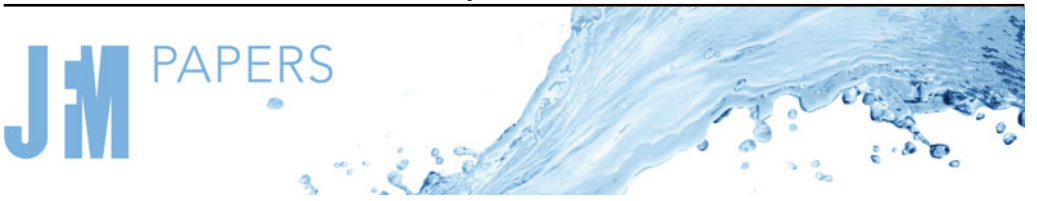
J. Fluid Mech. (2025), *vol.* 1024, A34, doi:10.1017/jfm.2025.10710



Symbolic identification of tensor equations in multidimensional physical fields

Tianyi Chen[1], **Hao Yang**[1], **Wenjun Ma**[1] **and Jun Zhang**[1]

[1]School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, PR China

Corresponding author: Jun Zhang, jun.zhang@buaa.edu.cn

(Received 2 July 2025; revised 31 August 2025; accepted 9 September 2025)

Recently, data-driven methods have shown great promise for discovering governing equations from simulation or experimental data. However, most existing approaches are limited to scalar equations, with few capable of identifying tensor relationships. In this work, we propose a general data-driven framework for identifying tensor equations, referred to as symbolic identification of tensor equations (SITE). The core idea of SITE – representing tensor equations using a host–plasmid structure – is inspired by the multidimensional gene expression programming approach. To improve the robustness of the evolutionary process, SITE adopts a genetic information retention strategy. Moreover, SITE introduces two key innovations beyond conventional evolutionary algorithms. First, it incorporates a dimensional homogeneity check to restrict the search space and eliminate physically invalid expressions. Second, it replaces traditional linear scaling with a tensor linear regression technique, greatly enhancing the efficiency of numerical coefficient optimization. We validate SITE using two benchmark scenarios, where it accurately recovers target equations from synthetic data, showing robustness to noise and flexible expressive capability. Furthermore, SITE is applied to identify constitutive relations directly from molecular simulation data, which are generated without reliance on macroscopic constitutive models. It adapts to both compressible and incompressible flow conditions and successfully identifies the corresponding macroscopic forms, highlighting its potential for data-driven discovery of tensor equation.

Key words: machine learning

1. Introduction

Physical phenomena in the real world are governed by fundamental laws. Traditionally, these laws have been derived from first principles through deductive reasoning and formulated as mathematical expressions, i.e. governing equations. However, as systems

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

become increasingly complex, this traditional first-principles approach faces significant challenges. Recent breakthroughs in artificial intelligence (AI) are revolutionizing the natural sciences, driving a paradigm shift from first-principles modelling to data-driven approaches. In particular, symbolic regression has gained increasing attention for its ability to identify analytical expressions of governing equations directly from data (Brunton, Noack & Koumoutsakos 2020; Wang *et al.* 2023).

Long before the explosive development of AI, the concept of symbolic regression had already been embodied in evolutionary algorithms. For instance, genetic programming (GP) constructs symbolic expression trees (ETs) to fit data by simulating biological evolution (Koza 1994). Bongard & Lipson (2007), as well as Schmidt & Lipson (2009), pioneered the application of GP concepts to data-driven equation discovery, successfully constructing governing equations for low-dimensional systems from experimental data. Subsequent advances in GP-based methodologies have yielded notable equation discovery frameworks, including evolutionary partial differential equations (EPDE) (Maslyaev, Hvatov & Kalyuzhnaya 2019), symbolic genetic algorithm to discover open form partial differential equations (PDEs) (SGA-PDE) (Chen *et al.* 2022) and PySR (Cranmer 2023). Another evolutionary algorithm used for symbolic regression is gene expression programming (GEP) (Ferreira 2001). Unlike GP, GEP employs fixed-length linear strings to control expression complexity and utilizes open reading frames (ORFs) to improve expression diversity. The GEP have been applied to the construction of Reynolds stress models (Weatheritt & Sandberg 2016; Zhao *et al.* 2020), subgrid-scale models for large eddy simulations (Li *et al.* 2021) and non-equilibrium constitutive equations (Ma *et al.* 2024*b*). In parallel, a distinct class of symbolic regression approaches based on sparse regression has emerged. A landmark contribution is the sparse identification of nonlinear dynamics (SINDy) method developed by Brunton, Proctor & Kutz (2016). The SINDy method transforms the discovery of ordinary differential equations into a sparse regression problem by identifying dominant terms from a library of nonlinear functions. Subsequent developments in this direction include the use of constrained Galerkin projection (Loiseau & Brunton 2018), the application of sparse regression to macroscopic hydrodynamic systems (Zhang & Ma 2020), the incorporation of weak-form formulations to expand candidate libraries (Gurevich *et al.* 2024) and the inference of hidden stochastic dynamics from high-dimensional data (Lu *et al.* 2024).

While data-driven equation discovery has made significant strides, studies have pointed out that purely data-driven methods are highly sensitive to data quality and quantity (Chen, Liu & Sun 2021; Reinbold *et al.* 2021; Cohen, Beykal & Bollas 2024). Such methods often suffer from overfitting and yield expressions that lack physical interpretability. To address these issues, researchers have incorporated physical constraints and prior knowledge into symbolic regression. For instance, Udrescu & Tegmark (2020) guided the search using physical inductive biases; Bakarji *et al.* (2022) and Fukami, Goto & Taira (2024) both leverage the Buckingham Pi theorem to guide data-driven discovery of physical laws; Chen *et al.* (2021) combined physics-informed neural networks with sparse regression technique; Wang *et al.* (2025) combined physics-informed neural networks with unit constraints for robust PDE discovery.

Despite these advances, most existing methods focus on discovering scalar equations. As system dimension increases, the candidate function space in scalar regression methods grows exponentially. For instance, consider a time-invariant system in Cartesian coordinates. If only positions, velocities and their first-order spatial derivatives are considered, the number of independent variables jumps from three in one dimension to 15 in three dimensions. This exponential growth of the search space greatly diminishes the efficiency of equation identification.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

A natural solution is to extend symbolic regression from discovering scalar equations to identifying tensor equations. Tensors inherently capture directional dependencies and exhibit transformation invariance (e.g. under translations, rotations and reflections), enabling more compact and physically meaningful representations of high-dimensional systems. In many physical domains, such as fluid mechanics, most physical quantities are intrinsically tensors. Therefore, the identification of tensor equations offers not only greater expressive power but also a more physically appropriate modelling framework.

At present, general-purpose methods for symbolic tensor equation discovery are still lacking. One representative method, the multidimensional GEP (M-GEP), requires prior construction of tensor and scalar bases via the Cayley–Hamilton theorem – a process that introduces case-specific bias and demands additional manual effort (Weatheritt & Sandberg 2016). Another method, the Cartesian tensor-based sparse regression, effectively identifies equations through regression techniques but shares certain limitations common to sparse regression approaches – such as constrained expressive capability and sensitivity to noise (Zhang *et al.* 2025). Besides, both methods identify non-dimensional governing equations without explicitly incorporating physical constraints during training. In addition, other methods such as rheological universal differential equations (Lennon, McKinley & Swan 2023) and sparse physics-informed discovery of empirical relations (Golden *et al.* 2023; Gurevich *et al.* 2024), incorporate translational invariance and symmetry into their frameworks.

