward. This is also shown well in the corresponding isotherm plots which show that the expressions found at complexity 7 match the data as well as the target model. Importantly, these expressions do not satisfy the thermodynamic constraints.

Unlike PySR, BSR does not find expressions with accuracy close to the target model until the same complexity.

For BSR, including constraints shifts the whole Pareto front down (figures 7(a)–(e)), indicating that more accurate expressions were found at many complexity levels. While PySR did not find accurate expressions consistent with the constraints, BSR did. In this case, BSR finds the target model expression while PySR does not. This is not apparent on either the Pareto fronts or isotherm plots however, because the accuracy of the expression found is about 10x worse than the fit target model and the best expressions found at that complexity. This is likely because, while the target model is found, the form it was originally produced in (before being simplified) is much more complex.

In PySR, penalizing expressions that violate constraints actually led to populations of equations that violated constraints two and three more often, with a decrease of about 10% in each case (see table 2). This was surprising—we anticipated that imposing penalties would lead to fewer violating expressions, but the opposite occurred. For BSR as well, including constraints in the prior actually led to a decrease in expressions satisfying the second constraint (from 46% to 36%), and a slight increase in the first and third constraints. We hypothesize that the penalty for overly complex models may be too strict for the isobutane dataset, meaning that models which are accurate, and which follow all the constraints are less likely. Another reason for this behavior may be how the constraints restrict movement through the space of expressions (by penalizing expressions that may be key steps to model improvement).

3.4. BET dataset

The BET dataset is unique because the target model expression diverges to infinity as the pressure approaches 1 (pressure in this case is relative vapor pressure, p/p^{sat}; the vapor being adsorbed becomes a liquid as $p/p^{\text{sat}} \to 1$. So in this case, the third constraint (that it is monotonically non-decreasing) no longer holds for all pressure (seen in figure 8). Nonetheless, we found that whether or not constraints were enabled, many of the most accurate expressions generated by PySR for this dataset pass the third constraint (78.65% without

