FIND: A Framework for Discovering Formulas in Data

Tingxiong Xiao[1], **Yuxiao Cheng**[1], **Jinli Suo**[1,2,*]

[1]Department of Automation, Tsinghua University, Beijing 100084, China
[2]Institute for Brain and Cognitive Science, Tsinghua University, Beijing 100084, China
jlsuo@tsinghua.edu.cn

Abstract

Scientific discovery serves as the cornerstone for advances in various fields, from the fundamental laws of physics to the intricate mechanisms of biology. However, two existing mainstream methods—symbolic regression and dimensional analysis, are significantly limited in this task: the former suffers from low computational efficiency due to the vast search space and often results in formulas without physical meaning; the latter provides a useful theoretical framework but also struggles in searching in a huge space because of lacking effective analysis for the latent variables. To address this issue, here we propose a framework for efficiently discovering underlying formulas in data, named FIND. We draw inspiration from Buckingham's Pi theorem, imposing dimensional constraints on the input and output, thereby ensuring discovered expressions possess physical meaning. Additionally, we propose a theoretical scheme for identifying the latent structure as well as a coarse-to-fine framework, significantly reducing the search space of latent variables. This framework not only improves computational efficiency but also enhances model interpretability. From comprehensive experimental validation, FIND showcases its potential to uncover meaningful scientific insights across various domains, providing a robust tool for advancing our understanding of unknown systems.

Introduction

Over the years, researchers have actively devoted large efforts in the field of scientific discovery (Bergen et al. 2019; Bhaskar and Nigam 1990; Camps-Valls et al. 2023; Xiao et al. 2024a; Bongard and Lipson 2007; Wang et al. 2023; Iten et al. 2020). In the quest for valuable domain knowledge and deeper field understanding, it is pivotal to explore the underlying formulas buried under data observations, which can assist in dimension reduction, prediction, or discovering the laws of nature.

There are two mainstream methods for formula discovery. Symbolic regression (SR) (Schmidt and Lipson 2009; Makke and Chawla 2024) is a machine learning technique that aims to automatically discover mathematical expressions to fit the given data, which can be used to discover physical laws, establish mathematical models, and predict unknown data. SR typically employs methods like genetic algorithms (Koza 1994) to explore the search space and find the best mathematical expression, and many researchers have begun to integrate deep learning with SR. Hernandez et al. develop a machine-learning algorithm based on SR in the form of genetic programming that is capable of discovering accurate, computationally efficient many-body potential models (Hernandez et al. 2019). Weng et al. use SR to guide the design of new oxide perovskite catalysts with improved oxygen evolution reaction activities (Weng et al. 2020). Cranmer et al. adopt sparse latent representations when training a GNN in a supervised setting, and then apply SR to components of the learned model to extract explicit physical relations (Cranmer et al. 2020). Kamienny et al. task a Transformer to directly predict the full mathematical expression, constants included and subsequently refined the predicted constants by feeding them to the non-convex optimizer as an informed initialization (Kamienny et al. 2022). Above SR methods can generate explicit expressions, which helps in understanding the underlying mechanisms of the observations. However, due to the typically large and complex search space of symbolic regression, the algorithms often suffer from limited efficiency and scalability, and most discovered formulas have no physical meaning.

Dimensional analysis (DA) aims to discover dimensionless equations with Buckingham's Pi theorem (Buckingham 1914). As is described, almost all physical laws can be expressed as dimensionless relationships with fewer dimensionless numbers and in a more compact form (Barenblatt 2003). Dimensionless numbers are power-law monomials of some physical quantities (Tan 2011), which can simplify a problem by reducing the number of variables. Xie et al. propose two-level optimization schemes with dimensional invariance to efficiently discover dimensionless numbers in static and dynamic systems (Xie et al. 2022). HiDeNN draws on the Pi theorem and designs a universal dimensionless learning AI framework to solve challenging computational science and engineering problems with little or no available physics as well as with extreme computation cost (Saha et al. 2021). Bakarji et al. develop three data-driven techniques that use the Buckingham Pi theorem as a constraint and show decent accuracy, robustness, and computational complexity (Bakarji et al. 2022). DHC-GEP effectively discovers function forms and coefficients using basic mathematical operators and physical variables, without preassumed candidate

functions, while the constraint of dimensional homogeneity filters out overfitting equations (Ma et al. 2024). DA has its theoretical system, but there is no effective analysis method for its latent variables, and its search space is still relatively large. Despite these progresses, DA-based methods are also slow and a high-efficiency discovery approach is highly demanded.

Here, we propose a framework, named FIND, for discovering formulas in data by designing a network structure including a latent layer and an expression layer. The latent layer is used to reduce dimensionality and discover meaningful input combinations, while the expression layer is used to find complete expressions. This scheme is inspired by Buckingham's Pi theorem, which imposes dimensional constraints on the input and output, and thereby ensures the discovered expressions possess physical meaning. Overall, the proposed framework can support both accuracy and efficiency. For reliable discovery, we analyze the relationship between network weights and data derivatives to estimate the number of latent variables, their connection relationships, and the weight ratios, providing theoretical guidance for scientific discovery. For high efficiency, we propose a coarse-to-fine (C2F) searching scheme to progressively depict the probability distribution map of the optimal solution, which significantly shortens the running time and reduces the likelihood of being trapped in local optima. The source code can be found at https://github.com/HarryPotterXTX/FIND.git.

The FIND Framework

Given a dataset $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} \in \mathbb{R}^{b\times p}$ is the input composed of variable $\mathbf{x} \in \mathbb{R}^p$, $\mathbf{Y} \in \mathbb{R}^{b\times 1}$ is the output composed of $\mathbf{y} \in \mathbb{R}$, and both $\mathbf{x}$ and $\mathbf{y}$ have units. We assume there exists a mapping $\mathbf{y} = f(\mathbf{x})$ from $\mathbf{X}$ to $\mathbf{Y}$ and designed FIND to discover this underlying relationship from data observations, with the scheme shown in Fig. 1.

Explainable Structure

Assuming that underlying natural laws can be represented by a concise and elegant equation, here we propose to decompose $f(\mathbf{x})$ into two parts—a latent layer $\mathbf{z} = f_1(\mathbf{x}) \in \mathbb{R}^s$ and an expression layer $\mathbf{y} = f_2(\mathbf{z})$, as shown in Fig. 1a.

For the latent layer, drawing inspiration from Buckingham's Pi theorem (Buckingham 1914), we set

$$\mathbf{z}_i = \prod_{j=1}^{p} \mathbf{x}_j^{\mathbf{W}_{ij}}, i = 1, \ldots, s, \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{s\times p}$ is the power matrix. The latent layer $f_1(\cdot)$ transforms the input $\mathbf{x}$ into $\mathbf{z}$, achieving dimensionality reduction and meaningful combinations of inputs.

Regarding the expression layer, as is well known, most functions or even deep neural networks can be expanded into a Taylor series (Xiao et al. 2024b), allowing us to approximate them using polynomials. Therefore, for $f_2(\cdot)$, we adopted a polynomial form.