In this work, we propose a general data-driven framework for identifying tensor equations, termed symbolic identification of tensor equations (SITE). The SITE offers a broadly applicable approach for identifying tensor governing equations across diverse domains. Inspired by M-GEP, SITE encodes tensor expressions using a host–plasmid architecture but adopts a different evolutionary strategy to enhance robustness. To address the challenges of large, redundant search spaces and the generation of physically meaningless expressions, SITE incorporates dimensional homogeneity check (DHC) during evolutionary process. Additionally, a tensor linear regression (TLR) technique is introduced to replace the conventional linear scaling step (Keijzer 2003) used in scalar symbolic regression, significantly accelerating convergence and improving numerical accuracy.

The remainder of this paper is organized as follows. Section 2 introduces the SITE methodology in detail. Section 3 presents numerical experiments on two benchmark problems to validate the robustness and accuracy of SITE. Section 4 explores the discovery of constitutive relations from molecular simulation data, and § 5 summarizes our key conclusions.

2. Methodology

This section presents the methodological design of SITE. As illustrated in figure 1, the process of symbolic identification using SITE begins with data preparation and preprocessing. A terminal library for both tensors and scalars is then constructed based on these physical quantities. Using this library and a predefined set of symbolic operators, SITE generates candidate equations by composing various symbolic structures, which are iteratively optimized through an evolutionary workflow.

We elaborate on the three core components of SITE that enable the identification of tensor equations with both physical interpretability and quantitative accuracy. The evolutionary process serves as the foundation for identifying tensor equation structures. In parallel, DHC and TLR ensure the physical interpretability and quantitative accuracy of the identified results, respectively.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

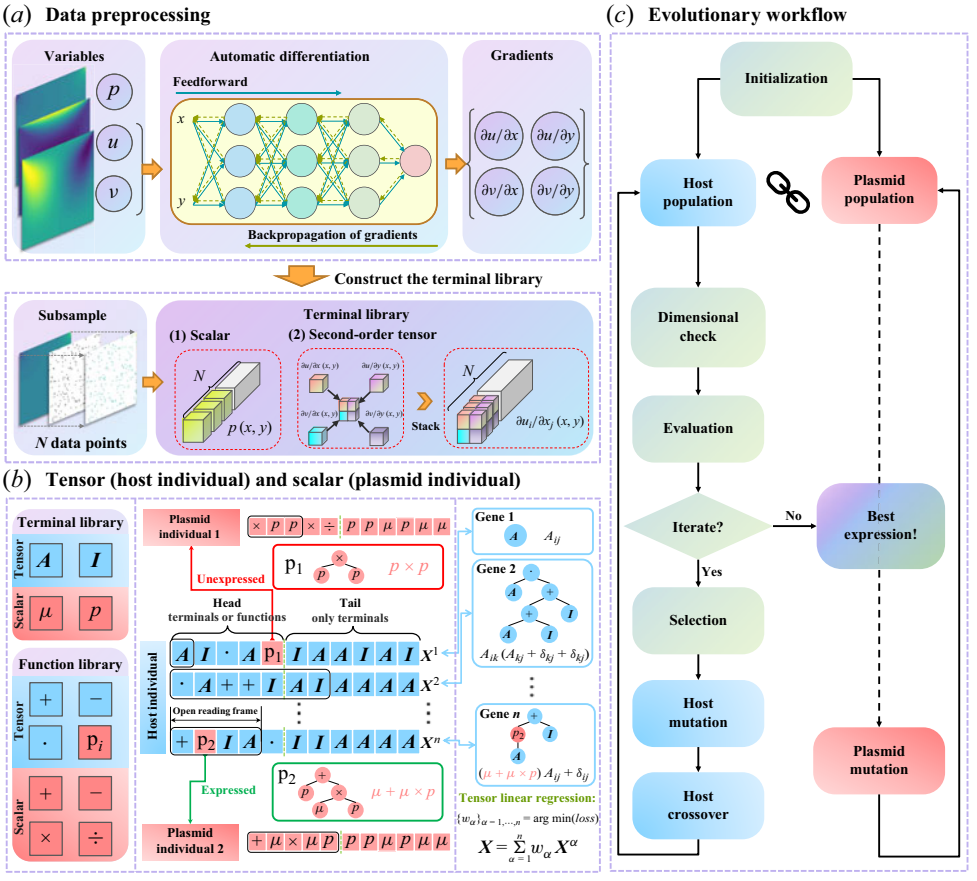


Figure 1. Overview of the SITE framework. (*a*) Data preprocessing, including the calculation of the gradients of macroscopic quantities and construction of the terminal library. (*b*) Schematic diagram of tensor (host individual) and scalar (plasmid individual). (*c*) Flowchart of the evolutionary workflow.

2.1. *Evolutionary process of SITE*

2.1.1. *Conventional GEP*

The SITE implements the evolutionary process based on the conventional GEP and extends it to handle tensor equations through a nested host–plasmid architecture. To provide the necessary context, we begin with a brief overview of the conventional GEP algorithm.

In conventional GEP, an initial population of individuals – each representing a mathematical expression – is constructed. Each individual, or chromosome, is composed of one or more linear sequences called genes. These genes are formed by combining elements from a terminal library (containing variables and constants) and a function library (containing mathematical operators). Each gene consists of a head of fixed length h, which may include functions and terminals, and a tail of length $t = h \times (a_{max}) + 1$, restricted to terminals only. Here, a_{max} means the max arity (the number of arguments a mathematical function takes) of the functions in the candidate function library. Each gene can be parsed into an ET using a deterministic decoding scheme, and the final expression represented by a chromosome is obtained by linking the outputs of multiple ETs using a predefined linking function (e.g. addition). Having multiple genes within a single chromosome enhances the representational power and exploratory capabilities.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

After the initial population is generated, the fitness of each individual in the population is evaluated. The mean relative error is a common loss function used for evaluation, which is defined as

$$\mathcal{L}(Y, \hat{Y}) = \frac{1}{N} \sum_{k=1}^{N} \left| \frac{\hat{Y}_k - Y_k}{Y_k} \right|, \tag{2.1}$$

where $\hat{Y}_k$ is the kth predicted value of all the N data points and Y_k is the ground truth.

Selection then proceeds using a hybrid strategy combining elitism and tournament selection, as designed in Ferreira (2001). Elites are retained directly, while others undergo genetic operations like mutation, transposition and crossover. For a detailed description of the genetic operators, refer to Appendix A. The evolutionary cycle repeats until either the loss of an expression falls below a predefined threshold or the maximum number of evolution generations is reached.

2.1.2. *Extension to tensor equations based on M-GEP*

We now extend the conventional GEP to handle tensor equations, focusing on second-order Cartesian tensors. The nested host–plasmid architecture used in SITE is adapted from M-GEP. The SITE enhances the evolutionary strategy and generalizes the framework to support a broader application in tensor equation identification.

Assume the general form of the tensor equation to be identified is

$$T_{ij} = F(\{\boldsymbol{a}_{ij}\}, \{\boldsymbol{x}\}), \tag{2.2}$$

where T_{ij} is the target second-order tensor, while $\boldsymbol{a}_{ij}$ and $\boldsymbol{x}$ represent a group of second-order tensors and scalars, respectively. Our goal is to identify the underlying symbolic form $F(\cdot)$ from data.

To handle mixed-order tensor operations and incorporate scalars into tensor expressions, we adopt a host–plasmid architecture in SITE. Specifically, host individuals are used to represent second-order tensor expressions, while plasmid individuals, which exist in dependency on their host individuals, are used to embed scalars into the tensor expressions.

As shown in figure 1(*b*), the host gene uses a special 'p' operator to invoke a corresponding plasmid, enabling scalar-tensor coupling. In both the terminal library and the function library, we need to prepare two parts, corresponding, respectively, to the hosts (the blue part in the diagram) and the plasmids (the red part in the diagram). The host terminal library includes candidate second-order tensors, while its function library contains tensor operators with equal input and output tensor orders, such as tensor addition, tensor multiplication and inner product, as well as the 'p' operator used to invoke embedded scalar expressions. In contrast, the plasmid terminal library contains candidate scalars, and its function library includes basic scalar operators.