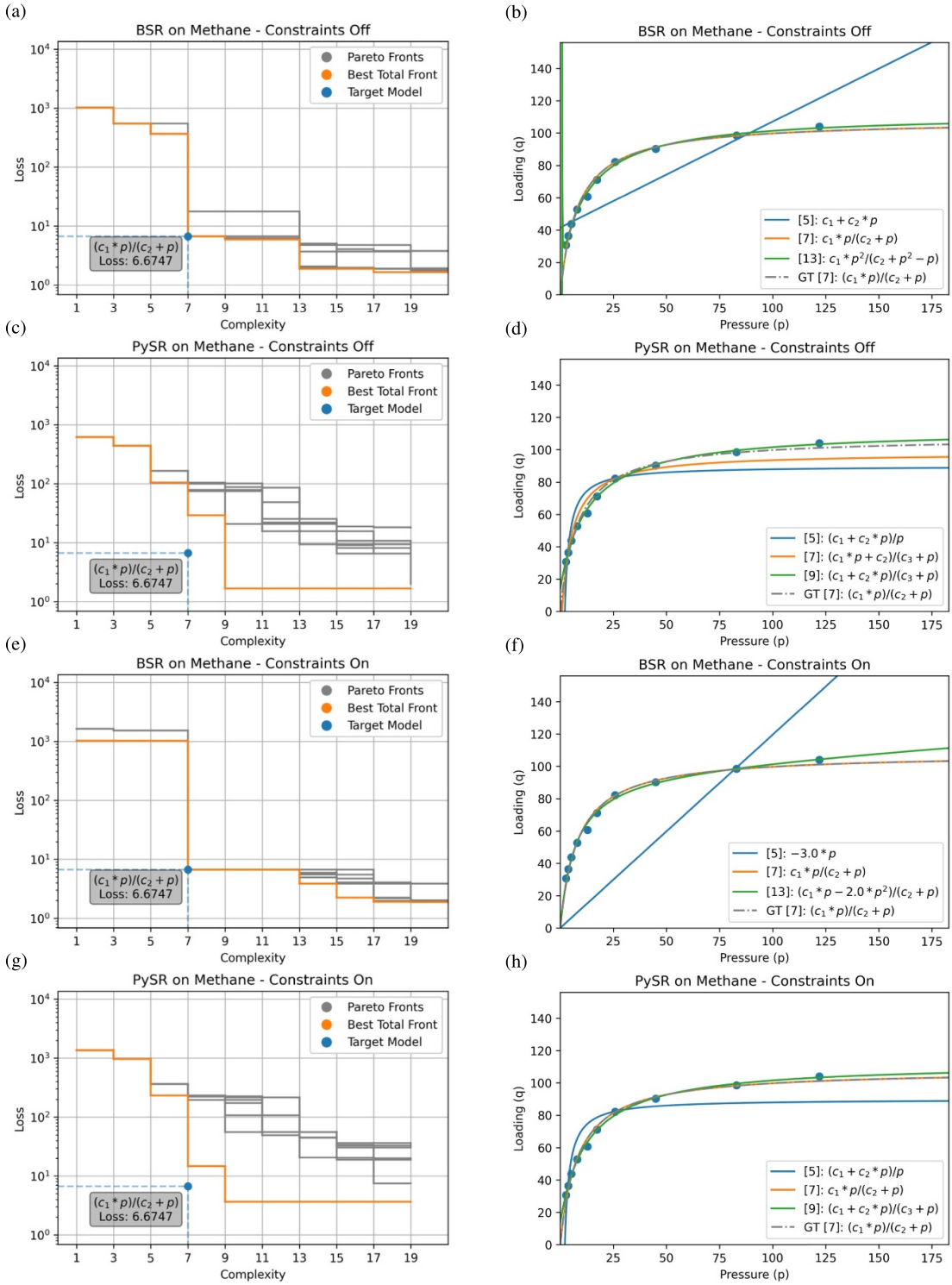

Figure 6. BSR and PySR on the Methane dataset. The left column shows combined Pareto fronts across 8 runs and the right column shows interesting isotherms found at the defining corners of those Pareto fronts. The constraints are disabled in the top four subplots and enabled in the bottom four. The rows alternate between BSR and PySR.

constraints and 81.29% with), contrary to the target model theory. Furthermore, PySR satisfies the first two constraints less frequently with constraints on compared to with constraints off. One possible explanation for this behavior is that the dataset itself is more easily fit by expressions with expressions that are monotonically non-decreasing, at least from the perspective of the PySR algorithm. Overall, while PySR can find accurate expressions for the BET dataset, it fails to find expressions that also follow the constraints, even when they are enabled.

In contrast, BSR did not generate many expressions that were monotonically nondecreasing, and the incorporation of constraints had a substantial effect on the search. Specifically, the second constraint is