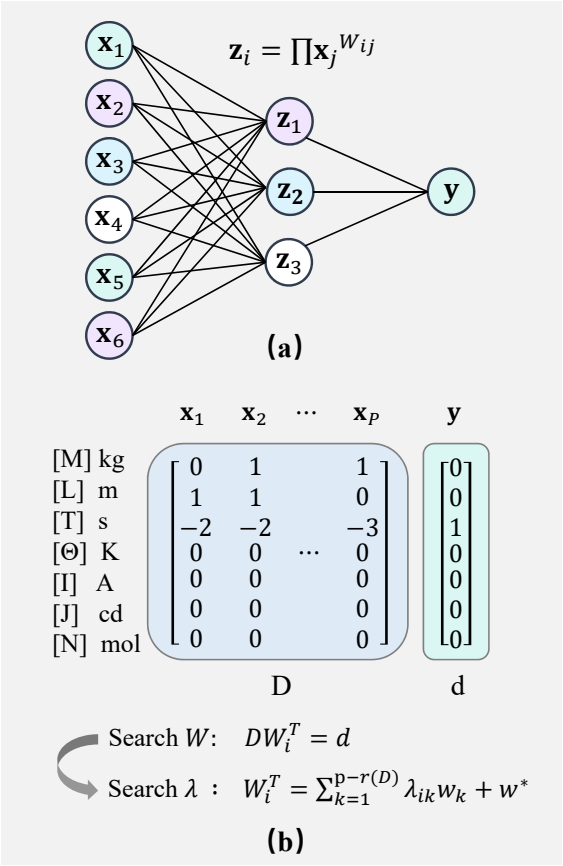

Figure 1: The scheme of the FIND framework. (a) The network structure that consists of a latent layer for dimensionality reduction and discovering meaningful input combinations, and an expression layer in polynomial form. (b) Search in the latent space with dimensional invariance.

Structure Identification

From Eq. (1),

$$\frac{\partial \mathbf{z}_i}{\partial \mathbf{x}_j} = \mathbf{W}_{ij}\mathbf{x}_j^{\mathbf{W}_{ij}-1}\prod_{k\neq j}\mathbf{x}_k^{\mathbf{W}_{ik}} = \mathbf{W}_{ij}\frac{\mathbf{z}_i}{\mathbf{x}_j}, \tag{2}$$

and further we get

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_j} = \sum_{i=1}^{s}\frac{\partial \mathbf{z}_i}{\partial \mathbf{x}_j}\frac{\partial \mathbf{y}}{\partial \mathbf{z}_i} = \sum_{i=1}^{s}\mathbf{W}_{ij}\frac{\mathbf{z}_i}{\mathbf{x}_j}\frac{\partial \mathbf{y}}{\partial \mathbf{z}_i}. \tag{3}$$

If $\mathbf{x}_j$ and $\mathbf{x}_k$ only connect to $\mathbf{z}_i$, we can get

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}_j}\Big/\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k} = (\mathbf{W}_{ij}\frac{\mathbf{z}_i}{\mathbf{x}_j}\frac{\partial \mathbf{y}}{\partial \mathbf{z}_i})/(\mathbf{W}_{ik}\frac{\mathbf{z}_i}{\mathbf{x}_k}\frac{\partial \mathbf{y}}{\partial \mathbf{z}_i}), \tag{4}$$

and further

$$\frac{\mathbf{W}_{ij}}{\mathbf{W}_{ik}} = \frac{\mathbf{x}_j\frac{\partial \mathbf{y}}{\partial \mathbf{x}_j}}{\mathbf{x}_k\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k}}. \tag{5}$$

By comparing $\mathbf{x}_j\frac{\partial \mathbf{y}}{\partial \mathbf{x}_j}$ and $\mathbf{x}_k\frac{\partial \mathbf{y}}{\partial \mathbf{x}_k}$, we can determine whether $\mathbf{x}_j$ and $\mathbf{x}_k$ exist in the same latent variable. If there

is a clear linear relationship between $\mathbf{x}_j \frac{\partial \mathbf{y}}{\partial \mathbf{x}_j}$ and $\mathbf{x}_k \frac{\partial \mathbf{y}}{\partial \mathbf{x}_k}$ calculated at multiple points, it indicates a high probability that $\mathbf{x}_j$ and $\mathbf{x}_k$ exist in the same latent variable, or that they contribute to this latent variable more than other latent variables. Because the dataset is discrete, we cannot obtain the exact value of $\mathbf{x}_j \frac{\partial \mathbf{y}}{\partial \mathbf{x}_j}$, but we can get reliable estimation based on the difference. Here we define

$$\boldsymbol{\rho}_j = \mathbf{x}_j \frac{\Delta \mathbf{y}_j}{\Delta \mathbf{x}_j}, \tag{6}$$

where $\Delta \mathbf{x}_j$ is the change in $\mathbf{x}_j$ and $\Delta \mathbf{y}_j$ is the corresponding change in $\mathbf{y}$, and have

$$\frac{\mathbf{W}_{ij}}{\mathbf{W}_{ik}} \approx \frac{\boldsymbol{\rho}_j}{\boldsymbol{\rho}_k}. \tag{7}$$

Based on this, we can calculate the Pearson correlation coefficient between $\boldsymbol{\rho}_1$ and $\boldsymbol{\rho}_p$. A larger correlation indicates a higher probability of being associated with the same latent variable, allowing us to estimate the number of latent variables, the ratios between weights as well as their positive or sign of the correlation, which greatly reduces the search space.

Parameter Optimization

Dimensional Invariance. To ensure that the resulting equation has physical meaning, it is necessary to ensure dimension consistency between $f_2 \circ f_1(\mathbf{x})$ and $\mathbf{y}$, i.e.,

$$f_2(\mathbf{D}\mathbf{W}^T) = \mathbf{d}, \tag{8}$$

where $\mathbf{D} \in \mathbb{R}^{7\times p}$ is the dimension matrix of $\mathbf{x}$ and $\mathbf{d} \in \mathbb{R}^7$ is the dimension vector of $\mathbf{y}$. As shown in Fig. 1b, the dimension matrix $\mathbf{D} = [\mathbf{d}_1, \ldots, \mathbf{d}_p]$ consists of dimension vectors for each corresponding input variable. The dimension vector represents the exponents of physical quantities concerning the fundamental dimensions in the natural world—mass [M], length [L], time [T], temperature [Θ], electric current [I], luminous intensity [J], and amount of substance [N]. For example, the dimension for gravitational acceleration is m/s^2, and its dimension vector is $[0, 1, -2, 0, 0, 0, 0]^T$.

