Received: 30 July 2021 | Revised: 31 January 2022 | Accepted: 16 March 2022

DOI: 10.1002/aic.17695

RESEARCH ARTICLE

Transport Phenomena and Fluid Mechanics

AIChE JOURNAL

Iterative symbolic regression for learning transport equations

Mehrad Ansari | Heta A. Gandhi | David G. Foster | Andrew D. White

Department of Chemical Engineering, University of Rochester, Rochester, New York, USA

Correspondence
Andrew D. White, Department of Chemical Engineering, University of Rochester, Rochester, NY 14627, USA.
Email: andrew.white@rochester.edu

Funding information
National Institute of General Medical Sciences, Grant/Award Number: R35GM137966; National Science Foundation, Grant/Award Number: 1751471; National Science Foundation, Grant/Award Number: 1547580

Abstract

Computational fluid dynamics (CFD) analysis is widely used in chemical engineering. Although CFD calculations are accurate, the computational cost associated with complex systems makes it difficult to obtain empirical equations between system variables. Here, we combine active learning (AL) and symbolic regression (SR) to get a symbolic equation for system variables from CFD simulations. Gaussian process regression-based AL allows for automated selection of variables by selecting the most instructive points from the available range of possible parameters. The results from these experiments are then passed to SR to find empirical symbolic equations for CFD models. This approach is scalable and applicable for any desired number of CFD design parameters. To demonstrate the effectiveness, we use this method with two model systems. We recover an empirical equation for the pressure drop in a bent pipe and a new equation for predicting backflow in a heart valve under aortic insufficiency.

KEYWORDS
artificial intelligence, computational fluid dynamics, fluid mechanics

1 | INTRODUCTION

Computational fluid dynamics (CFD) provides a numerical approximation to the conservation of mass, momentum and energy (Navier–Stokes equations) that govern the fluid flow behavior. Although experimentally quantifying fundamental mechanisms of a process is often insightful, CFD modeling can provide an alternative approach for better understanding of the underlying physics in a less resource-intensive manner.[1] With continued growth of computational power and advances in CFD techniques, even complex models can be done on commodity hardware. Thus, CFD modeling is routinely applied in several fields of science and engineering such as chemistry,[2,3] materials,[4] fluid flow and heat transfer,[5,6] biology,[7] drug delivery,[8] semiconductors,[9] environmental engineering,[10,11] biomedical engineering,[12] and aeronautics.[13]

Parametric analysis has been widely used in process modeling and design optimization using a semi-automated or fully-automated workflows.[14–16] However, running the entire CFD calculations on all possible design parameters can be computationally expensive, especially for complex CFD models. Thus, there is a need to identify which feature points are most important in experiment design, and allow for system analysis with fewer CFD simulations. Another challenge in the mentioned settings is the lack of quantitative general equations that can be applied to different systems. This includes different geometrical designs, as well as having different operating conditions such as temperature, pressure, velocity or fluid properties. These system variables that can be inputs to CFD models are referred to as feature points in this work.

Here, we apply active learning (AL) to CFD modeling experiment design and then use symbolic regression (SR) to find empirical symbolic equations for these CFD models. AL is an iterative supervised learning technique that attempts to learn a good model from a few data points, by allowing the model to pick which data points it trains from.[17] In other words, AL is the process of choosing the next experiment or feature point optimally using less resources and adding this new point to the training data. In this article, we use AL with Gaussian process regression (GPR) to choose the next optimal CFD feature points. GPR is often associated with Bayesian optimization, where the goal is to optimize an expensive black box function. However, our goal is to find a symbolic equation across all feature values rather than a single optimum with as few CFD simulations as possible.

Mehrad Ansari and Heta A. Gandhi equally contributed to this study.

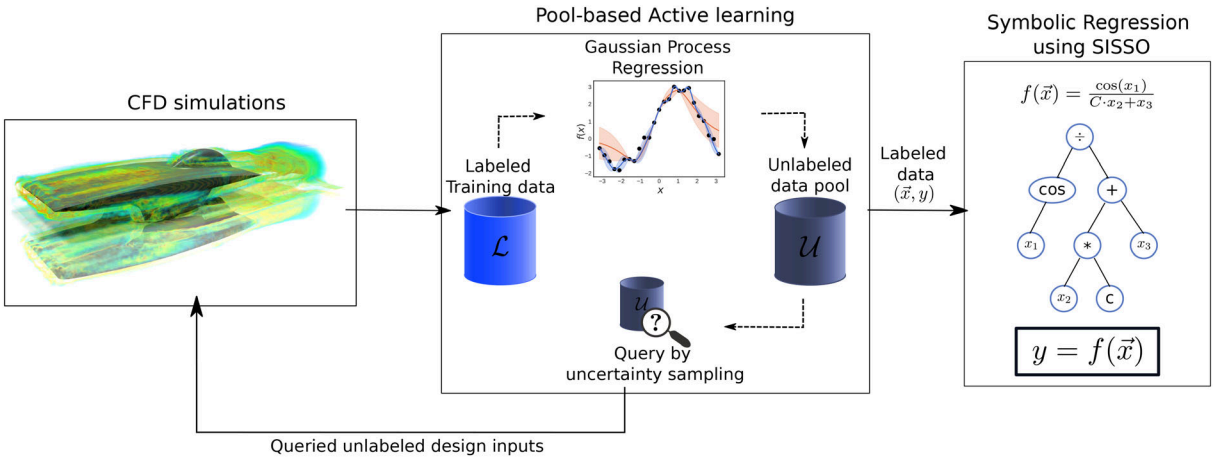

FIGURE 1 Overview of the workflow. A fully automated parameterized computational fluid dynamics (CFD) model is coupled with pool-based active learning (AL) and symbolic regression (SR). The CFD model is used to generate the labeled training data. The AL model is used to predict the next optimal feature point for CFD simulations and obtain a balance between exploration and exploitation of the parametric space. An iteration of AL consists of learning the available data using Gaussian process regression and then using uncertainty sampling to find the next feature point to label using a CFD simulation. The labeled data is then used as training data for SR to find the empirical symbolic equation for CFD feature inputs and outputs. SISSO, sure-independence screening and sparsifying operator

SR is a machine learning approach used to systematically determine symbolic equations that fit certain data with an unknown underlying function.[18–20] Unlike regression, where data is fit to a pre-defined function, SR attempts to find both the model and model parameters simultaneously. Neural networks (NN) are a popular choice for learning from data. However, even though they can approximate any function, the output from NN is difficult to interpret and cannot be converted to an analytical function. SR gives interpretable symbolic equations from data, which makes this approach appealing. Here, "interpretable" means that the exact relationship between input features and outputs is known in equation form. There are limited studies that explore SR to find general relationships from CFD simulations.[21,22] In this study, we demonstrate the use of AL for design of CFD experiments, and then apply the SISSO SR method to determine the physics of fluid systems. Figure 1 provides an overview of this method. A fully-automated workflow is combined with AL to generate CFD data, which is then used to get an empirical symbolic equation using SR. To avoid non-physical symbolic equations, we include known asymptotic points using prior understanding of physics of the fluid systems being studied. These asymptotic points are included in the SR training data. This forces SR to return equations have the correct asymptotic behavior at extreme geometries and velocities. The AL and SR methods are described in detail in sections that follow.