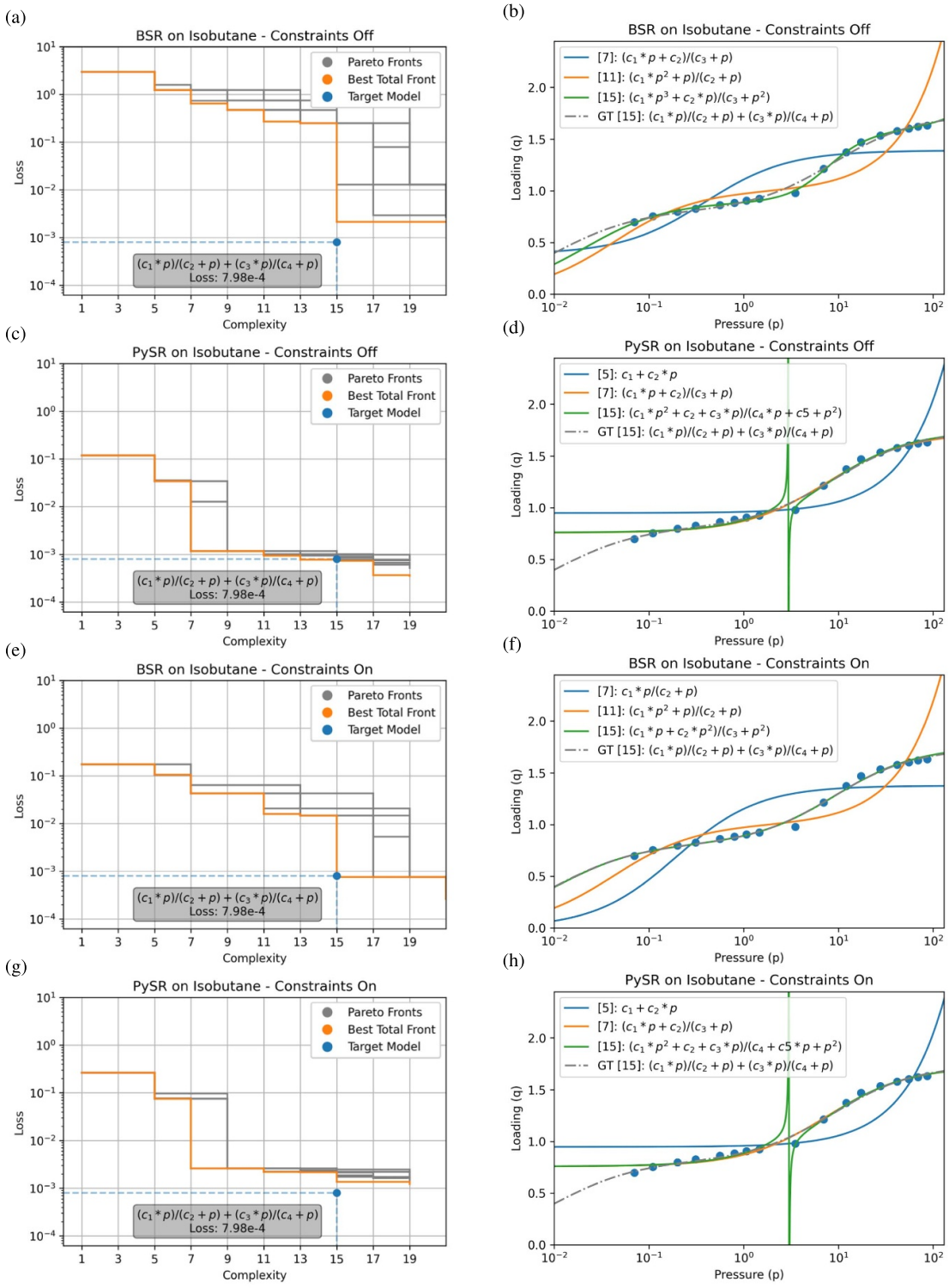

Figure 7. BSR and PySR on the Isobutane dataset. The left column shows combined Pareto fronts across 8 runs and the right column shows interesting isotherms found at the defining corners of those Pareto fronts. The constraints are disabled in the top four subplots and enabled in the bottom four. The rows alternate between BSR and PySR.

passed about 92% of the time both with it enabled and disabled and the portion passing the first constraint increases dramatically from 16% to 85% once it is enabled. This leads to a large number of models which agree with the requisite constraints for BET, but none of these are the target model rediscovered. Instead, many expressions with close to (or better than) the accuracy of the target model are found by both algorithms in both cases, none of the isotherms plotted appear similar. The asymptote at a partial pressure of 1 is not replicated by any similarly accurate expressions and the slight curve of the target model in the middle of the dataset is also absent. These results together seem to indicate that the constraints, while