From Eq. (1), the i-th latent variable $\mathbf{z}_i$ is determined by $\mathbf{W}$'s i-th row $\mathbf{W}_i$. When the output $\mathbf{y}$ has no unit, i.e., $\mathbf{d} = 0$, we set $\mathbf{D}\mathbf{W}_i^T = 0$ to ensure $\mathbf{z}_1, \ldots, \mathbf{z}_s$ are dimensionless numbers, and the degree of $f_2(\cdot)$ is not limited. When $\mathbf{y}$ has a unit, i.e., $\mathbf{d} \neq 0$, we set $\mathbf{D}\mathbf{W}_i^T = \mathbf{d}$ to ensure the latent variables have the same unit as the output, and use a linear regression model to fit $\mathbf{y}$ and $\mathbf{z}$. Incorporating the above two cases, we have the following constraints on the weight

$$\mathbf{D}\mathbf{W}_i^T = \mathbf{d}, \tag{9}$$

which ensures consistency among the dimensions of latent variables and the output. One can get the closed-form solution to the above equation

$$\mathbf{W}_i^T = \sum_{k=1}^{p-r(\mathbf{D})} \boldsymbol{\lambda}_{ik}\mathbf{w}_k + \mathbf{w}^*, \tag{10}$$

where $\{\mathbf{w}_k\}$ is the set of homogeneous solutions that satisfies $\mathbf{D}\mathbf{w}_k = 0$, and $\mathbf{w}^*$ is a particular solution to $\mathbf{D}\mathbf{w}^* = \mathbf{d}$. As shown in Fig. 1b, once we have obtained $\{\mathbf{w}_k\}$ and $\mathbf{w}^*$, the task of searching for $\mathbf{W} \in \mathbb{R}^{s\times p}$ turns into searching for $\boldsymbol{\lambda} \in \mathbb{R}^{s\times(p-r(\mathbf{D}))}$ with $r(\cdot)$ denoting the rank of a matrix.

Prior Constraints. Eq. (10) transforms the search space from $s \times p$ to $s \times (p - r(\mathbf{D}))$, but the search space remains large. Here, we restrict $\mathbf{W}$ from various perspectives to narrow down $\boldsymbol{\lambda}$'s search space.

(i) Dataset Constraint. The dataset contains a wealth of information, which can be utilized to refine the search scope. From Eq. (1), there exists $\mathbf{x}_j^{\mathbf{W}_{ij}}$ term in $\mathbf{z}_i$. If $\exists \mathbf{x}_j < 0$ in the dataset $(\mathbf{X}, \mathbf{Y})$, we let $\mathbf{W}_{ij} \in \mathbb{Z}$ to avoid the occurrence of imaginary numbers. If $\exists \mathbf{x}_j = 0$ in the dataset, we force $\mathbf{W}_{ij} \geq 0$ to avoid a zero divisor.

$$\begin{cases} \mathbf{W}_{ij} \in \mathbb{Z}, & \text{if } \exists \mathbf{x}_j < 0 \\ \mathbf{W}_{ij} \geq 0. & \text{if } \exists \mathbf{x}_j = 0 \end{cases} \tag{11}$$

(ii) Equivalence Constraint. Some weight coefficients are equivalent when the number of latent variables is greater than 1, e.g., if $\mathbf{z}_1 = \mathbf{z}_2$, it is unnecessary to introduce an additional variable $\mathbf{z}_2$, since the weight $[\mathbf{W}_1, \mathbf{W}_2] \sim [\mathbf{W}_1]$. Besides, exchanging two rows of $\mathbf{W}$ will not affect the result, e.g., the case $\mathbf{z}_1 = \mathbf{x}_1^1\mathbf{x}_2^2\mathbf{x}_3^3, \mathbf{z}_2 = \mathbf{x}_1^4\mathbf{x}_2^5\mathbf{x}_3^6$ is equivalent to $\mathbf{z}_1 = \mathbf{x}_1^4\mathbf{x}_2^5\mathbf{x}_3^6, \mathbf{z}_2 = \mathbf{x}_1^1\mathbf{x}_2^2\mathbf{x}_3^3$. Mathematically, we have

$$\begin{cases} [\mathbf{W}_i, \mathbf{W}_k] \sim [\mathbf{W}_i], & \text{if } \mathbf{W}_i = \mathbf{W}_k \\ [\mathbf{W}_i, \mathbf{W}_k] \sim [\mathbf{W}_k, \mathbf{W}_j]. & \text{else} \end{cases} \tag{12}$$

(iii) Sparsity Constraint. In fact, in most cases, each latent variable is only composed of a partial combination of input variables, i.e., $\mathbf{W}$ is a sparse matrix

$$\mathbf{W}_{ij} \begin{cases} = 0, & \mathbf{x}_j \nrightarrow \mathbf{z}_i \\ \neq 0. & \mathbf{x}_j \rightarrow \mathbf{z}_i \end{cases} \tag{13}$$

To find concise and meaningful input combinations, we impose restrictions on the sparsity of the data. Specifically, We force $\mathbf{W}$ to have at most κ_1 non-zero value and each column has no more than κ_2 non-zero entries, i.e., each input is associated with at most κ_2 latent variables:

$$\begin{cases} \|\{\mathbf{W}_{ij} | \mathbf{W}_{ij} \neq 0, i = 1, \ldots, s, j = 1, \ldots, p\}\| \leq \kappa_1 \\ \|\{\mathbf{W}_{ij} | \mathbf{W}_{ij} \neq 0, i = 1, \ldots, s\}\| \leq \kappa_2, \ j = 1, \ldots, p. \end{cases} \tag{14}$$

C2F Search. Assuming we have an estimated version of $\boldsymbol{\lambda}$—$\hat{\boldsymbol{\lambda}}$, the latent variables for dataset $(\mathbf{X}, \mathbf{Y})$ can be estimated as

$$\hat{\mathbf{Z}} = f_1(\mathbf{X}|\hat{\boldsymbol{\lambda}}). \tag{15}$$

We minimize the least squares error to perform polynomial regression on $\hat{\mathbf{Z}}$ and $\mathbf{Y}$ and obtain $f_2(\cdot|\hat{\boldsymbol{\lambda}})$, an estimate of $f_2(\cdot)$. The predicted data $\hat{\mathbf{Y}}$ can be calculated as

$$\hat{\mathbf{Y}} = f_2(\hat{\mathbf{Z}}|\hat{\boldsymbol{\lambda}}). \tag{16}$$

We use the coefficient of determination R^2 to measure the performance

$$R^2 = 1 - \frac{\sum_{i=1}^{b}(\mathbf{Y}_i - \hat{\mathbf{Y}}_i)^2}{\sum_{i=1}^{b}(\mathbf{Y}_i - \bar{\mathbf{Y}})^2}, \tag{17}$$

where $\bar{\mathbf{Y}} = (\sum_{i=1}^{b} \mathbf{Y}_i)/b$ is the mean of $\mathbf{Y}$.