2 | COMPARISON TO RELATED WORK

Previous studies have implemented AL to accelerate simulation-driven design optimization. Owoyele et al.[23] used AL to perform simulation-based data generation, ML learning and surrogate optimization to refine solution in the vicinity of predicted optimum parameters for design of a compression ignition engine. Gonçalves et al.[24] studied the generation of simulation-based surrogate models with the task of parameter domain exploration using various sampling and regression-based AL strategies. In a similar study, Pan et al.[25] used AL for developing surrogate models for industrial fluid flow case studies under a constraint of a limited function evaluations. AL has also been implemented in specific experiment design to deploy efficient design space exploration to enhance model quality.[26,27] Over the past few decades, multiple methods to solve the SR problem have been developed. Traditional deterministic algorithms assume a predefined mathematical function and attempt to find parameters with the best fit to the data, whereas, evolutionary algorithms try to find parameters and learn the best-fit function, simultaneously. Some prevalent methods are genetic programming algorithms,[28–34] sparse regression,[20,35–37] pareto-optimal regression,[38,39] and the sure-independence screening and sparsifying operator (SISSO) method.[40,41] Most SR frameworks implement the popular Genetic programming,[42] which is an improved version of Genetic Algorithms (GA),[43,44] inspired by Darwin's theory of natural selection. Genetic programming has also been used to identify hidden physical laws from the input–output response prior.[19,45,46] We use SISSO because it has been shown to be robust with small amounts of data.[40,47,48] This is advantageous for analysis of CFD systems where the computational cost of simulations increases with increasing number of variables and complexity of the system. Compared to existing work, our approach is novel in the sense of combining AL and SR to optimize training efficiency and output a general equation for any fluid system of interest.

3 | THEORY

3.1 | Governing equations

The governing equations are the conservation of mass, momentum and energy. With the assumption of steady state, incompressible flow and constant temperature, we have the continuity equation:

$$\nabla.\vec{v}=0 \tag{1}$$

and the momentum equation can be simplified to

$$\rho\left(\vec{v}.\nabla\right)\vec{v}=-\nabla P+\mu\nabla^2\vec{v}, \tag{2}$$

where, $\vec{v}$ is the velocity vector, P is pressure and ρ and μ denote the fluid density and viscosity, respectively. A Dirichlet boundary condition is imposed at the inlet, with a parabolic velocity profile normal to the boundary for all cases. This assumption allows the analysis for a fully-developed flow without the need for unnecessary geometry extensions, which results in extra mesh elements. The no-slip boundary condition imposed at the wall ensures a zero velocity relative to the pipe surface. Given the unknown pressure at the outlet, the outflow boundary condition is imposed. A convergence criterion is defined based on the conservation of mass at the inlet and outlet boundaries.

3.1.1 | Pressure drop in a bent pipe

The laminar fluid flow in circular pipes is a classical problem in fluid mechanics, and it has been analyzed by the means of momentum balance, resulting in the famous Hagen-Poiseuille (HP) equation[49]:

$$w=\frac{\pi(P_0-P)d^4\rho}{32\mu L}, \tag{3}$$

where the mass flow rate w is the product of cross-sectional area, density, and average velocity $\langle v\rangle$. Here d and L are the pipe diameter and length, respectively, and P_0 and P denote the pressure at the inlet and outlet of the pipe. Note that Equation (3) is only valid for continuous, laminar, incompressible, steady, Newtonian flow that is fully developed. Given the pipe dimensions, fluid properties and average inlet velocity, one can easily obtain the pressure drop using Equation (3). Our goal here is to find an empirical equation for the pressure drop in a bent circular pipe as a function of the average inlet velocity ($\langle v\rangle$), pipe diameter (d), and bend angle (θ). The fluid is considered to be water at 25°C with constant properties. This setting is implemented to limit the number of feature points to three. However, more complicated models involving chemical reactions and convective heat transfer can be analyzed with more feature points. The geometry has been parameterized and meshed with hexahedral elements, as represented in Figure 2. More details on model parameterization can be found in Section S1.1.

Using Equation (3), the model is validated based on five different inputs with a bend angle of 1°, which is approximately equivalent to a straight pipe. The mean error for the unit length pressure drop with respect to the HP equation is about 2%.

3.1.2 | Backflow at an expansion joint

We can consider the human heart to operate as two pumps in series. The right heart pumps blood to the pulmonic circulation and the left heart to the systemic circulation.[50] The valves in the human heart open and close efficiently, allowing the blood flow in the forward direction and minimizing the regurgitation of flood to the chamber it came from. Aortic insufficiency is a condition in which the heart valve fails to tightly close, allowing blood to flow backwards into the heart instead of pumping out.[51] We have considered an expansion joint to simulate this condition in a simplified geometry and quantify the backflow volume. The fluid is blood with constant properties with density and viscosity set to 1060 kg/m^3 and 0.004 Pa s at 37°C. A fully-developed flow is defined at the inlet boundary, no-slip velocity at wall and outflow boundary condition at the outlet. The inlet pressure is set to 120 mmHg absolute. Once again, the geometry and the hexahedral mesh are constrained to avoid invalid models given different inputs (Figure 3). Blood dominantly flows in the z direction, thus the z component of the velocity is used as our metric for defining the backflow. The backflow volume is calculated by summing over the volume of mesh with negative velocity in z direction (Section S1.2). The inputs to the model in this setting are the average inlet velocity ($\langle v\rangle$), inlet diameter (d), and expansion angle (θ). The model outputs the percentage backflow (f) by finding the ratio of V_{bf} to the total system volume.