When a 'p' operator appears in a host gene, it triggers a scalar multiplication between the tensor below the 'p' in the ET and the scalar expression generated by the corresponding plasmid. For example, the last gene of the host individual in figure 1(*b*) has a head of length 5 and a tail of length 6. Here, the ORF determines which elements of a gene are used to construct the ET. Starting from the head of the gene, the ORF sequentially reads elements (functions or terminals) from left to right until it naturally terminates due to the gene's structural constraints (such as the end of the tail or grammatical rules), ensuring that the generated ET is complete. The ORF and the tensor expression of this gene (X_{ij}^n) are

$$\begin{bmatrix} + & \mathrm{p}_2 & \delta_{ij} & A_{ij} \end{bmatrix}, \quad X_{ij}^n = \mathrm{p}_2(A_{ij}) + \delta_{ij}, \tag{2.3}$$

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

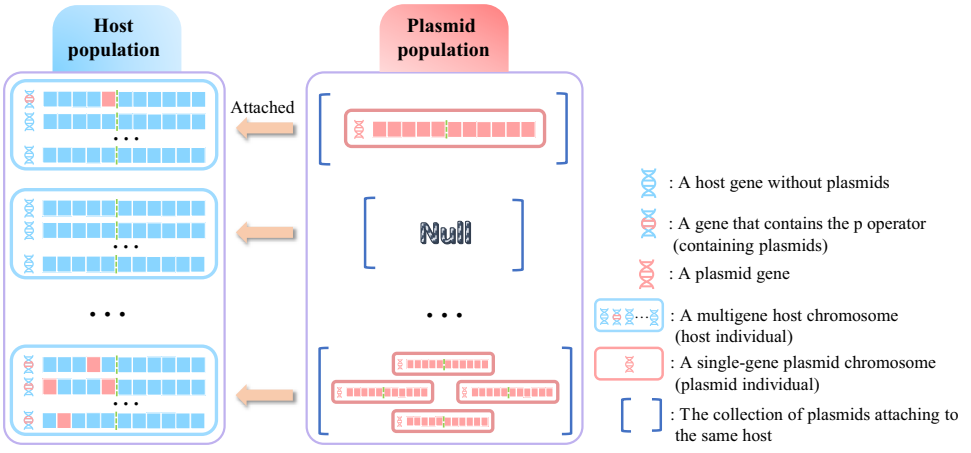


Figure 2. Schematic diagram of generating plasmid populations based on host population.

respectively, where 'p_2' will evoke the plasmid individual 2. The scalar expression represented by plasmid individual 2 is

$$\mu + (\mu \times p), \tag{2.4}$$

and the corresponding complete expression should be

$$X_{ij}^{n} = (\mu + \mu \times p)A_{ij} + \delta_{ij}. \tag{2.5}$$

In the specific implementation of SITE, the plasmid population construction is dynamic. The number of 'p' operators in a host chromosome determines how many plasmid individuals are attached to it. As illustrated in figure 2, for each 'p' operator, a plasmid individual is randomly generated at initialization, and all the plasmid individuals constitute the plasmid population. Essentially, the core evolutionary process is carried out by the host population with a fixed number of individuals, while the plasmid population is merely an appendage to the host population. The number of individuals in the plasmid population fluctuates in accordance with changes in the host population. Note that SITE retains unexpressed plasmids temporarily suppressed by genetic operations rather than destroying them. This strategy enhances robustness by retaining genetic diversity and preventing the premature loss of potentially useful subexpressions.

The evolutionary process of SITE (see figure 1*c*) is similar to that of conventional GEP. In each generation, the host population is evaluated and selected based on fitness, guiding the progression of the evolutionary process. In contrast, the plasmid population does not undergo direct evaluation or selection. Instead, these plasmid individuals serve to assist in the evaluation of the host population, indirectly contributing to the evolutionary process. After selection, the host offspring will execute both modification and crossover operations, while the plasmid population evolves solely through modification. This design retains the linkage integrity between plasmids and their hosts, thereby avoiding disruptive recombination. The SITE adopts the same stopping criterion as conventional GEP, meaning that the evolutionary process ends when the loss value falls below a prescribed threshold or the maximum number of generations has been reached.

Since the target variable changes from a scalar to a tensor, we also need to design a loss function suitable for tensor symbolic regression. In SITE, the loss function used for evaluation is defined as

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

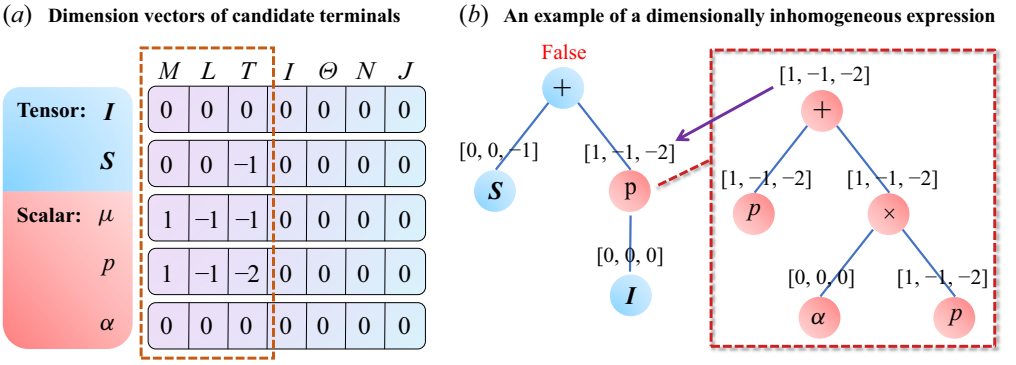


Figure 3. Schematic diagram of DVs and DHC.

$$\mathcal{L}(y_{ij}, \hat{y}_{ij}) = \sum_{i=1}^{n_{dim}} \sum_{j=1}^{n_{dim}} \frac{\|\hat{y}_{ij} - y_{ij}\|_2^2}{\|y_{ij}\|_2^2}, \tag{2.6}$$

where $y_{ij} \in \mathbb{R}^{n_{data}}$ and $\hat{y}_{ij} \in \mathbb{R}^{n_{data}}$ denote the value sequences across all data points for the (i, j)th component of the target tensor $\boldsymbol{Y}$ and the predicted tensor $\hat{\boldsymbol{Y}}$, respectively. Here $\|\hat{y}_{ij} - y_{ij}\|_2^2$ represents the squared L^2 norm of the residual corresponding to the tensor component in a specific direction, while $\|y_{ij}\|_2^2$ represents the squared L^2 norm of the target tensor component. This loss function measures the average relative error across all tensor components. After several generations of evolution, the optimal tensor expression is identified.

2.2. *DHC and seed injection strategy*

In SITE, we introduce a fundamental physical constraint – dimensional homogeneity – into the equation discovery by performing DHC. We select the seven base physical quantities of the SI-unit system (i.e. mass (M), length (L), time (T), electric current (I), thermodynamic temperature (Θ), amount of substance (N) and luminous intensity (J)) as the basic dimension. All known physical quantities' dimensions can be derived by combining the above base dimensions.

Figure 3(a) presents the dimension vectors (DVs) used for dimensional analysis, where the specific numbers represent the exponents of the corresponding dimensions. In SITE, every element in the candidate terminal library is assigned a unique DV, which serves as an intrinsic attribute of the terminal. We determine the DV at each node, as well as the resulting DV of the entire expression, by traversing the ET from the bottom up. If the expression itself involves operations that violate the dimensional homogeneity constraint as shown in figure 3(b), a summation occurs between two nodes with different DVs, the individual is immediately classified as dimensionally inhomogeneous. If the expression is dimensional homogeneous, its resulting DV is then compared with that of the target variable; if the two DVs are identical, the individual is considered to satisfy the dimensional homogeneity constraint.