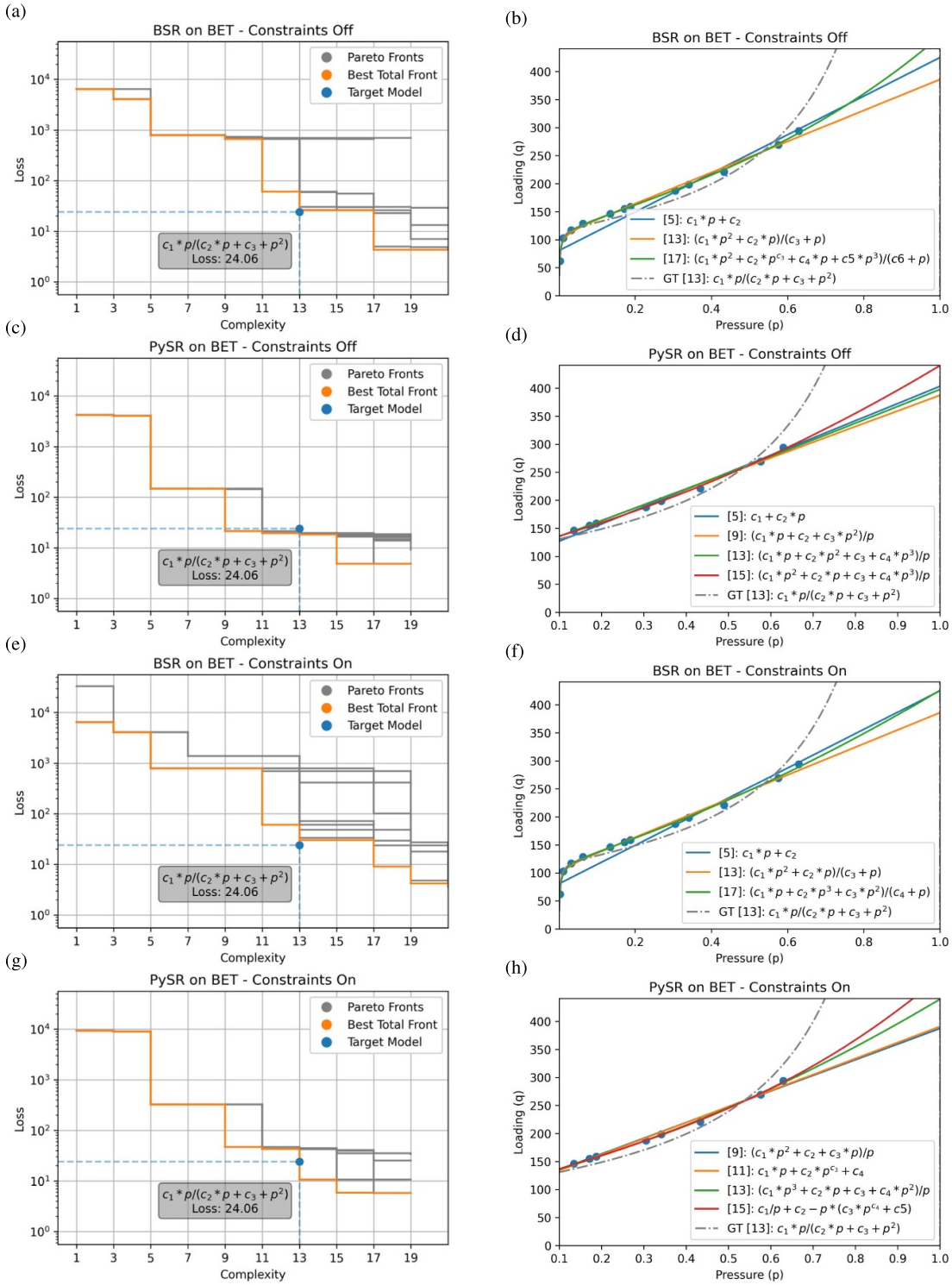

Figure 8. BSR and PySR on the BET dataset. The left column shows combined Pareto fronts across 8 runs and the right column shows interesting isotherms found at the defining corners of those Pareto fronts. The constraints are disabled in the top four subplots and enabled in the bottom four. The rows alternate between BSR and PySR.

thermodynamically correct, do not provide enough information (or even provide contradictory information) for rediscovering the BET target model expression.

4. Discussion

4.1. Effectiveness

This work highlights that sometimes, a stochastic search through equation space finds equations that are superior to target expressions—in our case achieving comparable accuracy to the target expressions while

also being less complex (shorter expression length). This is particularly observed in the case of isobutane (figure 7); both with constraints on and off, PySR finds the expression $\frac{c_1 p + c_2}{c_3 + p}$, which fits the data well but diverges from the target expression as it approaches 0. But many of these expressions, while consistent with the data, are inconsistent with thermodynamics, as they violate our tested constraints. We have demonstrated that accounting for these constraints in the search process can guide the population or distribution of expressions toward thermodynamically consistent expressions, and aid in the identification of the target expression. Sometimes this leads to more consistent equations, and sometimes it does not improve the search at all.

While this work does not exhaustively examine how successful this approach would be across datasets and constraints of different characteristics, we can speculate about why this approach works and does not work in different circumstances. Most of the time, adding a constraint led to an increase in that constraint's satisfaction in the population or distribution of expressions (table 2), and increased the likelihood of finding the target expression (e.g. PySR on the Langmuir datasets, and BSR on the isobutane dataset). However, in some cases, the population had more expressions violating that constraint (table 2). For example, adding constraints to both PySR and BSR runs with isobutane led to more violations of the second constraint (finite slope as $p \to 0$), on average. This may have occurred because, with multiple constraints being applied simultaneously, the algorithms converged upon expressions fitting the data and the first constraint (zero loading at zero pressure) instead of satisfying the second constraint, as well. Nonetheless, incorporating the constraints into the isobutane runs enabled identification of the target model; we emphasize that 'constraint satisfaction in the population' is a different metric from 'the target expression was recovered.'