3.2 | AL model

The goal of AL algorithms is to increase accuracy of a machine learning model, while minimizing the training data required to train the model. It is often formulated as an optimization problem.[17] Here, we use a pool-based AL setting. Pool-based AL assumes that the model has access to a large set of unlabeled samples. Consider a dataset $\mathcal{D}\supseteq\{\mathcal{L},\mathcal{U}\}$ comprised of a small set of labeled data $\mathcal{L}=\left\{\left(\vec{x}_i,y_i\right)\right\}_{i=1}^{n_{\mathcal{L}}}$ with features $\vec{x}_i$ (n-dimensional vector) and corresponding labels y_i, and a large pool of unlabeled data $\mathcal{U}=\left\{\vec{x}_j\right\}_{j=1}^{n_{\mathcal{U}}}$ containing only features $\vec{x}_j$. $n_{\mathcal{L}}$ and $n_{\mathcal{U}}$ are the number of samples in the labeled and unlabeled dataset, respectively, and $n_{\mathcal{L}}\ll n_{\mathcal{U}}$. A model Φ is initially trained using the labeled data $\mathcal{L}$. Next, an unlabeled data sample, called a query, $\vec{x}_{i+1}$ is selected from the unlabeled data pool $\mathcal{U}$ using a query strategy. The selected query is then labeled by using an "oracle." An oracle is a human expert, or an experiment, but in this work, it is a CFD simulation. The label y_{i+1} is found by conducting a CFD simulation for the flow conditions and geometry specified by $\vec{x}_{i+1}$. This new observation $\left(\vec{x}_{i+1},y_{i+1}\right)$ is added to labeled data $\mathcal{L}$ and the model Φ is retrained based on this updated value. The process of query-label-retrain is iteratively repeated until the labeling budget is exhausted.

In this study, we use GPR[52] as our predictive model Φ and uncertainty sampling[53] as the query strategy. The uncertainty sampling strategy selects the feature point whose prediction is most uncertain. The model has the least information in the vicinity of the most uncertain prediction, and it is relatively more confident about other predictions. Hence, labeling the most uncertain point is most informative for the model.[17,54] GPR is a probabilistic estimation method and has the ability to provide uncertainty measurement because it provides confidence intervals for predictions at each feature point. The goal of any

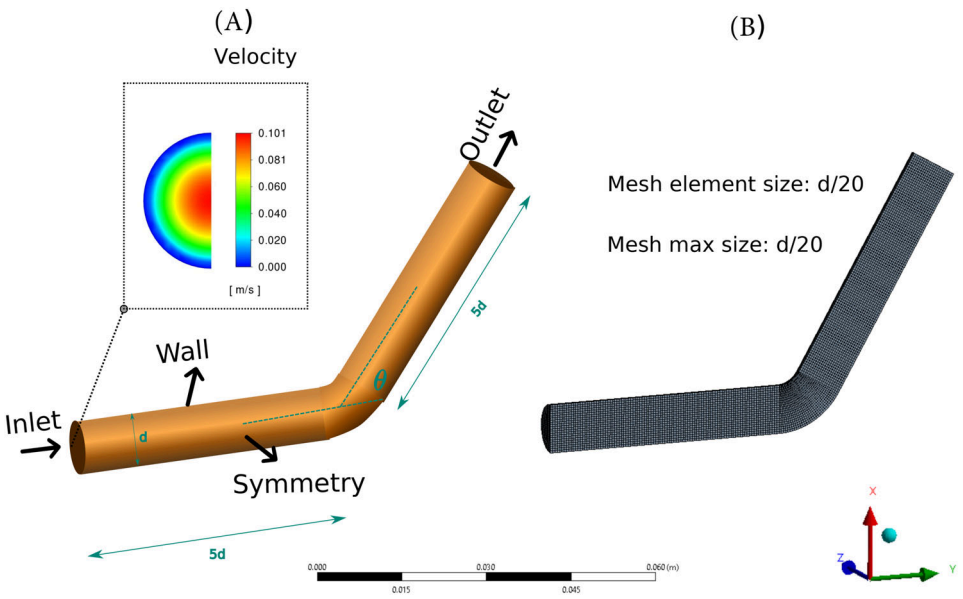

FIGURE 2 Bent pipe. (A) Parameterized geometry with inlet, wall, symmetry, and outflow boundary conditions. A parabolic velocity profile is defined normal to the inlet boundary to satisfy the assumption of fully-developed flow. The geometric constraints allow having different inputs for d and θ and ensure valid geometries. (B) Parameterized hexahedral mesh with element size and max element size of $d/20$, which allows for adjustable meshing given different geometric inputs for d and θ

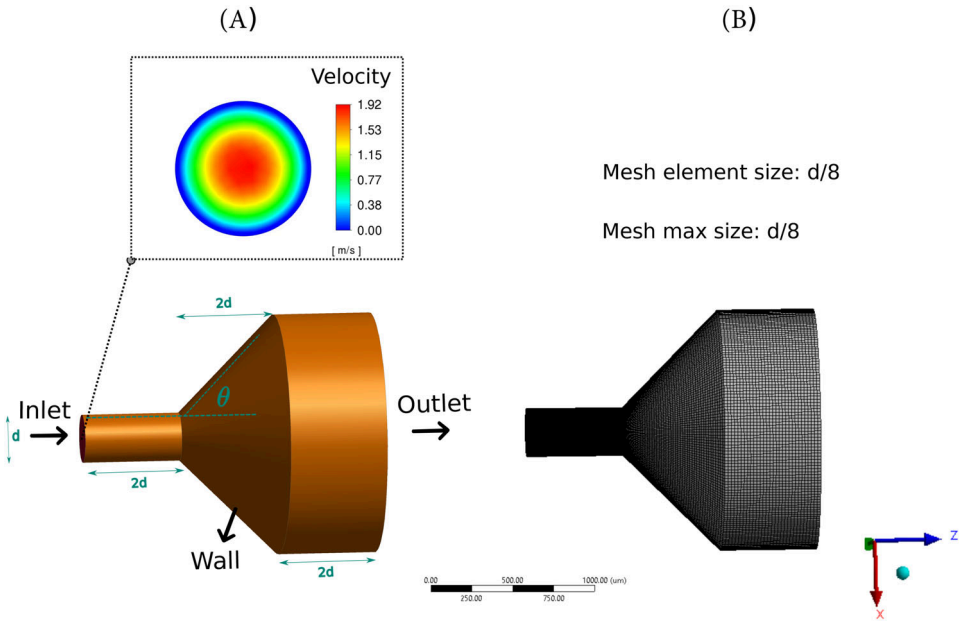

FIGURE 3 Expansion joint. (A) Parameterized geometry with inlet, wall, and outflow boundary conditions. A parabolic velocity profile is defined normal to the inlet boundary to satisfy the assumption of fully-developed flow. The geometric constraints allow having different inputs for d, θ and ensure valid geometries. (B) Parameterized hexahedral mesh with element size and max element size of $d/8$, which allows for adjustable meshing given different geometric inputs for d and θ

regression model is to fit a function to datapoints. There are infinitely many functions that can possible fit a set of points. GPR assigns a probability to each of these functions.[55] Once the GPR model Φ is trained, the uncertainty in predictions and next feature point choice are calculated using Equations (4) and (5):

$$U\left(\vec{x}\right) = \left[1 - \max\left(P_{\Phi}^{i}\left(y|\vec{x}\right)\right)\right] \tag{4}$$

$$\vec{x}_{i+1} = \underset{\vec{x}}{\operatorname{argmax}}\, U\left(\vec{x}\right). \tag{5}$$

To ensure the method is robust to label noise, $U\left(\vec{x}\right)$ is itself made a probability and $\vec{x}_{i+1}$ is sampled rather than computing the argmax. This is accomplished by computing the softmax of the predicted uncertainty, $U\left(\vec{x}\right)$.[56] Thus, our AL equation is

$$P\left(\vec{x}_{i+1}\right) = \operatorname{softmax}\left[U\left(\vec{x}\right)\right] = \frac{e^{U\left(\vec{x}\right)_j}}{\Sigma_{j=1}^{n_U} U\left(\vec{x}\right)_j} \tag{6}$$

Readers are referred to Section S2 for further details.