In the SITE framework, the DHC is performed not only for scalar quantities but also extended to two additional cases involving tensors. First, for operations between tensors, since all components within a tensor share the same dimension, addition and inner products follow the same rules as in the scalar case. Second, for operations between tensors and scalars, SITE only involves scalar multiplication between a tensor and a scalar,

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

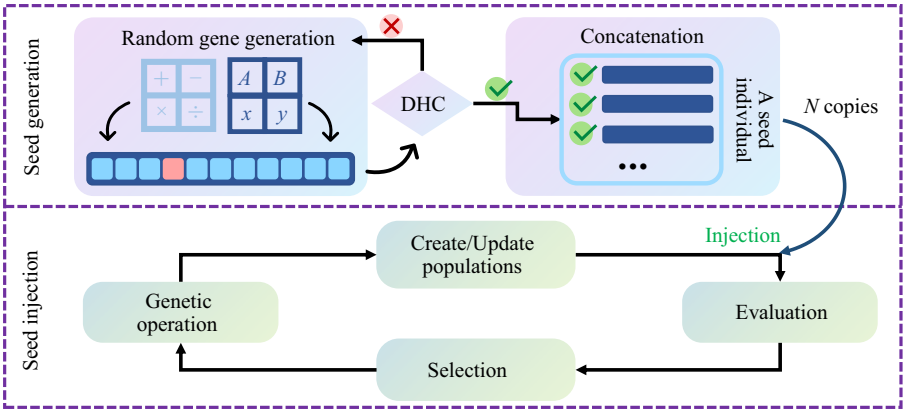


Figure 4. Schematic diagram of the seed injection strategy.

following the scalar multiplication rules. Details of DHC for scalar expressions using DVs refer to Reissmann *et al.* (2025).

In the evaluation step of the evolutionary workflow, for individuals that fail the DHC, the loss function is not applied to compute their loss value. Instead, they are assigned a sufficiently large loss, thus gradually eliminating expressions without physical meaning in the evolutionary process. The DHC acts as a strict physical constraint, which guarantees that all surviving equations satisfy dimensional homogeneity and effectively guides the evolutionary process towards physically meaningful solutions.

Note that increasing the number and length of host and plasmid genes to represent complex tensor expressions can sharply reduce the proportion of dimensionally consistent individuals, making it difficult for the population to generate valid expressions even after many generations. To mitigate this, we propose a seed injection strategy, inspired by population diversity enhancement mechanisms in evolutionary computation (Goldberg & Richardson 1987). Specifically, at fixed intervals during evolution, several host genes that individually satisfy the dimensional homogeneity constraint are manually generated and concatenated to form a single seed individual. Multiple copies of this individual are then injected into the host population. A schematic of this seed injection strategy is shown in figure 4. This strategy facilitates the emergence of dimensionally homogeneous expressions in early generations. Additionally, it helps the population escape local optima, analogous to how invasive species reshape ecological systems.

2.3. *Tensor linear regression technique*

Note that conventional GEP excels at exploring the functional form of equations but struggles with determining accurate numerical coefficients. This limitation arises because coefficient optimization in conventional GEP relies solely on the mutation of random numerical constants (RNC), a highly stochastic process that results in low efficiency in optimizing coefficients. To address this issue, the linear scaling technique (Keijzer 2003) has been introduced into conventional GEP. This approach adjusts the output of candidate expressions through a global scaling factor, allowing for a better fit to the target data. This method proves effective in scalar contexts, as each individual adjusts all numerical values with a single scaling factor.

In tensor symbolic regression, using a single global scaling factor cannot effectively fit all components of a predicted tensor expression. To address this, we introduce a TLR technique, which assigns an independent coefficient to each gene and fits the

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

predicted tensor expression to the target through a linear combination of tensor features. This approach enhances coefficient optimization efficiency and enables implicit feature selection by assigning different importance to different genes.

The TLR task in SITE is formulated as follows: for a given target tensor Y_{ij} and a set of tensor features $((X_{ij})_1 \quad \cdots \quad (X_{ij})_{n_{gene}})$ expressed by host genes, where $i, j \in \{1, 2, \cdots, n_{dim}\}$ are tensor indices corresponding to the physical spatial dimensions and n_{gene} denotes the number of host genes, we need to prepare their values at all n_{data} points. Given the values of all tensors at each data point, we can define the target tensor as

$$y_{ij} = \begin{pmatrix} (Y_{ij})_1 \\ \vdots \\ (Y_{ij})_{n_{data}} \end{pmatrix} \tag{2.7}$$

and the set of tensor features as

$$\Theta_{ij} = \begin{pmatrix} (X_{ij})_{11} & \cdots & (X_{ij})_{1n_{gene}} \\ \vdots & \ddots & \vdots \\ (X_{ij})_{n_{data}1} & \cdots & (X_{ij})_{n_{data}n_{gene}} \end{pmatrix}. \tag{2.8}$$

Our objective is to seek a coefficient vector $\boldsymbol{w}$ such that the predicted $\hat{y}_{ij}$ fits the target y_{ij} and minimizes the loss function. The loss function using in evaluation has been defined as (2.6). Here, the complete loss function in TLR is

$$\mathcal{L}(y_{ij}, \hat{y}_{ij}) = \sum_{i=1}^{n_{dim}} \sum_{j=1}^{n_{dim}} \frac{(\hat{y}_{ij} - y_{ij})^T (\hat{y}_{ij} - y_{ij})}{y_{ij}{}^T y_{ij}}. \tag{2.9}$$

The summation in (2.9) means that every component of a tensor is taken into account. The predicted $\hat{y}_{ij}$ is obtained by a weighted linear combination of n_{gene} features:

$$\hat{y}_{ij} = \begin{pmatrix} (\hat{Y}_{ij})_1 \\ \vdots \\ (\hat{Y}_{ij})_{n_{data}} \end{pmatrix} = \Theta_{ij}\boldsymbol{w} = \begin{pmatrix} (X_{ij})_{11} & \cdots & (X_{ij})_{1n_{gene}} \\ \vdots & \ddots & \vdots \\ (X_{ij})_{n_{data}1} & \cdots & (X_{ij})_{n_{data}n_{gene}} \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_{n_{gene}} \end{pmatrix}. \tag{2.10}$$

The coefficient vector will be determined from

$$\boldsymbol{w} = \arg\min \mathcal{L}(y_{ij}, \hat{y}_{ij}). \tag{2.11}$$

By taking the partial derivative of the loss function and setting it to zero, an analytical solution for the coefficient vector $\boldsymbol{w}$ can be obtained as

$$\boldsymbol{w} = \left(\sum_{i=1}^{n_{dim}} \sum_{j=1}^{n_{dim}} \frac{\Theta_{ij}^T \Theta_{ij}}{y_{ij}^T y_{ij}} \right)^{-1} \left(\sum_{i=1}^{n_{dim}} \sum_{j=1}^{n_{dim}} \frac{\Theta_{ij}^T y_{ij}}{y_{ij}^T y_{ij}} \right). \tag{2.12}$$

For the analytical form of the coefficient vector to be valid, there is a prerequisite condition that the coefficient matrix must be non-singular. This condition is essentially equivalent to the features in the feature set Θ_{ij} being linearly independent. Therefore, to apply (2.12) for solving the coefficient vector, we need to first perform feature selection on the feature set, selecting all the linearly independent features to construct a reduced feature set Θ'_{ij} and ensuring that it is column-linearly independent.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

With the coefficient vector $\boldsymbol{w}$, we change the predicted expression from the summation of each feature tensor to (2.10) before evaluation. The entire TLR process takes place after DHC and before evaluating the final loss, serving as the final optimization before an individual is evaluated. The coefficient vector of each individual is independent of its sub-ETs and is computed only before evaluation without participating in any subsequent genetic operations.