Biasing the search to satisfy constraints, then measuring effectiveness according to those constraints, has us reflecting on Goodhart's law, 'When a measure becomes a target, it ceases to be a good measure' [49]. Goodhart's law invokes caution because targeting particular metrics can lead to unintended consequences. This would most obviously be a problem if unnecessary constraints were added. However, Goodhart's law becomes less relevant in multiobjective scenarios; simultaneously addressing numerous objectives mitigates over-emphasis on few. SR typically targets 1) accuracy on training and validation data and 2) short expressions. We add another metric, 3) satisfaction of constraints, and while this appears in the objective function so as to bias the equation search, we anticipate unintended consequences will be mitigated by continuing to include targets (1) and (2) above.

4.2. Exceptions to constraints

While the three constraints presented in this work do follow from a broader thermodynamic theory, not all isotherm models in this work satisfy all the constraints. Specifically, the BET isotherm does not satisfy the third constraint because it reaches an asymptote as p/p_0 approaches 1. While this does break the monotonicity constraint, it also seems reasonable when considering what p_0 represents (the pressure at which the adsorbate becomes a liquid and the interaction fundamentally changes). This raises the question: what constraints to include and why? The decision to test if expressions pass the third constraint necessarily excludes the BET model because, while it is theoretically grounded, it only applies from in the range $(0, p_0)$. Furthermore, the data used does not extend past that range so expressions that do pass the third constraint may not appear much different in terms of what is relevant. We recommend carefully examining what constraints one may want to test and in what places they should actually be checked. Introducing incorrect constraints may hinder the search with our biases, and prevent algorithms from discovering phenomena outside our assumptions.

4.3. Computational complexity/runtime

As seen in figure 10 (in SI), consideration of constraints increases runtime by an order of magnitude; this is even after we carefully integrated the CAS into the SR algorithms to reduce overhead, and leveraged memory to avoid redundant checks on expressions previously visited. On average, numerically checking models is much faster than manipulating them symbolically (especially for larger expressions) – checking *every* new expression is quite expensive. This is unfortunately necessary if the constraints are to be considered as an integral part of the search. If a cheaper solution is needed, the search can be performed without constraints, and constraints checked after the fact. In fact, this approach enables even more elaborate methods of considering background knowledge, such as comparing against complex, multi-premise background theories using an ATP.

4.4. Challenges around complexity, simplification, and canonical form

In this work, simplification is necessary in order to identify whether a generated expression matches the ground truth and to assign generated expressions an appropriate complexity. We augmented SymPy's 'simplify' function, to shorten the numerous rational expressions we generated into a 'canonical form'

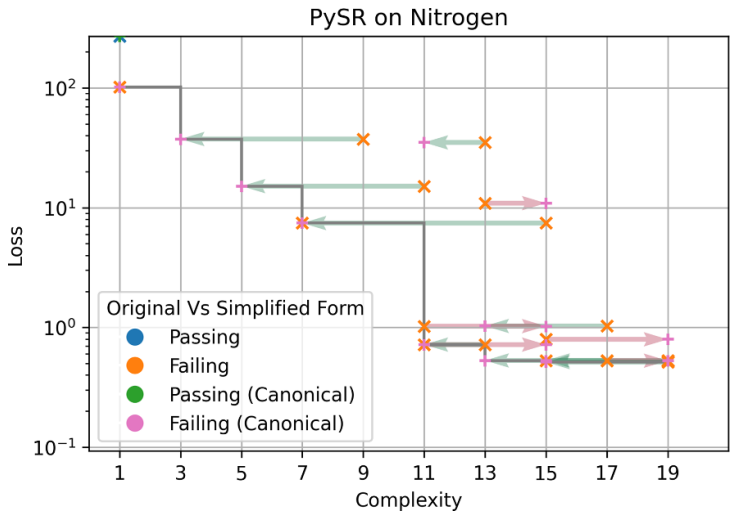

Figure 9. The effect of the canonical form checking function on a Pareto front showing the results from PySR on the Nitrogen dataset. Points on this plot are marked as 'Passing' only if they pass all constraints checked. Points marked with a blue or orange X are represented at their originally generated complexity, and points marked with a green or pink + are the 'canonical' form of those expressions. This canonical form tends to simplify these expressions, although not always.