3.3 | SR model

To learn an interpretable model from CFD simulations, we use SISSO, developed by Ouyang et al.[40] SISSO aims to construct a symbolic

equation between primary features $\vec{x}$ and labels y. Given M samples, SISSO assumes that the labels can be expressed as a linear combination of non-linear functions of primary features. So, $y = f(\Psi)$ where $\Psi = \{\psi_1, \psi_2, ..., \psi_r\}$ is a set of secondary features. The secondary features ψ_i are non-linear, closed form functions of primary features. If $\vec{x} = [x_1, x_2, x_3, ..., x_n]$ are primary features, then examples of secondary features are $\{x_1/x_3,\ x_3 - x_1x_2,\ x_4x_5/x_1, ...\}$. These secondary features are obtained by recursively applying a set of user-defined operators on the primary features and creating a set of potential secondary features. The operator set can be any combination of unary and binary operators. The number of potential secondary features is proportional to the number of primary features used, the number of operators used, and the level of recursion. At each iteration, SISSO selects the subsets of secondary features that have the largest linear correlations with y. The number of terms in the linear expansion $f(\Psi)$ (called descriptors) are controlled by a sparsifying l_0 regularization. Note, here the number of descriptors refers to the number of terms in the output equation. For each iteration q, SISSO constructs multiple models for $f(\Psi)$ using the secondary features and selects the one with the largest correlation with the target property. More details on this procedure can be found in Ouyang et al.[40]

In SISSO, dimensional analysis is performed to retain only valid combinations of primary features. This ensures that secondary features do not have unphysical units (e.g., force + time). To achieve this, there is an option in SISSO to group primary features that have the same derived units. We modified this option so that the primary features are expressed in terms of fundamental units of measurement (mass, length, time, angle) and grouped based on these fundamental units instead of derived units.

4 | METHODS

In this section, the fully-automated CFD workflow, AL, and SR procedures are explained. As seen in Figure 1, we first use AL to get labeled data from the CFD model and then perform SR using that as training data, to obtain an empirical symbolic equation between features and labels.

Our fully-automated workflow has been used on two different fluid flow problems, as described in previous sections. These problems are not necessarily complex, and the main goal here is to demonstrate a more robust approach that can also be applied to complex problems. In this study, we have coupled ANSYS Workbench with python. The parameterized CFD models are developed by defining input feature points. These inputs can include geometric features, operating conditions, and fluid flow properties. They are easily adjustable in a python script and the outputs are updated accordingly.

For both the systems described preciously, we have three-dimensional (3D) features, meaning we vary are three input parameters for the CFD simulations. For the bent pipe system, these features

TABLE 1 Feature ranges and test data split for the two systems

	Features					
System	d (m)	θ (°)	$\langle v_{in} \rangle$ (m/s)	Labels	Test data	Data split (train:test counts)
Bent pipe	0.005 – 0.1	1 – 180	0.005 – 0.02	$\Delta P/L$	d > 0.07m and θ > 120°	3696:400
Expansion joint	0.0005 – 0.005		0.002 – 0.5	f	d > 0.0025m and θ > 50°	1764:589

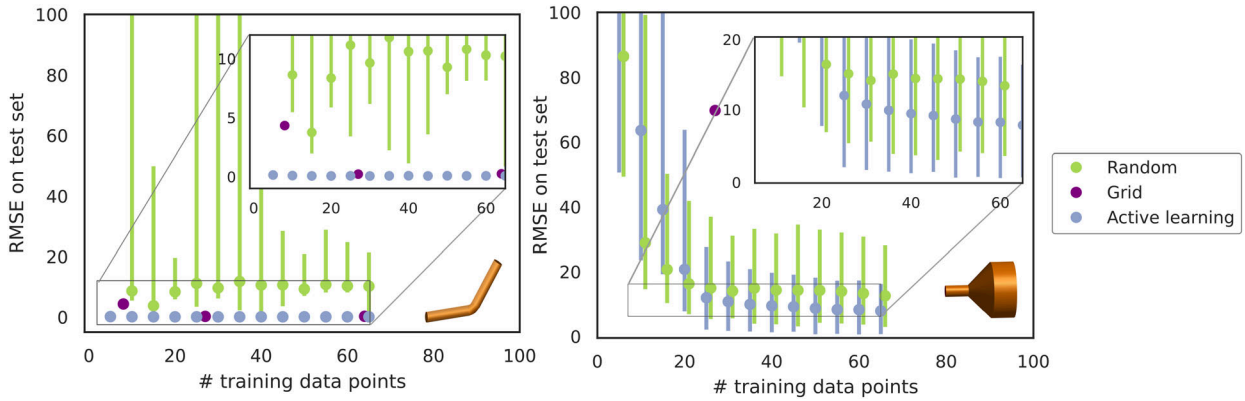

FIGURE 4 Root mean squared errors (RMSE) as a function of number of training data points. The SR equations obtained from different models for varying number of training data points are evaluated for test data, and the RMSE in predictions is plotted. Error bars on these points indicate the 50th quantile for RMSEs on test data. It is observed that active learning (AL) model for experiment design has the best performance followed by grid search for the bent pipe, and random selection of experiment points for the expansion joint. RMSE distributions for AL and random selection are significantly different, as shown by an independent sample t-test with $p = 8.8 \times 10^{-23}$ and $p = 0.0125$, respectively, for bent pipe and expansion joint systems

TABLE 2 Equations obtained from SISSO for the bent pipe system

Method	Training points	Mean test RMSE	Equation
AL	5	0.143	$\frac{C_1 v + d(C_2 v \sin(\theta) + C_3 \theta \cos(\theta))}{d^2}$
Random	5	349.42	$\frac{C_1 \cos(d)}{d^2} + \frac{C_2 e^{-v}}{d^2} + \frac{C_3 \theta v^2}{d}$
Grid	8	4.32	$\frac{C_1 v}{d^2} + \frac{C_2 \theta^2 v}{d} - C_3 e^{v} + C_3 e^{-\theta}$
AL	60	0.072	$\frac{v(C_1 + C_2\theta + C_3 d\theta v)}{d^2}$
Random	60	10.29	$\frac{v(C_1 + C_2\theta + C_3 d\theta v)}{d^2}$
Grid	64	0.246	$\frac{C_1 v + C_2 d\theta v^2 + C_3 \theta^2}{d^2}$
Full set	3696	0.085	$\frac{v(C_1 + C_2\theta + C_3 d\theta v)}{d^2}$

Abbreviations: AL, active learning; RMSE, root mean squared errors; SISSO, sure-independence screening and sparsifying operator.