3. Validation on benchmark problems

We validate and demonstrate the capabilities of SITE in identifying tensor equations using two benchmark scenarios. These two cases correspond to well-established models in mathematical physics, namely the Maxwell stress tensor and the Reynolds-stress transport equation. Based on these benchmarks, we conduct a series of studies, including ablation experiments on key SITE components (TLR and DHC), comparative evaluations against state-of-the-art method, and tests of the robustness to data noise.

For benchmark problems with a known target equation, correctness can be assessed in an objective manner. Specifically, an identified equation is regarded as correct only if all its terms exactly match those in the target equation (Du *et al.* 2024). Minor deviations due to numerical fitting errors are acceptable. Based on this criterion, the performance of SITE is quantified by a success rate, defined as the ratio of successful identifications to the total number of independent runs (five in this study). For both benchmark cases, we report the corresponding success rates in the results that follow. If the success rate is 100 %, the coefficients of the final reported equation are obtained by averaging the results of five independent runs. If the success rate is under 100 % (not 0 %), we take the successfully identified equations and perform the same average processing to obtain the reported equation. If the correct equation is not found in the five independent experiments, we report the optimal equation among them (the equation with the lowest loss).

3.1. *Maxwell stress tensor*

We first consider the Maxwell stress tensor in the context of electromagnetism. The Maxwell stress tensor, typically defined as

$$T_{ij} = \varepsilon_0 \left(E_i E_j - \frac{1}{2} E_k E_k \delta_{ij} \right) + \frac{1}{\mu_0} \left(B_i B_j - \frac{1}{2} B_k B_k \delta_{ij} \right), \tag{3.1}$$

describes the distribution of electromagnetic momentum and forces within a medium. Here, E_i and B_i are the components of the electric and magnetic fields, respectively, ε_0 is the vacuum permittivity and μ_0 is the vacuum permeability.

A three-dimensional cube with sinusoidally varying fields is used to evaluate the behaviour governed by (3.1). The electric field is given by

$$E(x, y, z) = [E_0 \sin(\pi x), E_0 \cos(\pi y), E_0 \sin(\pi z)], \tag{3.2}$$

and the magnetic field is given by

$$B(x, y, z) = [B_0 \cos(\pi x), B_0 \sin(\pi y), B_0 \cos(\pi z)], \tag{3.3}$$

where $E_0 = 10^6$ V m^{-1} and $B_0 = 10^{-3}$ T are fixed amplitudes. To construct the dataset, 150 spatial points are randomly sampled within the cube $(-1 \leqslant x, y, z \leqslant 1)$, and the corresponding Maxwell stress tensor is computed analytically using (3.1). The schematic diagram of this case is shown in figure 5(*a*).

In this case, the target variable is defined as T_{ij}. For tensors, the candidate function library and the candidate terminal library are defined as $\{+, -, \cdot, \mathrm{p}\}$ and $\{E_i E_j, B_i B_j, \delta_{ij}\}$,

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

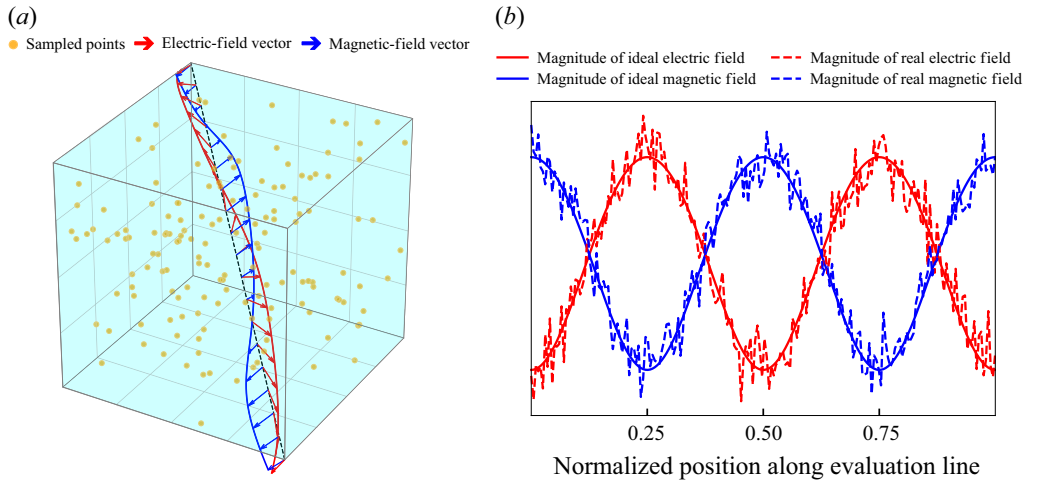


Figure 5. (*a*) Schematic diagram of the electric field, the magnetic field and sampled points. For clarity of illustration, only the vectors along the guiding dashed line are shown. (*b*) Trend lines of the electromagnetic field magnitude at the points along the guiding line in (*a*) before and after adding 5 % Gaussian noise. The solid line represents the ideal setting, and the dashed line represents the real situation.

respectively, while for scalars, the candidate function library and the candidate terminal library are defined as $\{+, -, \times, \div\}$ and $\{E_i E_i, B_i B_i, \varepsilon_0, \mu_0\}$. The candidate terminal libraries of different tests are exactly the same.

To highlight the contribution of TLR to improving coefficient optimization efficiency, we conduct comparative experiments under three configurations: (i) using only RNC, (ii) using only TLR and (iii) combining TLR with RNC. When TLR is not employed, RNC must be introduced into the scalar candidate terminal library to form numerical constants in expressions. These RNC are randomly sampled floating-point numbers within the range $[-10, 10]$. The hyperparameter settings for the comparative experiments are basically the same, except that additional genetic operations need to be performed on RNC when they are introduced (see Appendix A). Detailed parameter settings for the SITE algorithm, such as the maximum number of evolutionary generations and the number of host individuals, are provided in Appendix B. The derived equations for this test, along with the number of iterations and the central processing unit (CPU) runtime (measured on a system equipped with a 13th Gen Intel Core i9-13900K CPU at 3.00 GHz, 32 GB RAM and running a 64-bit operating system), are summarized in table 1.

It can be seen from table 1 that regardless of whether TLR is used, SITE is capable of discovering complex tensor equations from the data. However, the success rate shows a clear difference across configurations. Without TLR, the framework fails to identify a completely correct equation in all five runs, yielding a success rate of 0/5 even after the maximum number of evolutionary generations. Specifically, the coefficients of the discovered terms are incorrect and the combination of $B_k B_k \delta_{ij}$ and $(1/\mu_0)$ is not identified. This is because optimization based on RNC relies only on completely random mutations to alter the numerical coefficients in the expression. In contrast, when TLR is used, SITE consistently identifies the exact target equation in all five runs, achieving a success rate of 5/5 and substantially reducing the number of iterations required. It is evident that TLR substantially improves the efficiency and accuracy of tensor equation identification. Moreover, we tested a third configuration where RNC are removed and only TLR is used. The SITE achieves the same accuracy within 182 iterations on average, demonstrating

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

TLR used	RNC used	Derived equation	Average iterations (time)	Success rate
No	Yes	$T_{ij} = \varepsilon_0 E_i E_j - 0.529\varepsilon_0 E_k E_k \delta_{ij} + \frac{1}{\mu_0} B_i B_j$	500 (466 s)	0/5
Yes	No	$T_{ij} = \varepsilon_0 E_i E_j - 0.5\varepsilon_0 E_k E_k \delta_{ij} + \frac{1}{\mu_0} B_i B_j - 0.5\frac{1}{\mu_0} B_k B_k \delta_{ij}$	182 (188 s)	5/5
Yes	Yes	$T_{ij} = \varepsilon_0 E_i E_j - 0.5\varepsilon_0 E_k E_k \delta_{ij} + \frac{1}{\mu_0} B_i B_j - 0.5\frac{1}{\mu_0} B_k B_k \delta_{ij}$	65 (58 s)	5/5

Table 1. Performance comparison of SITE using different coefficient optimization configurations.