(details in the supporting information). While some methods such as BSR attempt simplification during runtime, PySR does not because of the added computation needed per expression. Generating a 'canonical form' for expressions generated by PySR sometimes increases, and sometimes decreases, the complexity. Some expressions are generated as complex expression trees that are much more complex than their canonical forms (figure 9).

Simplification is a crucial challenge of this work because complexity plays a significant role in SR. After all, models are compared via accuracy and complexity to make decisions during the search. A single model may very well take on different scores / likelihoods because of how it is written, influencing not just its standing, but subsequent steps in the search. Ideally, every model would always be written in the simplest form, but this is computationally intractable in some circumstances [50]. Because of this, comparing functions based on behavior (symbolic constraints) may be more appropriate, because limiting behavior is invariant to the numerous ways an expression can be written.

4.5. Reducing underspecification through inductive biases

Machine learning researchers at Google recently highlighted the role of underspecification in machine learning pipelines [51]. They suggested that one way to combat underspecification is to use credible inductive biases to allow for selection from otherwise similarly effective models, and that these constraints should have little negative effect on accuracy if selected correctly. In this work, we find expressions that are roughly equivalent in terms of accuracy and complexity but have different functional forms, leading to different behavior outside the range of the data—signatures similar to those discussed in [51]. We find that adding thermodynamic constraints can help improve the search for good expressions, but this does not necessarily restrict the hypothesis space in the same way that inductive biases do; we were unable to effectively search with hard constraints, and so our hypothesis space still included expressions that are inconsistent with constraints. Instead, we can reduce the hypothesis space after the search is complete; by rejecting accurate-but-inconsistent expressions using our background knowledge, we improve on the issues of underspecification. Nonetheless, for datasets with reasonably complex behavior, there still exist multiple distinct thermodynamically-consistent expressions of similar accuracy and complexity. The space of equations defined by the limited number of operators considered here, even for one dimensional datasets, is just that vast!

5. Conclusions

In this work, we couple a CAS to two SR algorithms in order to check the consistency of generated expressions with background knowledge. We find that including appropriate mathematical constraints can improve search effectiveness or break the search entirely, depending on the dataset and implementation details. Although computational costs increase by an order of magnitude, tightly integrating SR with a CAS is a practical way to check for constraints on each expression generated during the search.

We have shown that consideration of constraints helps in rediscovering ground-truth isotherm models from experimental data, including the Langmuir and the dual-site Langmuir isotherms (though the dual-site Langmuir isotherm was not identified on the Pareto front, it was present in the generated models). In contrast, the BET isotherm was not rediscovered; more accurate and concise models were generated instead, and the most meaningful model (BET) was consequently missed. We found that Bayesian SR is a more effective and intuitive platform for incorporating symbolic constraints in a Bayesian prior, rather than by modifying the fitness function in traditional GAs; the resulting populations of expressions were more attuned to the constraints with BSR. Finally, though background knowledge can screen out accurate yet inconsistent solutions, SR pipelines remain underspecified in our context, capable of generating multiple distinct solutions with similar performance and adherence to constraints.

Data availability statement

The data that support the findings of this study, as well as the code developed in the course of this work, are openly available at the following URL/DOI: https://github.com/ATOMSLab/pySR_adsorption.

Acknowledgments

We thank Marta Sales-Pardo and Roger Guimerà for discussions about the Bayesian Machine Scientist, and Miles Cranmer for assistance with PySR. This material is based upon work supported by the National Science Foundation under Grant No. #2138938, as well as startup funds from the University of Maryland, Baltimore County.

ORCID iDs

Neil D Tran https://orcid.org/0000-0001-9033-1759
Samiha Sharlin https://orcid.org/0000-0002-6379-9206
Tyler R Josephson https://orcid.org/0000-0002-0100-0227