When $\boldsymbol{\lambda}$ is determined, the polynomial coefficients for $f_2(\cdot)$ can be quickly calculated, so the challenge lies in

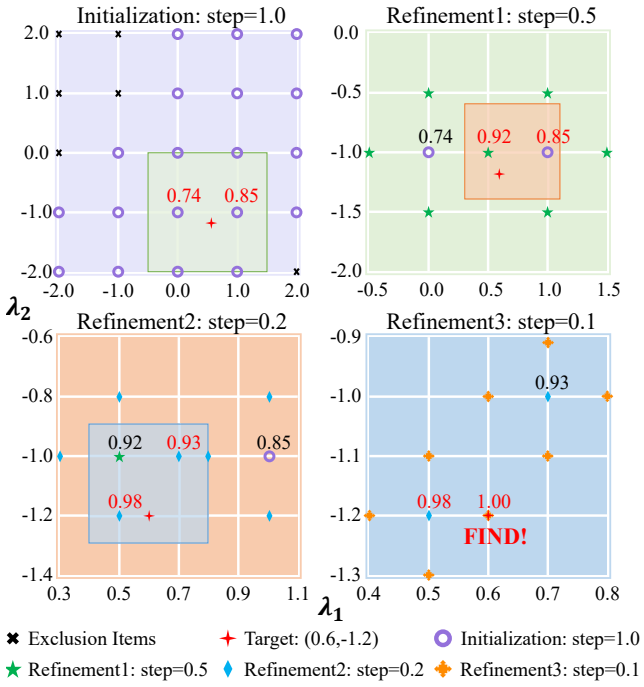

Figure 2: An example of C2F search. The initialization stage searches $[-2, 2]^2$ with a step of 1, obtaining a rough probability distribution map of the solution. The three refinement stages, by iteratively searching with smaller steps around the top coefficients, gradually shrink the solution's range and move toward the optimal position.

searching $\boldsymbol{\lambda}$. In most cases, people tend to use input combinations with small exponents, like the law of universal gravitation $F = Gm_1m_2r^{-2}$, and Kepler's third law $T = ka^{1.5}$. Here, we limit $\boldsymbol{\lambda} \in [-2, 2]^c$, where $c = s \times (p - r(\mathbf{D}))$.

There exist some typical options for searching the $\boldsymbol{\lambda}$. If we use a gradient optimization algorithm for searching, $\boldsymbol{\lambda}$ tends to get stuck in local optima and results in irregular decimals such as 0.5234, rather than concise ones like 0.5. If linear searching is applied, we can avoid local optimal issues but encounter a huge search space. Therefore, we propose a coarse-to-fine (C2F) optimization framework to gradually search for the optimal solution from coarse to fine. By initialization, we locate the rough position of the target and then progressively refine the searches to increasingly smaller ranges and toward the optimal value.

(i) Initialization. We firstly divide $[-2, 2]^c$ with a step of 1 to obtain 5^c initial estimations of $\boldsymbol{\lambda}$ and then exclude a lot of infeasible searching candidates with the prior constraints proposed before. For the left estimations $\hat{\boldsymbol{\lambda}}$, one can obtain their R^2 scores according to Eqns. (15)(16)(17) and record the candidates with top performances.

(ii) Refinement. We perform the next round of search with a step of 0.5 around the recorded top coefficients to obtain the new R^2 distribution and update the top coefficients. Then we repeat this process progressively, decreasing the step from 0.5 to 0.2 and finally to 0.1.

If we want $\boldsymbol{\lambda}$ to be precise to 0.1, the number of candidates to be searched with the linear search algorithm and C2F algorithm is respectively

$$\begin{cases} \text{Linear Search:} & n(\hat{\boldsymbol{\lambda}}) = 41^c \\ \text{C2F Search:} & n(\hat{\boldsymbol{\lambda}}) \ll 5^c + 3t2^c \end{cases} \tag{18}$$

where t refers to searching around the top t candidates.

A simple example is shown in Fig. 2, in which we need to find the best estimation for $(\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2)$, whose true solution is $(0.6, -1.2)$. In the initialization stage, we divide $[-2, 2]^2$ with a step of 1 and obtain 25 candidate values, with 6 exclusion items due to prior constraints. According to the R^2 metric for each candidate, $(0.0, -1.0)$ and $(1.0, -1.0)$ have the highest R^2 scores. In the first round of refinement, we explore the vicinity of $(0.0, -1.0)$ and $(1.0, -1.0)$ with a step size of 0.5, resulting in 7 new candidates, and the top 2 results are $(0.5, -1.0)$ and $(1.0, -1.0)$. In the subsequent round of refinement, we focus on the updated top coefficients with a smaller step size. The search path progresses as follows: $(1.0, -1.0) \rightarrow (0.5, -1.0) \rightarrow (0.5, -1.2) \rightarrow (0.6, -1.2)$. By iteratively exploring smaller step sizes around the leading coefficients and gradually narrowing the target range, we notably improve the likelihood of locating the optimal solution.

Note that the C2F search is quite flexible. When the search space for $\boldsymbol{\lambda}$ is large, we can change the initialization step from 1.0 to 2.0 or even larger, and the refine step size to [1.0,0.5,0.2,0.1], which can greatly reduce the search space. Besides, the C2F search reduces the possibility of falling into local optima and greatly reduces the search space. Compared to irregular decimals, the coefficients obtained by C2F align better with human intuitions.

Experiments

In this section, we first validate the conclusion in Eq. (7) and demonstrate that by estimating the $\boldsymbol{\rho}$-values, we can obtain the number of latent variables, the connection relationships as well as the weight ratios. We then introduce three typical applications of the FIND framework, including discovering dimensionless functions, dimensionless numbers, and physical laws. All the experiment details are accessible in the Supplementary Material.

Identifying the Latent Variables

We designed two functions—5D and 7D respectively, to demonstrate our capability of identifying latent variables, as illustrated in Fig. 3a.

5D-Function Example. The latent variables and expression of the first function is

$$\begin{aligned} &\mathbf{z}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1.0}, \mathbf{z}_2 = \mathbf{x}_3^{-1.2}\mathbf{x}_4^{1.4}, \mathbf{z}_3 = \mathbf{x}_5^{1.0} \\ &\mathbf{y} = 3 + 0.4\mathbf{z}_1 + 1.3\mathbf{z}_2 - 0.7\mathbf{z}_3 + 0.6\mathbf{z}_1\mathbf{z}_2 \\ &\quad + 1.2\mathbf{z}_3^2 + \mathbf{z}_1\mathbf{z}_2\mathbf{z}_3. \end{aligned} \tag{19}$$

We sampled 3125 points in the input domain, calculated the difference with $\Delta\mathbf{x} = 0.04$ to estimate the partial derivatives of each point, and obtained the $\boldsymbol{\rho}_1 \sim \boldsymbol{\rho}_5$ values on each point with Eq. (6). According to Eq. (7), if both $\mathbf{x}_j$ and $\mathbf{x}_k$