Method	Derived equation	Average iterations	Success rate
SITE	$T_{ij} = \varepsilon_0 E_i E_j - 0.5\varepsilon_0 E_k E_k \delta_{ij} + \frac{1}{\mu_0} B_i B_j - 0.5\frac{1}{\mu_0} B_k B_k \delta_{ij}$	182	5/5
M-GEP	$T_{ij} = \varepsilon_0 E_i E_j - 0.403\varepsilon_0 E_k E_k \delta_{ij} + 0.916\frac{1}{\mu_0} B_i B_j - 0.749\frac{1}{\mu_0} B_k B_k \delta_{ij}$	500	3/5*

Table 2. Performance comparison between SITE and M-GEP.

*Note that one of the three success runs does not explicitly include ε_0 and μ_0, but the approximate coefficients were numerically consistent with their values. Considering that M-GEP uses non-dimensionalization to preprocess the data instead of using DHCs to constrain the evolutionary process, we think this equation with low discrepancy can also be regarded as successfully identifying the target equation.

that TLR alone is sufficient to achieve both high accuracy without relying on RNC. Since the mingling of RNC with symbolic terminals sometimes may hinder the generation of concise expressions, all cases in this article use only TLR to optimize coefficients unless otherwise specified.

To further benchmark SITE against the state-of-the-art method, we compare it with the latest EVE-3 implementation of M-GEP using the same dataset as above. As shown in table 2, M-GEP achieves a success rate of 3/5 with an average of 500 iterations. In one of these successful runs, the discovered equation does not explicitly include ε_0 and μ_0, yet the approximate coefficients are numerically consistent with their physical values. Since M-GEP relies on non-dimensionalization for preprocessing instead of enforcing dimensional homogeneity as in SITE, such cases are still regarded as successful identifications. In contrast, SITE consistently identifies the exact Maxwell stress tensor equation in all five runs, with an average of only 182 iterations. These results highlight that SITE not only ensures higher reliability (success rate 5/5 versus 3/5 for M-GEP), but also achieves better efficiency.

To evaluate the robustness of SITE to Gaussian noise, we test its performance under three noise levels: 5 %, 10 % and 20 %. Specifically, as illustrated as figure 5(*b*), Gaussian noise is first added to the electric and magnetic field strengths to generate perturbed physical quantities. Then, Maxwell stress tensors are computed from these noisy fields and further perturbed by Gaussian noise of the same level. These noisy data are used for the subsequent tests. Here, we use both RNC and TLR here to get results quickly.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

Noise level	Derived equation	Error	Success rate
5 %	$T_{ij} = 0.999\,\varepsilon_0 E_i E_j - 0.500\,\varepsilon_0 E_k E_k \delta_{ij} + 1.007\,\frac{1}{\mu_0} B_i B_j - 0.499\,\frac{1}{\mu_0} B_k B_k \delta_{ij}$	0.25 ± 0.27 %	5/5
10 %	$T_{ij} = 0.998\,\varepsilon_0 E_i E_j - 0.500\,\varepsilon_0 E_k E_k \delta_{ij} + 1.014\,\frac{1}{\mu_0} B_i B_j - 0.501\,\frac{1}{\mu_0} B_k B_k \delta_{ij}$	0.45 ± 0.55 %	5/5
20 %	$T_{ij} = 0.997\,\varepsilon_0 E_i E_j - 0.500\,\varepsilon_0 E_k E_k \delta_{ij} + 1.032\,\frac{1}{\mu_0} B_i B_j - 0.513\,\frac{1}{\mu_0} B_k B_k \delta_{ij}$	1.53 ± 1.40 %	5/5

Table 3. Equations identified by SITE under different levels of Gaussian noise (5 %, 10 % and 20 %).

The derived equations with different levels of noise are listed in table 3. The error in the table is defined as the average relative error of the identified non-zero coefficients with respect to the ground truth. When Gaussian noise is added, the coefficients in the equation identified by SITE deviate increasingly from the ground truth as the noise level increases, and the error grows almost linearly with the noise level. Even with up to 20 % Gaussian noise, SITE still successfully identifies the complete structure of the expression, although there are some deviations in the coefficients. This proves that the framework has great robustness against artificial noise.

3.2. *Reynolds-stress transport equation*

In many physical systems, the evolution of tensors is governed by time-dependent differential equations. To evaluate the capability of SITE to identify such dynamic tensor equations, we consider the following general form of a time-dependent tensor evolution equation:

$$\frac{\mathrm{d}T_{ij}}{\mathrm{d}t} = F(\{\boldsymbol{a}_{ij}\}, \{\boldsymbol{x}\}) \tag{3.4}$$

where T_{ij} is a second-order tensor, while $\boldsymbol{a}_{ij}$ and $\boldsymbol{x}$ represent a group of second-order tensors and scalars, respectively. Note that when spatial derivatives of the tensor T_{ij} appear on the right-hand side of the equation, it becomes a PDE.

To generate data for the discovery of this kind of tensor equation, simulations are performed for three-dimensional compressible decaying homogeneous isotropic turbulence. The simulation employs a uniform grid of $32 \times 32 \times 32$ cells, a backward Euler time advancement scheme with a time step of 1×10^{-7} s, and a total simulation time of 4×10^{-6} s. Argon gas is used as the working fluid, with initial conditions of 273.15 K temperature, 100 000 Pa pressure and zero mean velocity. The initial turbulent kinetic energy is set to 4265.9 $\mathrm{m^2\,s^{-2}}$, corresponding to an initial turbulent Mach number of 0.3. The flow is initialized following the procedure of Ma *et al.* (2024*a*), and computed using the open-source solver OpenFOAM with periodic boundary conditions (Jasak, Jemcov & Tukovic 2007) In our simulation, a modelled Reynolds stress transport equation is solved in OpenFOAM using the Launder–Reece–Rodi model (Launder, Reece & Rodi 1975). Therefore, the target governing equation takes the form

$$\frac{\mathrm{D}(\rho R_{ij})}{\mathrm{D}t} = \rho P_{ij} - 1.8\frac{\varepsilon}{k}\rho R^*_{ij} - 0.6\rho P^*_{ij} - \frac{2}{3}\rho\varepsilon\delta_{ij} + D_{ij}, \tag{3.5}$$

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

Data	Derived equation	Success rate
Clean dataset	$\frac{D(\rho R_{ij})}{Dt} = 1.000\rho P_{ij} - 1.800\frac{\varepsilon}{k}\rho R^*_{ij} - 0.600\rho P^*_{ij} - 0.667\rho\varepsilon\delta_{ij} + 1.000 D_{ij}$	5/5
Noisy dataset	$\frac{D(\rho R_{ij})}{Dt} = 1.018\rho P_{ij} - 1.729\frac{\varepsilon}{k}\rho R^*_{ij} - 0.618\rho P^*_{ij} - 0.566\rho\varepsilon\delta_{ij} + 0.932 D_{ij}$	4/5

Table 4. Equations identified by SITE from clean dataset and noisy dataset.

where the Reynolds stress is defined as $R_{ij} = \overline{u'_i u'_j}$, ρ is the density, k the turbulent kinetic energy and ε the dissipation rate. Here, $(\cdot)^*$ denotes the trace-free part of a tensor and $\mathrm{D}(\cdot)/\mathrm{D}t$ the material derivative. In this formulation, the production term P_{ij} is computed explicitly from the resolved mean flow, while the remaining terms are modelled slow and rapid pressure-strain correlations, dissipation and diffusion terms, respectively. All tensors and scalars in (3.5) are directly computed and output by the solver.