References

[1] Koza J R 1992 *Genetic Programming* (The MIT Press)
[2] Oviedo F, Lavista Ferres J L, Buonassisi T and Butler K T 2022 Interpretable and explainable machine learning for materials science and chemistry *Acc. Mater. Res.* **3** 597–607
[3] Zhong X, Gallagher B, Liu S, Kailkhura B, Hiszpanski A and Han T 2022 Explainable machine learning in materials science *npj Comput. Mater.* **8** 1–19
[4] Esterhuizen J A, Goldsmith B R and Linic S 2022 Interpretable machine learning for knowledge generation in heterogeneous catalysis *Nat. Catal.* **5** 175–84
[5] Kordon A, Castillo F, Smits G and Kotanchek M 2006 Application issues of genetic programming in industry *Genetic Programming Theory and Practice III* (Springer) pp 241–58
[6] Savic D A, Walters G A and Davidson J W 1999 A genetic programming approach to rainfall-runoff modelling *Water Res. Manage.* **13** 219–31
[7] Schmidt M and Lipson H 2009 Distilling free-form natural laws from experimental data *Science* **324** 81–85
[8] Hernandez A, Balasubramanian A, Yuan F, Mason S and Mueller T 2019 Fast, accurate, and transferable many-body interatomic potentials by symbolic regression (arXiv:1904.01095 [cond-mat, physics:physics])
[9] Ansari M, Gandhi H A, Foster D G and White A D 2022 Iterative symbolic regression for learning transport equations *AIChE J.* **68** e17695
[10] Cranmer M, Sanchez Gonzalez A, Battaglia P, Xu R, Cranmer K, Spergel D and Ho S 2020 Discovering symbolic models from deep learning with inductive biases *Advances in Neural Information Processing Systems* vol 33 (Curran Associates, Inc.) pp 17429–42
[11] Ouyang R, Curtarolo S, Ahmetcik E, Scheffler M and Ghiringhelli L M 2018 SISSO: a compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates *Phys. Rev. Mater.* **2** 083802
[12] Chakraborty A, Sivaram A and Venkatasubramanian V 2021 AI-DARWIN: a first principles-based model discovery engine using machine learning *Comput. Chem. Eng.* **154** 107470
[13] Goldberg D E 1989 *Cenetic Algorithms in Search, Optimization, Machine Learning*
[14] Kronberger G, de França F O, Burlacu B, Haider C and Kommenda M 2022 Shape-constrained symbolic regression – improving extrapolation with prior knowledge *Evol. Comput.* **30** 75–98
[15] Haider C, De Franca F O, Burlacu B and Kronberger G 2023 Shape-constrained multi-objective genetic programming for symbolic regression *Appl. Soft Comput.* **132** 109855
[16] Tenachi W, Ibata R and Diakogiannis F I 2023 Deep symbolic regression for physics guided by units constraints: toward the automated discovery of physical laws (arXiv:2303.03192)
[17] Udrescu S-M and Tegmark M 2020 AI Feynman: a physics-inspired method for symbolic regression *Sci. Adv.* **6** eaay2631
[18] Simon Keren L S, Liberzon A and Lazebnik T 2023 A computational framework for physics-informed symbolic regression with straightforward integration of domain knowledge *Sci. Rep.* **13** 1249
[19] Lu Q, Ren J and Wang Z 2016 Using genetic programming with prior formula knowledge to solve symbolic regression problem *Comput. Intell. Neurosci.* **2016** 1–1
[20] Kubalík J, Derner E and Babuška R 2021 Multi-objective symbolic regression for physics-aware dynamic modeling *Expert Syst. Appl.* **182** 115210