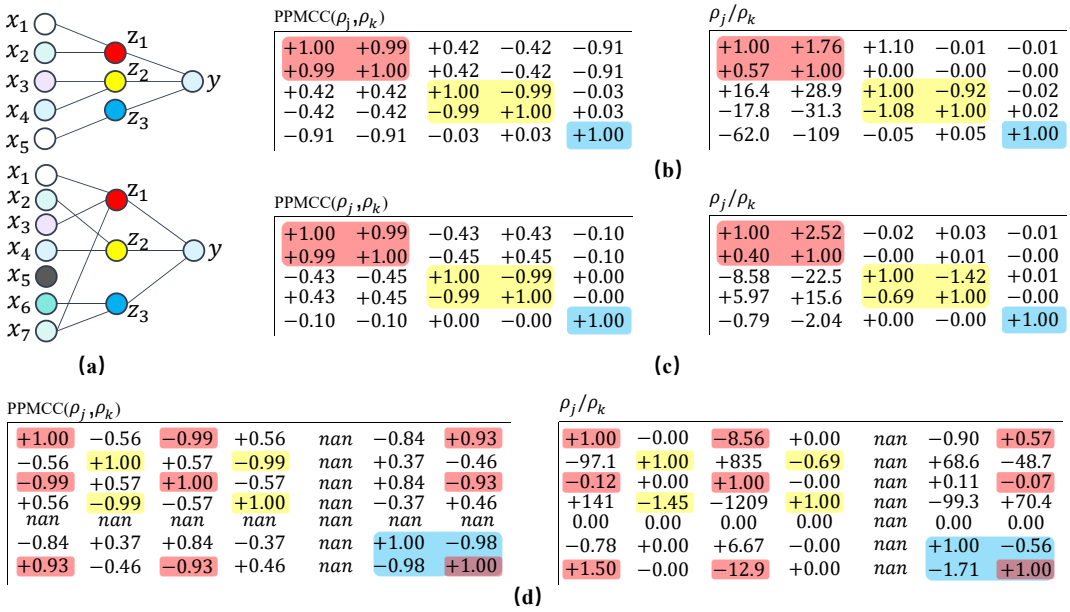

Figure 3: Identification of latent variables. (a) A 5-D function and a 7-D function. (b-d) show the PPMCC (left) and ratio tables (right) for the 5-D function with $\Delta\mathbf{x} = 0.04$, the 5-D function with $\Delta\mathbf{x} = 0.4$, and the 7-D function with $\Delta\mathbf{x} = 0.04$.

are connected to $\mathbf{z}_i$ and their weights on $\mathbf{z}_i$ are much greater than their weights on other latent variables, then $\boldsymbol{\rho}_j$ and $\boldsymbol{\rho}_k$ show a clear proportional relationship.

We calculated the Pearson product-moment correlation coefficient (PPMCC) between $\boldsymbol{\rho}_j$ and $\boldsymbol{\rho}_k$, and used the least squares method to calculate their slope to estimate $\boldsymbol{\rho}_j/\boldsymbol{\rho}_k$. The PPMCC and ratio tables are shown in Fig. 3b: from which we can get $\{\mathbf{x}_1, \mathbf{x}_2\} \to \mathbf{z}_1$, $\{\mathbf{x}_3, \mathbf{x}_4\} \to \mathbf{z}_2$, $\{\mathbf{x}_5\} \to \mathbf{z}_3$, directly displaying the number of latent variables and the connection relationships; the ratio table shows the estimations for the weight ratios $\boldsymbol{\rho}_1/\boldsymbol{\rho}_2 = 1.76, \boldsymbol{\rho}_3/\boldsymbol{\rho}_4 = -0.92$, which consist with the true values $-1.7/-1.0 = 1.7, -1.2/1.4 = -0.86$ and can greatly reduce the search space by providing a rough range via partial derivatives.

In Fig. 3c, we increased $\Delta\mathbf{x}$ to 0.4. The PPMCC results remain consistent with the original function. However, the estimated ratio values are beginning to diverge from the true value due to inaccurate derivative estimation. Nevertheless, we can still get the sign of the between-weight correlation.

7D-Function Example. The latent variables and expression of the second examplar function is

$$\begin{aligned}\mathbf{z}_1 &= \mathbf{x}_1^{-1.7}\mathbf{x}_3^{0.2}\mathbf{x}_7^{-1.0}, \mathbf{z}_2 = \mathbf{x}_2^{1.0}\mathbf{x}_4^{-1.3}, \mathbf{z}_3 = \mathbf{x}_6^{-0.6}\mathbf{x}_7^{0.7},\\ \mathbf{y} &= \sin(2\mathbf{z}_1 + \pi/3) - \mathbf{z}_1\mathbf{z}_2 + \mathrm{e}^{\mathbf{z}_1\mathbf{z}_3} + \sin(\mathbf{z}_3^2)\\ &\quad + \mathbf{z}_1\mathbf{z}_2\mathbf{z}_3 + \mathbf{z}_2^2.\end{aligned} \tag{20}$$

We set $\Delta\mathbf{x} = 0.04$ and get its PPMCC and ratio tables in Fig. 3d. The PPMCC results show that $\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_7\} \to \mathbf{z}_1$, $\{\mathbf{x}_2, \mathbf{x}_4\} \to \mathbf{z}_2$, $\{\mathbf{x}_6, \mathbf{x}_7\} \to \mathbf{z}_3$, and $\mathbf{x}_5$ is an independent variable. One can see that although the expression is not in polynomial form, we can still find the latent variables and the connection relationship correctly. The ratio table also reflects the ratio relationship between weights very well, except for $\mathbf{x}_7$ which cannot be accurately estimated due to the simultaneous connection with two latent variables.

Recall that not all observations are complete enough for reliable derivative estimation, so this estimation method might be inapplicable for extremely sparse data. In the subsequent experiments, we show the results of our C2F framework searching for the optimal solution from sparse data.

Application #1: Finding Dimensionless Functions

We collected datasets from 7 distinct systems and employed our FIND framework to identify the original functions, as demonstrated in Tab. 1. All 7 datasets consist of simulation data with 1% Gaussian noise. Notably, no unit is assigned to the input and output, i.e., $\mathbf{D} = 0, \mathbf{d} = 0$.

The experimental findings demonstrate that FIND excels in identifying latent variables across all scenarios. In the first experiment, employing C2F search only necessitates exploring 186 potential points, contrasting with a linear search that would entail investigating 41^3 points. Throughout experiments 1 to 3, augmenting the input variables from 3 to 5 did not impede the successful identification of both latent variables and expressions. Despite the original expression in experiment 4 deviating from a polynomial form, our method adeptly uncovers the latent variable and derives a polynomial surrogate for the initial expression. In experiments 5 to 7, where there are 2 or 3 latent variables, the original functions can still be efficiently and accurately determined.

Under the C2F framework, we iteratively refine the search space to pinpoint the optimal point effectively. This process enables us to identify the optimal solution even when

the original expression deviates from a polynomial form. In such cases, we can still derive a polynomial representation that serves as a viable replacement.

Application #2: Finding Dimensionless Numbers

Dimensionless numbers are quantities used in physics and engineering to describe and analyze problems without specific units, playing a crucial role in understanding and predicting natural phenomena and designing engineering systems. These numbers normalize problems, remove unit dependencies, and facilitate comparisons across various scenarios. Here, we assess our method for identifying dimensionless numbers in both static and dynamic systems.