TABLE 3 Equations obtained from SISSO for the expansion joint

Method	Training points	Mean test RMSE	Equation
Random	5	86.42	$C_1 v \sin(\theta) + C_2 \theta v e^{d} + C_3 d e^{d}$
AL	5	107.13	$C_1 \sin(\theta) + \frac{C_2 \theta}{v \sin(d)} + C_3 \sin\left(\frac{\theta}{d}\right)$
Grid	8	4955.92	$\theta\left(C_1 v \sin(\theta) + C_2 d\theta v + \frac{C_3 d}{\cos(d)}\right)$
AL	60	8.25	$\frac{\theta(C_1 dv \cos(\theta) + C_2 + C_3\theta)}{dv}$
Random	60	13.30	$C_1 e^{-d\theta} + C_2 e^{\theta} + C_2 e^{-\theta} + C_3 dv e^{-d}$
Grid	64	79.35	$C_1 d\theta e^{-d} + C_2 d^3 v + C_3 \sin(\theta)$
Full set	1764	12.65	$\frac{C_1 \theta dv \cos(\theta) + C_2 dv \sin(d\theta) + C_3}{dv}$

Abbreviations: AL, active learning; RMSE, root mean squared errors; SISSO, sure-independence screening and sparsifying operator.

are pipe diameter (d), bend angle (θ), and average inlet velocity ($\langle v_{in} \rangle$). For the expansion joint, features are inlet pipe diameter (d), expansion angle (θ), and average inlet velocity ($\langle v_{in} \rangle$). The target property y for bent pipe is pressure drop $\Delta P/L$ and for the expansion joint, it is the backflow volume percentage f. To start with, a range of acceptable values for the input parameters is chosen to ensure laminar fluid flow and the 3D feature space is divided into training data and testing data. The ranges for features and test/train split criteria are shown in Table 1. Since we are not tuning hyperparameters, we do not use a validation split. To make sure that physics of the system are obeyed, we define asymptotes based on prior system knowledge. In the bent pipe system, $d = 0$ will result infinite pressure drop ($\lim_{d \to 0} \frac{\Delta P}{L}(d, \theta, v) = \infty$), which is a valid asymptote as per the HP equation (Equation 3). Parameter θ is set to be bounded between 15° and 60° in the expansion joint system. The rationale behind this choice comes from the fact that no backflow is formed at smaller expansion angles, which results in zero variance in the labels for all possible variations of the other two features ($f(d, \theta, v)|_{\theta < 15^\circ} = 0; \forall d, v$). On the other hand, at larger angles beyond this limit, the size of the system increases significantly as a result of geometrical constraint set for the length of the expansion section (Figure 3). Specifically, for the case of $\theta = 90^\circ$, the system size becomes infinite as a result of having an infinite outlet diameter.

For AL, three random points $\vec{x}$ from the training data are sampled, and CFD simulations are generated to find corresponding labels y. This is our initial training data ($\mathscr{L}$) for our pool-based AL and the rest of the feature points form the unlabeled data pool $\mathcal{U}$. CFD simulations are used to label data for feature points obtained from Equation (6). After N such queries, feature-label pairs in $\mathscr{L}$ are used as training data for the SISSO algorithm. We then add asymptotic points to this training data. The number of asymptotic points added depends on the number of training data points N in $\mathscr{L}$. We add greater of 3 or 10% of N points to $\mathscr{L}$ and make sure that all asymptotic conditions are represented. To create asymptotic data points, the primary features apart from the variable for which the asymptote is defined are sampled randomly from the regime defined for that feature. So, for the bent pipe system, when $\lim_{d \to 0} \frac{\Delta P}{L}(d, \theta, v) = \infty$, θ and $\langle v_{in} \rangle$ are randomly sampled from the bounds defined in Table 1. Density and viscosity are also added as features for SISSO. We use the operator set $\left\{+, -, \times, \div, \exp, -\exp, ()^{-1}, ()^{2}, \sin, \cos\right\}$ with our features for both systems and set the number of descriptors to 3. The symbolic equation obtained from SISSO is used to predict labels for the test features. This method is compared against random search and grid search experiment design algorithms. Random search is random selection of feature points with uniform sampling probability from the data pool. Grid search is when a hypercube of points from the 3D feature grid are selected. Grid is equivalent to a factorial design if we view our levels as discretization of our features. SISSO is used to find equations for both these methods so they can be compared against our AL

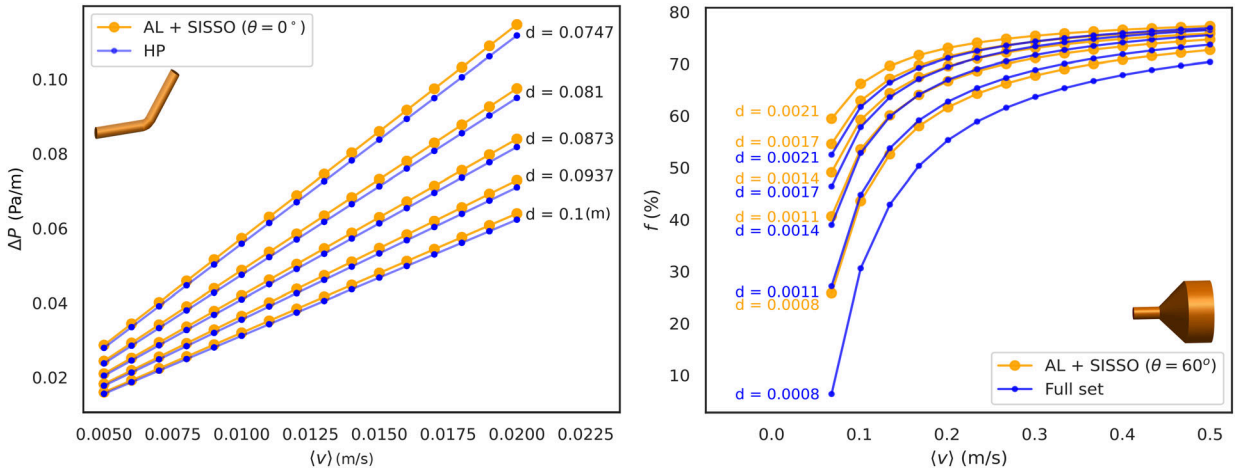

FIGURE 5 Comparison of active learning (AL) + sure-independence screening and sparsifying operator (SISSO) results with baseline models. The baseline model for bent pipe is the Hagen-Poiseuille (HP) equation, which gives $\Delta P/L$ for a laminar fluid in a straight pipe (i.e., $\theta = 0$). AL + SISSO equations show perfect agreement with the HP equation for different geometries. The baseline model for expansion joint is derived by performing SISSO on the entire feature pool. AL + SISSO equations mimic the baseline model's behavior exactly, with an offset along the y axis