In the previous case, we have evaluated the robustness of the framework under Gaussian noise. In this case, we further conduct more challenging noise robustness tests – all the physical quantities used are derived from the computational fluid dynamics simulation results. Specifically, two representative datasets are constructed: (i) a clean dataset, where the target material derivative is assembled directly from the right-hand side of (3.5); and (ii) a noisy dataset, where the target tensor is taken from solver outputs and thus contains discretization errors. Both datasets consist of 200 randomly sampled spatiotemporal points, and remain fixed across independent SITE runs.

The target variable is defined as $(\mathrm{D}(\rho R_{ij})/\mathrm{D}t)$. For tensors, the candidate function library and the candidate terminal library are defined as $\{+, -, \cdot, (\cdot)^*, \mathrm{p}\}$ and $\{P_{ij}, R_{ij}, D_{ij}, \delta_{ij}\}$, respectively, while for scalars, the candidate function library and the candidate terminal library are defined as $\{+, -, \times, \div\}$ and $\{\rho, k, \varepsilon\}$.

As shown in table 4, SITE consistently recovers the correct equation from the clean dataset in all five independent runs, without any coefficient errors. For the noisy dataset, four out of five runs yield the correct equation structure with comparable accuracy. In the unsuccessful run, an additional traceless diffusion term D^*_{ij} appears with a small coefficient of 0.03, which is one order of magnitude lower than those of the dominant terms. This spurious term is attributed to data noise rather than physical relevance, indicating that while noise may occasionally induce weak redundant terms, the main physical effects are still robustly captured.

Beyond the above baseline results, we further challenge SITE on the clean dataset to test its expressive capability. Methodologically, SITE's candidate tensor function library is designed to be both expressive and flexible, having the ability to contain operators such as trace-free and transpose. To intensify the test, we replace the turbulent production term P_{ij} with the raw velocity gradient tensor $\partial u_i/\partial x_j$, where

$$P_{ij} = -R_{ij}\frac{\partial u_j}{\partial x_k} - R_{jk}\frac{\partial u_i}{\partial x_k}. \tag{3.6}$$

In this setting, SITE is required not only to recover all terms in the complete Reynolds-stress transport equation, but also to reconstruct P_{ij} from its lower-level components. Remarkably, the exact target equation was successfully identified within 1000 generations, highlighting the strong expressive capability of SITE.

For this problem, a direct comparison with M-GEP is not feasible. The reason is that M-GEP requires all necessary tensors to be explicitly prespecified in advance – for

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

DHC used	Average iterations	Average CPU runtime	Success rate
No	146	459 s	1/5
Yes	126	92 s	5/5

Table 5. Performance comparison of SITE with or without DHC on the clean dataset.

example, the trace-free tensors must be provided beforehand – whereas in our setting we allow the trace-free operator itself to be included as a candidate symbol in the function library, letting SITE autonomously select and construct the appropriate symbolic structures.

Additionally, we conduct an ablation study on the role of DHC using the clean dataset here. Table 5 summarizes the results for SITE with and without DHC. With DHC enabled, SITE consistently identifies the correct target equation in all five independent runs. In contrast, under our strict success criterion, the version without DHC produces the exact target equation only once out of five runs; although in the remaining runs it still achieves very low loss values by leveraging TLR, the resulting equations include spurious terms such as δ_{ij} and $(((\rho\varepsilon^2)/2k\varepsilon + k^2)R_{ij})$, which are physically uninterpretable. In terms of efficiency, introducing DHC significantly reduces the number of non-physical candidate combinations that require evaluation, leading to fewer iterations per run and a markedly lower CPU runtime. These results confirm that DHC not only enhances the physical interpretability of the identified equations but also reduces the effective search space, thereby improving computational efficiency.

4. Equation identification based on molecular simulation

We now apply SITE to high-fidelity data generated via direct simulation Monte Carlo (DSMC). Unlike the data of benchmark cases in § 3, the DSMC datasets are generated directly from microscopic particle interactions, without embedding any macroscopic governing equations such as the Navier–Stokes equations or predefined constitutive relations. Our goal is to demonstrate that SITE can discover the macroscopic tensor constitutive relation from molecular simulation data generated without imposing any macroscopic constitutive assumptions. Additionally, we evaluate SITE's ability to identify the corresponding macroscopic forms under two distinct flow conditions and assess its robustness in capturing small-magnitude terms.

4.1. *Data generation and preprocessing*

In this section, we employ the DSMC method (Bird 1994) to generate flow field data, following the approach in Zhang & Ma (2020) and Ma *et al.* (2024*b*). Since the aim of this section is to discover tensor macroscopic governing equations of fluid dynamics, it is crucial that the data be independent of macroscopic modelling assumptions. The DSMC method fulfils this requirement by simulating particle dynamics without reliance on predefined continuum-level equations such as the Navier–Stokes equations. In DSMC, each simulation molecule statistically represents a large number of real molecules. From these molecular-level dynamics, macroscopic physical quantities are obtained via statistical averaging. For instance, the velocity field (v_i) and the pressure tensor field (p_{ij}) are computed as

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

$$v_i = \frac{1}{N_p}\sum_{cell} c_i, \quad p_{ij} = \frac{mF}{V_{cell}}\left(\sum_{cell} c_i c_j - N_p v_i v_j\right), \tag{4.1}$$

where N_p is the number of simulation molecules in the sampling cell, c_i is the velocity of the ith particle, m is the molecular mass and F is the simulation ratio in DSMC. With the data generated from DSMC, we can construct a mapping from spatial coordinates to various macroscopic variables. As illustrated in figure 1(*a*), at each given data point (x, y), we associate corresponding physical quantities such as density $\rho(x, y)$, horizontal velocity $u(x, y)$ and vertical velocity $v(x, y)$.

In our study, two primary sources of noise affect the accuracy of equation discovery. One is the numerical noise arising from the computation of spatial gradients, and the other is the statistical noise inherent in particle-based methods such as DSMC. To mitigate the first type of noise, we apply the neural-network-based automatic differentiation technique to calculate gradients of physical variables. Specifically, an artificial neural network is trained to map spatial coordinates to macroscopic variables, and gradients of arbitrary order are obtained by backpropagating through the network. This allows accurate evaluation of spatial derivatives at any point in the domain. As for the statistical noise introduced by particle simulation, we can quantitatively estimate how the statistical error changes with the number of statistical samples (Hadjiconstantinou *et al.* 2003) with

$$E_Q = \frac{1}{\sqrt{N_0 M}}\frac{\sqrt{\langle Q^2\rangle}}{|\langle Q\rangle|}, \tag{4.2}$$

where E_Q represents the statistical error of quantity Q, $\langle\cdot\rangle$ means the statistical mean value, N_0 is the number of particles in a statistical cell and M is the number of independent samples. In our study, we set the $N_0 M$ of our simulation to 4×10^9, which means the statistical error of p_{ij} is around 10^{-5}. An error of this magnitude is acceptable for our experiments.

We simulate lid-driven cavity flows using the DSMC method described above. The simulated domain is a square cavity with a side length of $L = 1$ m, filled with argon gas. All four walls maintain at a constant temperature of 273.15 K. The bottom, left- and right-hand walls are stationary, while the top wall (lid) moves horizontally at a constant speed. The number of grids is set to 400×400. Each grid cell provides DSMC statistical outputs of the horizontal and vertical velocities, the four components of pressure tensor, the temperature and the static pressure. Two flow conditions are considered by setting the lid velocity to different values. In the first case, the lid speed is set to 50 m s^{-1}, corresponding to a low-speed, nearly incompressible flow. In the second case, the lid velocity is increased to 337 m s^{-1}, resulting in a transonic, compressible flow. The DSMC simulations employ the variable hard sphere model, in which the dynamic viscosity varies with temperature according to

$$\mu = \mu_{ref}\left(\frac{T}{T_{ref}}\right)^{\omega} \tag{4.3}$$

where $\mu_{ref} = 2.117 \times 10^{-5}\,\mathrm{kg(m\,s)^{-1}}$ denotes the reference viscosity coefficient, $T_{ref} = 273.15$ K denotes the reference temperature, and $\omega = 0.81$ denotes the viscosity exponent. The global Knudsen number in both cases is fixed at $Kn = 0.005$.