Static System. The laser–metal interaction is an important problem. During this interaction, a depression filled with vapor, known as a keyhole, typically emerges on the molten metal surface. The formation of the keyhole stems from the recoil pressure induced by vaporization. Owing to its intricate reliance on numerous physical mechanisms, comprehending the kinetic essence of the keyhole poses inherent challenges.

The keyhole size e is related to a lot of parameters, such as the laser power ηP, the laser scan speed V_s, the laser beam radius r_0, the thermal diffusivity α, the material density ρ_0, the heat capacity C_p, and the difference between melting and ambient temperatures ΔT. We assess the performance of our FIND framework using a dataset about keyholes (Xie et al. 2022), encompassing 90 experiments conducted on three distinct materials: titanium alloy (Ti6Al4V), aluminum alloy (Al6061), and stainless steel (SS316) (Zhao et al. 2019; Gan et al. 2021). The output variable e is normalized as the keyhole aspect ratio denoted as $e^* = e/r_0$.

When the coefficient accuracy is set to 0.1, the outcome is

$$\begin{aligned} z &= \frac{\eta P^{1.6}}{V_s^{0.7} r_0^{2.3} \alpha^{0.9} \rho_0^{1.6} C_p^{1.6} \Delta T^{1.6}}, \\ e^* &= -0.04 + 0.02z, \end{aligned} \tag{21}$$

with a high $R^2 = 0.9865$. The weights obtained in this case are 1.6, -0.7, which may not align with human conventions. Consequently, we adjusted the accuracy to 0.5 and conducted another test, yielding

$$\begin{aligned} z &= \frac{\eta P}{\rho_0 C_p \Delta T \sqrt{\alpha V_s r_0^3}}, \\ e^* &= -0.61 + 0.15z, \end{aligned} \tag{22}$$

with $R^2 = 0.9810$. The latent variable z divided by π is a discovered keyhole number Ke (Gan et al. 2021; Ye et al. 2019), which can be derived from heat transfer theory.

Dynamic System. We use a dataset of Navier-Stokes equations with different Reynolds numbers (Xie et al. 2022) to demonstrate FIND's potential in discovering dimensionless numbers in partial differential equations (PDEs). By changing dynamic viscosity μ, cylinder diameter l, inlet velocity v, fluid density ρ_0, and the pressure difference p_0, different PDEs are created. In each PDE scenario, there are six variables t, x, y, u, v, w, and we use SINDy (Brunton, Proctor, and Kutz 2016) to process the data for each scenario and obtain the corresponding PDE equation. All the discovered PDEs have the following form

$$\frac{\partial w}{\partial t} = \lambda_1 u \frac{\partial w}{\partial x} + \lambda_2 v \frac{\partial w}{\partial y} + \lambda_3 \frac{\partial^2 w}{\partial x^2} + \lambda_4 \frac{\partial^2 w}{\partial y^2}. \tag{23}$$

There are three sets of PDE parameters here, which are

$$\boldsymbol{\lambda} = \begin{cases} [-0.9925, -0.9925, +0.0212, +0.0212]^T, \\ [-0.9909, -0.9909, +0.0126, +0.0126]^T, \\ [-0.9941, -0.9941, +0.0111, +0.0111]^T. \end{cases} \tag{24}$$

We can infer that $\lambda_1 = \lambda_2 = -1$ are fix coefficients, while $\lambda_3 = \lambda_4$ are dynamic dimensionless numbers. We use ρ_0, μ, v, l, p_0 as inputs and λ_3, λ_4 as the outputs to search for dimensionless numbers, and find that

$$\lambda_3 = \lambda_4 = \frac{\mu}{\rho_0 v l} = \frac{1}{Re}, \tag{25}$$

where Re is the Reynolds number (Reynolds 1883). Finally, we get a unified PDE form

$$\frac{\partial w}{\partial t} = -u \frac{\partial w}{\partial x} - v \frac{\partial w}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} \right). \tag{26}$$

Application #3: Finding Physical Laws

We examined a real-world dataset of planets in the solar system sourced from the NASA Planetary Fact Sheet (NASA 2017). The variables tested include planet mass m, planet diameter d_0, planet density ρ_0, gravitational acceleration g, escape velocity v_e, rotation period t_r, length of day t_d, distance from sun r_s, perihelion r_p, aphelion r_a, orbital period t_o, orbital velocity v_o. The set $S = \{m, d_0, \rho_0, g, v_e, t_r, t_d, r_s, r_p, r_a, t_o, v_o\}$ encompasses all these variables.

We explore the formulas in two ways, as shown in Tab. 2. The initial method involves utilizing physical units to confine the search space, thereby directly deriving relevant physical formulas. In the second method, input/output units are disregarded, and a dimensional constant is appended after the formula.

Consider m as the output and the other variables $S \backslash \{m\}$ as the input, FIND yielded the result $m = 0.45 d_0^3 \rho_0$. This formula establishes the connection among mass m, volume $\pi d_0^3/6$, and density ρ_0, with the theoretical expression being $m = 0.52 d_0^3 \rho_0$. Taking v_e as the output and $S \backslash \{v_e\}$ as the input, we discovered the relationship $v_e = 1.04\sqrt{g d_0}$ that corresponds to the escape velocity formula, while the ground-truth formulation is $v_e = \sqrt{g d_0}$. When designating t_o as the output and $S \backslash \{t_o\}$ as the input, the result is $t_o = 6.21 r_s / v_o$. This formula describes the connection among orbital period t_o, orbital circumference $2\pi r_s$, and orbital velocity v_0, and the true expression is $t_o = 6.28 r_s / v_o$.

In the preceding cases, all units of the inputs and outputs were considered, resulting in formulas with inherent physical meaning without additional adjustments. Subsequently, the exploration of formulas was conducted disregarding the input and output units. To prevent the rediscovery of already