+ SISSO method. The difference in these methods is reported using significance statistics obtained from an independent samples *t*-test. The independent samples *t*-test is a parametric test that compares the means of two independent distributions and gives statistics to confirm the hypothesis that the two populations are significantly different.[57]

5 | RESULTS AND DISCUSSION

The method described above, AL + SISSO, is tested and compared with baseline methods used for experiment design like random search and grid search. The objective was to obtain a symbolic relationship between inputs/features and outputs/labels, given data.

In Figure 4, we compare the root mean squared error (RMSE) on test data from SR equations for AL, random search and grid search. The symbolic relationships obtained from SISSO for AL, random search and grid search are evaluated on the test data points and respective RMSEs are calculated between these values and actual CFD values. Test data were withheld from the training data pool for both systems, as shown in Table 1. In Figure 4, each data point for AL and random search represents the mean value for RMSE from 100 independent iterations of training data sampling followed by SISSO. The uncertainty in AL comes from the randomly sampled initial training points and is in the coefficients C_i of SISSO equations. For bent pipe, we observe that AL converges quickly and requires fewer training points to get an accurate symbolic equation between features and labels. RMSEs for random search and AL are significantly different ($p = 8.8 \times 10^{-23}$, via independent samples *t*-test). Grid search outperforms random search and requires 27 training points to converge. Table 2 shows some equations obtained from SISSO for AL, random search, grid search, and the full training set. The complete list of equations obtained for different methods, as a function of training data points, is reported in Table S1. The equations reported are those that are observed maximum times (mode of the distribution) for a given value of training points. AL combined with SISSO provides an accurate equation with as low as 10 training data points, and the general form of the equation remains the same with increasing training data points. Random search requires 15 training points to obtain a similar equation. The equation obtained for grid search varies with increasing training points. Although, random search obtains the correct general equation with few points, the variance in the coefficients for these equations is high and hence, have a high RMSE in their approximation of pressure drop in the system. For the expansion joint, AL outperforms random search and grid search. The difference in RMSE between AL and random search increases as the number of training points increases ($p = 0.0125$, via independent samples *t*-test). Equations obtained from SISSO for this system are reported in Table 3 and the complete list as a function of training data points can be found in Table S2. The general equation for AL, random search and grid search remains the same after 30, 60, and 512 training points, respectively.

We also compare the performance of AL + SISSO models with baseline models. Figure 5 shows how AL + SISSO compares to the respective baseline models. At $\theta = 0^\circ$, the bent pipe becomes a straight pipe, for which $\Delta P/L$ can be calculated using the HP equation (Equation 3) and is considered the baseline. AL + SISSO equation results fit the HP equation. For backflow in the expansion joint, there is no such theory-derived equation to compare against. So, we perform SISSO on the entire feature pool and consider that as the baseline model. AL + SISSO predictions for $\theta = 60^\circ$ underestimate the backflow percentage compared to the baseline. However, the curves for AL + SISSO follow the same form as the baseline and there is an offset along the y axis. SISSO equations obtained for AL and the baseline are the same, and the difference in predicted labels comes from the coefficients C1, C2, C3 for the two equations. This is reasonable since our goal is to find a symbolic equation to understand the system and not to minimize the regression error.

In the final analysis, it is important to consider how well the results of AL + SISSO can be used to describe flow and how well they compare to known equations when they exist. In the case of the bent pipe, the data in Figure 5 confirm that the AL + SISSO matches nearly exactly with the HP Equation. Note that all symbolic equations beyond 10 training points shown in Table S1 are consistent with theory-driven HP equation (Equation 3) for $\theta = 0$ and accurately describe the pressure drop in the system. When known equations do not exist, which is the case for most complex flow scenarios, the ability of AL + SISSO to describe flow needs to be carefully interpreted and compared to best known approximations. For the expansion joint, there are no accepted equations for the backflow volume percentage to compare against for any geometry, so comparisons are made with the entire feature pool referred to as the full set in the figure. The data in Figure 5 do show a difference between the AL + SISSO and the full set. The important observation is that the shape of the graph for the volume percentage versus velocity are quite similar, rising and then leveling off with velocity as one would expect. The observed difference is the result of the coefficients in the equations as mentioned above and not an incorrect symbolic equation.

6 | CONCLUSIONS

We introduce an AL approach combined with SR for obtaining an empirical symbolic relationship between system variables for CFD simulations. This framework eliminates the need for the conventional trial-and-error or grid search methods for picking feature points since we let AL pick these points based on prior information available. We demonstrate the use of this method for two CFD systems and compare them against conventional methods. The results obtained from SISSO are more interpretable than those obtained from black box functions, and can be directly used. This method also greatly reduces the amount of data needed to get meaningful insights about a CFD system. One limitation of this method is that the obtained symbolic relationships are only valid for fluid flow regimes described by the feature domain considered for the training data pool (i.e., laminar flow regime for the two examples illustrated). Adding training data with asymptotic points from prior scientific knowledge helps ensure that the equations obey the physics of the system.

ACKNOWLEDGMENTS

We thank the Center for Integrated Research Computing (CIRC) at University of Rochester for providing computational resources and technical support. This material is based upon work supported by the National Science Foundation (NSF) under Grant 1751471, the Molecular Sciences Software Institute (MolSSI) under NSF grant 1547580, and the Maximizing Investigators' Research Award Grant R35 GM137966 by the National Institute of General Medical Sciences under the National Institutes of Health.

AUTHOR CONTRIBUTIONS

Mehrad Ansari: Data curation (equal); formal analysis (equal); investigation (equal); software (equal); validation (equal); visualization (equal); writing – original draft (equal); writing – review and editing (equal). **Heta A. Gandhi:** Data curation (equal); formal analysis (equal); investigation (equal); methodology (equal); software (equal); visualization (equal); writing – original draft (equal); writing – review and editing (equal). **David G. Foster:** Conceptualization (equal); methodology (equal); project administration (equal); software (equal); supervision (equal); validation (equal). **Andrew D. White:** Conceptualization (equal); funding acquisition (equal); project administration (equal); supervision (equal); writing – review and editing (equal).