[21] Medina J and White A D 2023 Active learning in symbolic regression performance with physical constraints (arXiv:2305.10379)
[22] Makarov D E and Metiu H 1998 Fitting potential-energy surfaces: a search in the function space by directed genetic programming *J. Chem. Phys.* **108** 590–8
[23] Akbarzadeh-T M R and Jamshidi M 1997 Incorporating a-priori expert knowledge in genetic algorithms *Proc. 1997 IEEE Int. Symp. on Computational Intelligence in Robotics and Automation CIRA'97.'Towards New Computational Principles for Robotics and Automation'* (IEEE) pp 300–5
[24] Schmidt M D and Lipson H 2009 Incorporating expert knowledge in evolutionary search: a study of seeding methods *Proc. 11th Annual Conference on Genetic and Evolutionary Computation* pp 1091–8
[25] Engle M R and Sahinidis N V 2022 Deterministic symbolic regression with derivative information: general methodology and application to equations of state *AIChE J.* **68** e17457
[26] Guimerá R, Reichardt I, Aguilar-Mogas A, Massucci F A, Miranda M, Pallarés J and Sales-Pardo M 2020 A Bayesian machine scientist to aid in the solution of challenging scientific problems *Sci. Adv.* **6** eaav6971
[27] Cornelio C, Dash S, Austel V, Josephson T R, Goncalves J, Clarkson K L, Megiddo N, El Khadir B and Horesh L 2023 Combining data and theory for derivable scientific discovery with AI-Descartes *Nat. Commun.* **14** 1777
[28] Ashok D, Scott J, Wetzel S J, Panju M and Ganesh V 2021 Logic guided genetic algorithms *Proc. AAAI Conf. on Artificial Intelligence* vol 35 pp 15753–4
[29] Ben-Mansour R, Habib M A, Bamidele O E, Basha M, Qasem N A A, Peedikakkal A, Laoui T and Ali M 2016 Carbon capture by physical adsorption: materials, experimental investigations and numerical modeling and simulations - a review *Appl. Energy* **161** 225–55
[30] Ritter J A and Ebner A D 2007 State of the art adsorption and membrane separation processes for hydrogen production in the chemical and petrochemical industries *Sep. Sci. Technol.* **42** 1123–93
[31] Stenzel M H 1993 Remove organics by activated carbon adsorption *Chem. Eng. Prog.* **89** 4
[32] Ruthven D 1984 *Principles of Adsorption and Adsorption Processes* (Wiley)
[33] Limousin G, Gaudet J-P, Charlet L, Stephanie Szenknect V B and Krimissa M 2007 Sorption isotherms: a review on physical bases, modeling and measurement *Appl. Geochem.* **22** 249–75
[34] Yuen Foo K and Hameed B H 2010 Insights into the modeling of adsorption isotherm systems *Chem. Eng. J.* **156** 2–10
[35] Ayawei N, Newton Ebelegi A and Wankasi D 2017 Modelling and interpretation of adsorption isotherms *J. Chem.* **2017** 1–11
[36] Wang J and Guo X 2020 Adsorption isotherm models: classification, physical meaning, application and solving method *Chemosphere* **258** 127279
[37] Freundlich H 1906 *Über die Adsorption in LöSungen. Habilitationsschrift Durch Welche... zu Haltenden Probevorlesung" Kapillarchemie und Physiologie" Einladet* (W. Engelmann)
[38] Langmuir I 1918 The adsorption of gases on plane surfaces of glass, mica and platinum *J. Am. Chem. Soc.* **40** 1361–403
[39] Brunauer S, Emmett P H and Teller E 1938 Adsorption of gases in multimolecular layers *J. Am. Chem. Soc.* **60** 309–19
[40] Sips R 1948 On the structure of a catalyst surface *J. Chem. Phys.* **16** 490–5
[41] Talu O and Myers A L 1988 Rigorous thermodynamic treatment of gas adsorption *AIChE J.* **34** 1887–93
[42] Toth J 1997 Some consequences of the application of incorrect gas/solid adsorption isotherm equations *J. Colloid Interface Sci.* **185** 228–35
[43] Cranmer M, Ashok D and Brehmer J 2021 MilesCranmer/PySR: v0.6.0
[44] Konfrst Z 2004 Parallel genetic algorithms: advances, computing trends, applications and perspectives *18th Int. Parallel and Distributed Processing Symp., 2004. Proc.* p 162
[45] Meurer A *et al* 2017 Sympy: symbolic computing in python *PeerJ Comput. Sci.* **3** e103
[46] Cranmer M 2023 Interpretable machine learning for science with PySR and symbolic regression.jl (arXiv:2305.01582 [astro-ph, physics:physics])
[47] Vlugt T J H, Zhu W, Kapteijn F, Moulijn J A, Smit B and Krishna R 1998 Adsorption of linear and branched alkanes in the zeolite silicalite-1 *J. Am. Chem. Soc.* **120** 5599–600
[48] Vlugt T J H, Krishna R and Smit B 1999 Molecular simulations of adsorption isotherms for linear and branched alkanes and their mixtures in silicalite *J. Phys. Chem.* B **103** 1102–18
[49] Strathern M 1997 improving ratings': audit in the British university system *Eur. Rev.* **5** 305–21
[50] Richardson D and Fitch J 1994 The identity problem for elementary functions and constants *Proc. Int. Symp. on Symbolic and Algebraic Computation, ISSAC'94* (Association for Computing Machinery) pp 285–90
[51] D'Amour A *et al* 2020 Underspecification presents challenges for credibility in modern machine learning (arXiv:2011.03395 [cs, stat])