ID	Latent	Expression	$n(\hat{\lambda})$	$\mathbf{R}^2$	Time
1	$\mathbf{z} = \mathbf{x}_1^{-1.2}\mathbf{x}_2^{-0.5}\mathbf{x}_3^{1.0}$	$\mathbf{y} = 2 + 0.7\mathbf{z} + 1.5\mathbf{z}^2 + 3.6\mathbf{z}^3$	186	0.99	0.54
	$\hat{\mathbf{z}} = \mathbf{x}_1^{-1.2}\mathbf{x}_2^{-0.5}\mathbf{x}_3^{1.0}$	$\hat{\mathbf{y}} = 2 + 0.7\hat{\mathbf{z}} + 1.5\hat{\mathbf{z}}^2 + 3.6\hat{\mathbf{z}}^3$			
2	$\mathbf{z} = \mathbf{x}_1^{1.7}\mathbf{x}_2^{-0.7}\mathbf{x}_3^{0.2}\mathbf{x}_4^{-0.4}$	$\mathbf{y} = 2.00 + 1.60\mathbf{z} - 1.80\mathbf{z}^2 + 3.60\mathbf{z}^3$	1600	0.99	3.59
	$\hat{\mathbf{z}} = \mathbf{x}_1^{1.7}\mathbf{x}_2^{-0.7}\mathbf{x}_3^{0.2}\mathbf{x}_4^{-0.4}$	$\hat{\mathbf{y}} = 1.99 + 1.59\hat{\mathbf{z}} - 1.77\hat{\mathbf{z}}^2 + 3.59\hat{\mathbf{z}}^3$			
3	$\mathbf{z} = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-0.7}\mathbf{x}_3^{0.2}\mathbf{x}_4^{-1.3}\mathbf{x}_5^{1.5}$	$\mathbf{y} = 6 + 3.6\mathbf{z} - 1.80\mathbf{z}^2 - 1.3\mathbf{z}^3$	5777	0.99	14.04
	$\hat{\mathbf{z}} = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-0.7}\mathbf{x}_3^{0.2}\mathbf{x}_4^{-1.3}\mathbf{x}_5^{1.5}$	$\hat{\mathbf{y}} = 6 + 3.6\hat{\mathbf{z}} - 1.84\hat{\mathbf{z}}^2 - 1.2\hat{\mathbf{z}}^3$			
4	$\mathbf{z} = \mathbf{x}_1^{-1.2}\mathbf{x}_2^{-0.5}\mathbf{x}_3^{1.0}$	$\mathbf{y} = \tanh(\sin(\mathbf{z} + \pi)) + \mathbf{z}^2$	198	0.99	0.51
	$\hat{\mathbf{z}} = \mathbf{x}_1^{-1.2}\mathbf{x}_2^{-0.5}\mathbf{x}_3^{1.0}$	$\hat{\mathbf{y}} = 0.7 + 0.28\hat{\mathbf{z}} + 0.73\hat{\mathbf{z}}^2$			
5	$\mathbf{z}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1}\mathbf{x}_3^{0.2}, \mathbf{z}_2 = \mathbf{x}_4^{-1.3}\mathbf{x}_5$	$\mathbf{y} = 3 + 0.40\mathbf{z}_1 + 1.3\mathbf{z}_2 + 0.60\mathbf{z}_1\mathbf{z}_2 + 1.2\mathbf{z}_2^2$	3343	0.99	12.23
	$\hat{\mathbf{z}}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1}\mathbf{x}_3^{0.2}, \hat{\mathbf{z}}_2 = \mathbf{x}_4^{-1.3}\mathbf{x}_5$	$\hat{\mathbf{y}} = 3 + 0.34\hat{\mathbf{z}}_1 + 1.3\hat{\mathbf{z}}_2 + 0.61\hat{\mathbf{z}}_1\hat{\mathbf{z}}_2 + 1.2\hat{\mathbf{z}}_2^2 + \dots$			
6	$\mathbf{z}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1}\mathbf{x}_3^{0.2}, \mathbf{z}_2 = \mathbf{x}_4^{-1.3}\mathbf{x}_5$	$\mathbf{y} = -\mathbf{z}_1\mathbf{z}_2 + \mathbf{z}_2^2 + \sin(2\mathbf{z}_1 + \pi/3)$	3679	0.99	15.83
	$\hat{\mathbf{z}}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1}\mathbf{x}_3^{0.2}, \hat{\mathbf{z}}_2 = \mathbf{x}_4^{-1.3}\mathbf{x}_5$	$\hat{\mathbf{y}} = -\hat{\mathbf{z}}_1\hat{\mathbf{z}}_2 + \hat{\mathbf{z}}_2^2 + 0.86 + 0.89\hat{\mathbf{z}}_1 + \dots$			
7	$\mathbf{z}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1}, \mathbf{z}_2 = \mathbf{x}_3^{-1.3}\mathbf{x}_4^{1.4}, \mathbf{z}_3 = \mathbf{x}_5$	$\mathbf{y} = 3 + 1.3\mathbf{z}_2 + 0.6\mathbf{z}_1\mathbf{z}_2 + 1.2\mathbf{z}_3^2 + \mathbf{z}_1\mathbf{z}_2\mathbf{z}_3$	2992	0.99	185.63
	$\hat{\mathbf{z}}_1 = \mathbf{x}_1^{-1.7}\mathbf{x}_2^{-1}, \hat{\mathbf{z}}_2 = \mathbf{x}_3^{-1.3}\mathbf{x}_4^{1.4}, \hat{\mathbf{z}}_3 = \mathbf{x}_5$	$\hat{\mathbf{y}} = 3 + 1.3\hat{\mathbf{z}}_2 + 0.6\hat{\mathbf{z}}_1\hat{\mathbf{z}}_2 + 1.2\hat{\mathbf{z}}_3^2 + \hat{\mathbf{z}}_1\hat{\mathbf{z}}_2\hat{\mathbf{z}}_3 + \dots$			

Table 1: Toy dataset with 1% Gaussian noise.

	Input	Output	FIND	Theory
$D \neq 0, d \neq 0$	$S \backslash \{m\}$	m	$m = 0.49 d_0^3 \rho_0$	$m = 0.52 d_0^3 \rho_0$
	$S \backslash \{v_e\}$	v_e	$v_e = 1.04\sqrt{g d_0}$	$v_e = \sqrt{g d_0}$
	$S \backslash \{t_o\}$	t_o	$t_o = 6.21 r_s / v_o$	$t_o = 6.28 r_s / v_o$
$D = 0, d = 0$	$d_0, g, t_r, t_d, r_s, t_o$	m	$m = 4.04e^{+09} d_0^2 g$	$m = 3.75e^{+09} d_0^2 g$
	m, d_0, g, t_d, r_s	v_e	$v_e = 1.63e^{-05}\sqrt{m/d_0}$	$v_e = 1.63e^{-05}\sqrt{m/d_0}$
	m, d_0, g, t_r, t_d, r_s	t_o	$t_o = 5.43e^{-10} r_s^{1.5}$	$t_o = 5.46e^{-10} r_s^{1.5}$

Table 2: Formulas found with FIND.

established formulas, a subset of variables was chosen for each experiment. For instance, after selecting m and d_0, we exclude ρ_0 because $m = 0.52 d_0^3 \rho_0$.

Here we provide the law discovery for three different outputs. (i) When m is designated as the output, the obtained formula is $m = 4.04e^{+09} d_0^2 g$. This formula is typically used for calculating planet mass, and the theoretical expression is $m = d_0^2 g/(4G) = 3.75e^{+09} d_0^2 g$, where G represents the constant of universal gravitation. To maintain consistency in dimensions on both sides of the formula, the unit $s^2 kg/m^3$ is appended to the constant $4.04e^{+09}$. (ii) When v_e is considered as the output, the formula obtained is $v_e = 1.63e^{-05}\sqrt{m/d_0}$. This represents an alternative calculation approach for escape velocity, where $v_e = \sqrt{4Gm/d_0} = 1.63e^{-05}\sqrt{m/d_0}$. The unit of the constant $1.63e^{-05}$ is $m^{1.5} s^{-1} kg^{-0.5}$. (iii) When t_o is the output, the result is $t_o = 5.43e^{-10} r_s^{1.5}$. This formula corresponds to Kepler's third law, which states $t_o = r_s^{1.5}/\sqrt{K} = 5.46e^{-10} r_s^{1.5}$, where K denotes the Kepler constant. The unit for the constant $5.43e^{-10}$ is $s/m^{1.5}$.