DATA AVAILABILITY STATEMENT

All code and models are available at https://github.com/ur-whitelab/alcfd

ORCID

Mehrad Ansari https://orcid.org/0000-0001-5696-9193
Heta A. Gandhi https://orcid.org/0000-0002-9465-3840
David G. Foster https://orcid.org/0000-0003-1837-2112
Andrew D. White https://orcid.org/0000-0002-6647-3965

REFERENCES

1. Mohd Hafiz Zawawi, A Saleha, A Salwa, NH Hassan, Nazirul Mubin Zahari, Mohd Zakwan Ramli, and Zakaria Che Muda. A review: fundamentals of computational fluid dynamics (CFD). AIP Conference Proceedings, Vol. 2030, p. 020252. AIP Publishing LLC, 2018.
2. Börnhorst M, Kuntz C, Tischer S, Deutschmann O. Urea derived deposits in diesel exhaust gas after-treatment: integration of urea decomposition kinetics into a cfd simulation. *Chem Eng Sci*. 2020;211:115319.
3. Cai L, Pitsch H. Optimized chemical mechanism for combustion of gasoline surrogate fuels. *Combust Flame*. 2015;162(5):1623-1637.
4. MC Subin, Jason Savio Lourence, Ram Karthikeyan, and C Periasamy. Analysis of materials used for greenhouse roof covering-structure using cfd. IOP Conference Series: Materials Science and Engineering, Vol 346, p. 012068. IOP Publishing, 2018.
5. Mahian O, Kolsi L, Amani M, et al. Recent advances in modeling and simulation of nanofluid flows-part i: fundamentals and theory. *Phys Rep*. 2019;790:1-48.
6. Tong Z-X, He Y-L, Tao W-Q. A review of current progress in multiscale simulations for fluid flow and heat transfer problems: the frameworks, coupling techniques and future perspectives. *Int J Heat Mass Transf*. 2019;137:1263-1289.
7. Jayathilake PG, Li B, Zuliani P, Curtis T, Chen J. Modelling bacterial twitching in fluid flows: a cfd-dem approach. *Sci Rep*. 2019;9(1):1-10.
8. Koullapis P, Ollson B, Kassinos SC, Sznitman J. Multiscale in silico lung modeling strategies for aerosol inhalation therapy and drug delivery. *Curr Opin Biomed Eng*. 2019;11:130-136.
9. Zhang Y, Ding Y, Christofides PD. Multiscale computational fluid dynamics modeling of thermal atomic layer deposition with application to chamber design. *Chem Eng Res Des*. 2019;147:529-544.
10. Lauriks T, Longo R, Baetens D, et al. Application of improved cfd modeling for prediction and mitigation of traffic-related air pollution hotspots in a realistic urban street. *Atmos Environ*. 2021;246:118127.
11. Collivignarelli MC, Miino MC, Manenti S, et al. Identification and localization of hydrodynamic anomalies in a real wastewater treatment plant by an integrated approach: Rtd-cfd analysis. *Environ Process*. 2020;7(563):578.
12. Azriff A, Khader SMA, Pai R, Zubair M, Ahmad KA, Prakashini K. Haemodynamics study in subject-specific abdominal aorta with renal bifurcation using cfd-a case study. *J Adv Res Fluid Mech Thermal Sci*. 2018;50(2):118-121.