Summary and Conclusions

To efficiently discover formulas from observations, we propose the FIND framework consisting of a latent layer and an expression layer: the former explores meaningful input combinations and reduces data dimension, while the latter pursues explicit expressions from latent variables to output.

To analyze the latent structure, we analyze the relationship between weights and derivatives. By statistically analyzing the linear correlation and ratio of $\boldsymbol{\rho}$ values from multiple points, we can obtain the connection relationships and the weight ratios. To get the optimal weights, we propose the C2F framework, which gradually depicts the optimal probability graph from coarse to fine, which greatly reduces the optimization time and avoids getting stuck in local optima.

FIND has built a simple, general, and explainable framework to obtain field knowledge from data observations quickly. The typical applications include discovering dimensionless functions, dimensionless numbers, and natural physical laws. We have conducted comprehensive experiments to verify the high accuracy and efficiency of FIND, as well as its wide applicability.

Limitations. Due to its fixed structure, FIND can only obtain solutions in polynomial form. We use grid search to reduce the likelihood of local optima but at the cost of increased difficulty in handling high-dimensional data. Moreover, FIND is intrinsically a data-driven method, so the accuracy of results might degenerate when the input dataset is highly dispersed.

Future Work. We will further explore better structures and more efficient solutions, especially from low-quality observations. We also plan to integrate SR techniques to convert polynomials into symbolic expressions. Moreover, we aim to broaden FIND's applications beyond fluid mechanics and astronomy, extending its use to areas such as industry, medicine, and chemistry.

Acknowledgments

This work is jointly funded by Ministry of Science and Technology of China (Grant No. 2024YFF0505703) and National Natural Science Foundation of China (Grant Nos. 61931012 and 62088102).

References

Bakarji, J.; Callaham, J.; Brunton, S. L.; and Kutz, J. N. 2022. Dimensionally consistent learning with Buckingham Pi. *Nature Computational Science*, 2(12): 834–844.

Barenblatt, G. I. 2003. *Scaling*, volume 34. Cambridge University Press.

Bergen, K. J.; Johnson, P. A.; de Hoop, M. V.; and Beroza, G. C. 2019. Machine learning for data-driven discovery in solid Earth geoscience. *Science*, 363(6433): eaau0323.

Bhaskar, R.; and Nigam, A. 1990. Qualitative physics using dimensional analysis. *Artificial Intelligence*, 45(1-2): 73–111.

Bongard, J.; and Lipson, H. 2007. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24): 9943–9948.

Brunton, S. L.; Proctor, J. L.; and Kutz, J. N. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15): 3932–3937.

Buckingham, E. 1914. On physically similar systems; illustrations of the use of dimensional equations. *Physical Review*, 4(4): 345.

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

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*, 33: 17429–17442.

Gan, Z.; Kafka, O. L.; Parab, N.; Zhao, C.; Fang, L.; Heinonen, O.; Sun, T.; and Liu, W. K. 2021. Universal scaling laws of keyhole stability and porosity in 3D printing of metals. *Nature Communications*, 12(1): 2379.

Hernandez, A.; Balasubramanian, A.; Yuan, F.; Mason, S. A.; and Mueller, T. 2019. Fast, accurate, and transferable many-body interatomic potentials by symbolic regression. *npj Computational Materials*, 5(1): 112.

Iten, R.; Metger, T.; Wilming, H.; Del Rio, L.; and Renner, R. 2020. Discovering physical concepts with neural networks. *Physical Review Letters*, 124(1): 010508.

Kamienny, P.-A.; d'Ascoli, S.; Lample, G.; and Charton, F. 2022. End-to-end symbolic regression with transformers. *Advances in Neural Information Processing Systems*, 35: 10269–10281.

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

Ma, W.; Zhang, J.; Feng, K.; Xing, H.; and Wen, D. 2024. Dimensional homogeneity constrained gene expression programming for discovering governing equations. *Journal of Fluid Mechanics*, 985: A12.

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

NASA. 2017. Planets Factsheet. https://nssdc.gsfc.nasa.gov/planetary/factsheet/.

Reynolds, O. 1883. III. An experimental investigation of the circumstances which determine whether the motion of water shall be direct or sinuous, and of the law of resistance in parallel channels. *Proceedings of the Royal Society of London*, 35(224-226): 84–99.

Saha, S.; Gan, Z.; Cheng, L.; Gao, J.; Kafka, O. L.; Xie, X.; Li, H.; Tajdari, M.; Kim, H. A.; and Liu, W. K. 2021. Hierarchical deep learning neural network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering. *Computer Methods in Applied Mechanics and Engineering*, 373: 113452.

Schmidt, M.; and Lipson, H. 2009. Distilling free-form natural laws from experimental data. *Science*, 324(5923): 81–85.

Tan, Q.-M. 2011. *Dimensional analysis: with case studies in mechanics*. Springer Science & Business Media.

Wang, H.; Fu, T.; Du, Y.; Gao, W.; Huang, K.; Liu, Z.; Chandak, P.; Liu, S.; Van Katwyk, P.; Deac, A.; et al. 2023. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972): 47–60.

Weng, B.; Song, Z.; Zhu, R.; Yan, Q.; Sun, Q.; Grice, C. G.; Yan, Y.; and Yin, W.-J. 2020. Simple descriptor derived from symbolic regression accelerating the discovery of new perovskite catalysts. *Nature Communications*, 11(1): 3513.

Xiao, T.; Yang, R.; Cheng, Y.; and Suo, J. 2024a. Shop: A deep learning framework for solving high-order partial differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 16032–16039.

Xiao, T.; Zhang, W.; Cheng, Y.; and Suo, J. 2024b. HOPE: High-Order Polynomial Expansion of Black-Box Neural Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xie, X.; Samaei, A.; Guo, J.; Liu, W. K.; and Gan, Z. 2022. Data-driven discovery of dimensionless numbers and governing laws from scarce measurements. *Nature Communications*, 13(1): 7562.

Ye, J.; Khairallah, S. A.; Rubenchik, A. M.; Crumb, M. F.; Guss, G.; Belak, J.; and Matthews, M. J. 2019. Energy coupling mechanisms and scaling behavior associated with laser powder bed fusion additive manufacturing. *Advanced Engineering Materials*, 21(7): 1900185.

Zhao, C.; Guo, Q.; Li, X.; Parab, N.; Fezzaa, K.; Tan, W.; Chen, L.; and Sun, T. 2019. Bulk-explosion-induced metal spattering during laser processing. *Physical Review X*, 9(2): 021052.