13. Jun H, Zheng S, Zhong M, et al. Recent development of a cfd-wind tunnel correlation study based on cae-avm investigation. *Chin J Aeronaut*. 2018;31(3):419-428.
14. John H Bucklow, Robin Fairey, and Mark R Gammon. An automated workflow for high quality cfd meshing using the 3D medial object. 23rd AIAA Computational Fluid Dynamics Conference, p. 3454, 2017.
15. Deininger ME, von der Grün M, Piepereit R, et al. A continuous, semi-automated workflow: from 3d city models with geometric optimization and cfd simulations to visualization of wind in an urban environment. *ISPRS Int J Geo Inf*. 2020;9(11):657.
16. Xiangyu G, Ciampa PD, Nagel B. An automated cfd analysis workflow in overall aircraft design applications. *CEAS Aeronaut J*. 2018;9(1):3-13.
17. Isabelle G, Gavin C, Gideon D, Vincent L & Alexander S Burr Settles. From theories to queries: active learning in practice. Active Learning and Experimental Design Workshop in Conjunction with AISTATS 2010, vol 16 of Proceedings of Machine Learning Research, pp. 1–18, Sardinia, Italy, 16 May 2011. JMLR Workshop and Conference Proceedings http://proceedings.mlr.press/v16/settles11a.html. Accessed: March 1, 2022
18. Voss H, Bünner MJ, Abel M. Identification of continuous, spatiotemporal systems. *Phys Rev E*. 1998;57(0):2820-2823. doi:10.1103/PhysRevE.57.2820
19. Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. *Science*. 2009;324(5923):81-85. doi:10.1126/science.1165893
20. Brunton SL, Proctor JL, Kutz JN. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc Natl Acad Sci*. 2016;113(15):3932-3937. doi:10.1073/pnas.1517384113
21. Neumann P, Cao L, Russo D, Vassiliadis VS, Lapkin AA. A new formulation for symbolic regression to identify physico-chemical laws from experimental data. *Chem Eng J*. 2020;387:123412. doi:10.1016/j.cej.2019.123412
22. Chakraborty A, Sivaram A, Samavedham L, Venkatasubramanian V. Mechanism discovery and model identification using genetic feature extraction and statistical testing. *Comput Chem Eng*. 2020;140:106900. doi:10.1016/j.compchemeng.2020.106900
23. Owoyele O, Pal P, Torreira AV. An automated machine learning-genetic algorithm framework with active learning for design optimization. *J Energy Res Technol Transact ASME*. 2021;143(8):082305-1-082305-10. doi:10.1115/1.4050489
24. Gonçalves GFN, Batchvarov A, Liu Y, et al. Data-driven surrogate modeling and benchmarking for process equipment. *Data Centric Eng*. 2020;1:e7. doi:10.1017/dce.2020.8
25. Indranil Pan, Gabriel Goncalves, Assen Batchvarov, Yuxin Liu, Yuyi Liu, Vikneswaran Sathasivam, Nicholas Yiakoumi, Lachlan Mason, and Omar Matar. Active learning methodologies for surrogate model development in CFD applications. In APS Division of Fluid Dynamics Meeting Abstracts, APS Meeting Abstracts, p S41.008, November 2019.
26. Deng H, Liu Y, Li P, Zhang S. Active learning for modeling and prediction of dynamical fluid processes. *Chemom Intell Lab Syst*. 2018;183:11-22. doi:10.1016/j.chemolab.2018.10.005
27. Vandermause J, Torrisi SB, Batzner S, et al. On-the-fly active learning of interpretable bayesian force fields for atomistic rare events. *Npj Comput Mater*. 2020;6(1):1-11.
28. Koza JR. Genetic programming as a means for programming computers by natural selection. *Stat Comput*. 1994;4:87-112. doi:10.1007/BF00175355
29. Ilknur Icke and Joshua C. Bongard. Improving genetic programming based symbolic regression using deterministic machine learning. 2013 IEEE Congress on Evolutionary Computation, pp. 1763–1770, 2013. doi: 10.1109/CEC.2013.6557774
30. Qiang L, Ren J, Wang Z. Using genetic programming with prior formula knowledge to solve symbolic regression problem. *Comput Intell Neurosci*. 2016;2016:1-17. doi:10.1155/2016/1021378
31. Wang Y, Wagner N, Rondinelli JM. Symbolic regression in materials science. *MRS Commun*. 2019;9:793-805. doi:10.1557/mrc.2019.85
32. El Hasadi YMF, Padding JT. Solving fluid flow problems using semi-supervised symbolic regression on sparse data. *AIP Adv*. 2019;9(11):115218.
33. Weatheritt J, Sandberg RD. Improved junction body flow modeling through data-driven symbolic regression. *J Ship Res*. 2019;63(04):283-293.
34. Androulakis IP, Venkatasubramanian V. A genetic algorithmic framework for process design and optimization. *Comput Chem Eng*. 1991;15(4):217-228.
35. Rudy SH, Brunton SL, Proctor JL, Kutz JN. Data-driven discovery of partial differential equations. *Sci Adv*. 2017;3(4):e1602614-1-e1602614-6. doi:10.1126/sciadv.1602614
36. Martin Schmelzer, Richard P Dwight, and Paola Cinnella. Discovery of algebraic Reynolds-stress models using sparse symbolic regression. *Flow Turb Combust*, 1040 (2):0 579–603, 2020.
37. Richa Ramesh Naik, Armi Tiihonen, Janak Thapa, Clio Batali, Zhe Liu, Shijing Sun & Tonio Buonassisi. Discovering equations that govern experimental materials stability under environmental stress using scientific machine learning. 2021. arxiv:2016.10951
38. Udrescu S-M, Tegmark M. Ai feynman: a physics-inspired method for symbolic regression. *Sci Adv*. 2020;6(16):eaay2631-1-eaay2631-16. doi:10.1126/sciadv.aay2631
39. Udrescu S-M, Tan A, Feng J, Neto O, Wu T, Tegmark M. Ai feynman 2.0: Pareto-optimal symbolic regression exploiting graph modularity. In: Larochelle H, Ranzato M, Hadsell R, Balcan MF, Lin H, eds. *Advances in Neural Information Processing Systems*. Vol 33. Curran Associates, Inc.; 2020:4860-4871.
40. Ouyang R, Curtarolo S, Ahmetcik E, Scheffler M, Ghiringhelli LM. Sisso: a compressed-sensing method for identifying the best low-dimensional descriptor in an immensity of offered candidates. *Phys Rev Mater*. 2018;2:083802. doi:10.1103/PhysRevMaterials.2.083802
41. Ouyang R, Ahmetcik E, Carbogno C, Scheffler M, Ghiringhelli LM. Simultaneous learning of several materials properties from incomplete databases with multi-task sisso. *J Phys Mater*. 2019;2:024002. doi:10.1088/2515-7639/ab077b
42. Koza JR, Koza JR. *Genetic Programming: on the Programming of Computers by Means of Natural Selection, Vol. 1*. MIT Press; 1992.
43. Mitchell M. *An Introduction to Genetic Algorithms*. MIT Press; 1998.
44. Holland JH et al. *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control, and Artificial Intelligence*. MIT Press; 1992.
45. Vaddireddy H, Rasheed A, Staples AE, San O. Feature engineering and symbolic regression methods for detecting hidden physics from sparse sensor observation data. *Phys Fluids*. 2020;32(1):015113.
46. Bongard J, Lipson H. Automated reverse engineering of nonlinear dynamical systems. *Proc Natl Acad Sci*. 2007;1040(24):9943-9948.
47. Xie SR, Kotlarz P, Hennig RG, Nino JC. Machine learning of octahedral tilting in oxide perovskites by symbolic classification with compressed sensing. *Comput Mater Sci*. 2020;180:109690.
48. De Breuck PP, Hautier G, Rignanese GM. Materials property prediction for limited datasets enabled by feature selection and joint learning with modnet. *Npj Comput Mater*. 2021;7:1-8. doi:10.1038/s41524-021-00552-2
49. Welty J, Rorrer GL, Foster DG. *Fundamentals of Momentum, Heat, and Mass Transfer*. John Wiley & Sons; 2020.
50. Chandran KB. Role of computational simulations in heart valve dynamics and design of valvular prostheses. *Cardiovasc Eng Technol*. 2010;10(1):18-38.

51. Prodromo J, D'Ancona G, Amaducci A, Pilato M. Aortic valve repair for aortic insufficiency: a review. *J Cardiothorac Vasc Anesth*. 2012; 26(5):923-932.
52. Rasmussen CE, Williams CKI. *Gaussian Processes for Machine Learning*. MIT Press; 2006.
53. Lewis DD, Catlett J. Heterogeneous uncertainty sampling for supervised learning. In: Cohen WW, Hirsh H, eds. *Machine Learning Proceedings*. Morgan Kaufmann; 1994:148-156. doi:10.1016/B978-1-55860-335-6.50026-X
54. Nguyen VL, Shaker MH, Hüllermeier E. How to measure uncertainty in uncertainty sampling for active learning. *Mach Learn*. 2021;6:1-34. doi:10.1007/S10994-021-06003-9/FIGURES/13
55. Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In 2007 IEEE 11th International Conference on Computer Vision, pp 1–8, 2007. doi: 10.1109/ICCV.2007.4408844
56. Goodfellow I, Bengio Y, Courville A. *Deep Learning*. MIT Press; 2016.
57. Kalpić D, Hlupić N, Lovrić M. *Student's t-Tests*. Springer Berlin Heidelberg; 2011:1559-1563. doi:10.1007/978-3-642-04898-2_641

SUPPORTING INFORMATION

Additional supporting information may be found in the online version of the article at the publisher's website.

How to cite this article: Ansari M, Gandhi HA, Foster DG, White AD. Iterative symbolic regression for learning transport equations. *AIChE J*. 2022;68(6):e17695. doi:10.1002/aic.17695