To enable the computation of velocity gradients, we trained neural networks to learn the mapping from spatial coordinates to the two velocity components within the flow field. Specifically, we used all 400×400 data points from the computational domain as the training set. The mapping was modelled using a fully connected feedforward

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

neural network comprising eight hidden layers, each with 60 neurons and tanh activation functions. The training process was carried out in two stages. First, we used the Adam optimizer (Kingma & Ba 2017) for 20 000 iterations to achieve a reasonable initial solution. Then, the optimization was refined using the L-BFGS-B (Byrd *et al.* 1995) optimization until convergence. This hybrid optimization strategy ensures both stability and precision in capturing smooth and differentiable velocity fields suitable for gradient extraction via automatic differentiation.

4.2. *Constitutive equation discovery under different flow conditions*

With the DSMC datasets described in § 4.1, we now apply SITE to identify the constitutive equations under both flow conditions.

In constructing the candidate function library, we introduced a set of basic tensor and scalar operators. As for the candidate terminal library, special consideration was given to the physical relevance of constitutive modelling. Since constitutive relation of pressure tensor reflects the transport of momentum in the flow field – and such transport is primarily driven by velocity gradients – we included physical quantities related to the velocity gradient tensor. The velocity gradient tensor $D_{ij} = \partial u_i/\partial x_j$ can be decomposed into a symmetric and an antisymmetric part, given as

$$D_{ij} = S_{ij} + \Omega_{ij}, \tag{4.4}$$

where $S_{ij} = (1/2)(D_{ij} + D_{ji})$ is the symmetric strain-rate tensor, and $\Omega_{ij} = (1/2)(D_{ij} - D_{ji})$ is the antisymmetric rotation-rate tensor. In both cases, the target variable is defined as p_{ij}. For tensors, the candidate function library is defined as $\{+, -, \cdot, \mathrm{p}\}$ and $\{S_{ij}, \delta_{ij}\}$, respectively, while for scalars, the candidate function library and the candidate terminal library are defined as $\{+, -, \times, \div\}$ and $\{D_{ii}, p, \mu\}$. For a detailed description of the construction of the candidate term library used in this study, please refer to Appendix C.

The last step before applying SITE is subsampling the dataset. To mitigate the influence of numerical noise when computing gradients near the boundaries, we restricted the sampling region to the central portion of the domain – specifically, the 20 % to 80 % range along both spatial directions. From this subregion, 200 data points were randomly selected and used for symbolic regression and equation discovery. Figure 6 illustrates the distribution of velocity magnitude (defined as $\sqrt{u^2 + v^2}$) across the domain in the compressible cavity flow case, along with a schematic representation of the sampled data points used for training. This selection strategy ensures that the discovered equations are based on high-quality gradient information. We exclude regions where boundary effects or numerical errors may distort the constitutive relation, while still retaining a sufficiently large and diverse dataset. It has been verified that the dominant terms identified by SITE are insensitive to the precise cutoff as long as noisy near-wall data are excluded. This strategy is a practical guideline. For other problems, a similar strategy can be adopted by first assessing the reliability of gradient information and then subsampling a central region large enough to capture the dominant physical effects.

For reference, the complete Newtonian constitutive relation is given by

$$p_{ij} = -2\mu S_{ij} + \frac{2}{3}\mu D_{kk}\delta_{ij} + p\delta_{ij}. \tag{4.5}$$

This form can be derived under the continuum hypothesis. The first term is the shear stress term, which arises from the internal friction caused by the relative sliding (shear deformation) between fluid layers, corresponding to the anisotropic behaviour of shear stress. The second term is the compressible term, accounting for the viscous

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

Lid velocity (m/s)	Derived equation
50	$p_{ij} = -1.909\mu S_{ij} + 1.000 p\delta_{ij}$
337	$p_{ij} = -1.956\mu S_{ij} + 0.683\mu D_{kk}\delta_{ij} + 1.000 p\delta_{ij}$

Table 6. Identified equations of cavity flow.

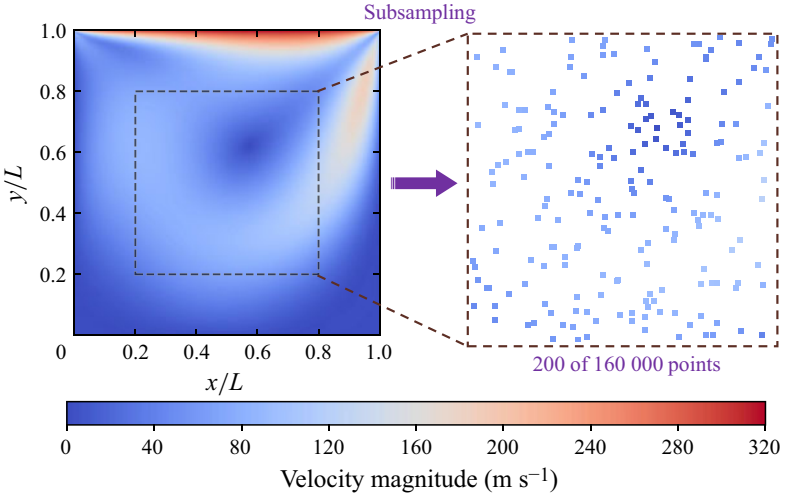


Figure 6. Distribution of velocity magnitude in the compressible cavity flow case and the spatial locations of sampled data points used for SITE.

resistance generated during volumetric changes. This term is typically negligible in incompressible flows where the velocity divergence is close to zero. The third term is the static pressure term, representing the isotropic compressive stress imposed by the thermodynamic pressure, which is independent of the flow state. In our DSMC simulations, the macroscopic constitutive relation is not imposed *a priori*. Since the global Knudsen number is small in this case, it is reasonable to expect that the effective constitutive relation should be close to Newtonian, but this does not imply that the 'ground truth' is (4.5). This is fundamentally different from a benchmark case. In this section, the success of equation identification is not judged by comparison with the Newtonian constitutive form, but rather by the objective metric of test error (refer to Appendix D for details). The reported identified equations are constructed from the results of five independent runs. Specifically, we only keep the common terms that appear across these runs, and average their coefficients to form the reported equation.

As shown in table 6, we summarize the constitutive equations discovered in the two test cases. The identified constitutive equations vary with the lid velocity, reflecting the features of different flow conditions. At the lower velocity, the discovered relation contains only the shear stress term and the static pressure term, which is consistent with the expectation that compressibility effects are weak when the flow remains nearly incompressible. At the higher velocity, an additional term proportional to $\mu D_{kk}\delta_{ij}$ emerges. This term accounts for the resistance associated with compressibility and indicates that the framework successfully detects flow-dependent physical contributions.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

It is also noteworthy that the coefficients identified by SITE exhibit slight deviations from those in the classical Newtonian constitutive model. These discrepancies may be attributed to the fact that the simulated flows are not strictly within the continuum regime. Locally, some features of slip flow may appear. Based on the Knudsen number (Kn), flow regimes can be classified into continuum ($Kn < 0.001$), slip ($0.001 < Kn < 0.1$), transition ($0.1 < Kn < 10$) and free-molecular ($Kn > 10$) regimes (Reese, Gallis & Lockerby 2003). Since molecular simulations are computationally expensive in the fully continuum regime, we selected a global $Kn = 0.005$ to balance accuracy and efficiency. Consequently, local deviations from continuum assumptions may still exist.

4.3. *Discussion on magnitude disparities*

A key challenge in data-driven modelling lies in the simultaneous identification of terms with vastly different magnitudes from noisy data. In our compressible cavity flow case, this issue becomes particularly evident. The off-diagonal components, p_{xy} and p_{yx}, are governed solely by shear deformation and is thus dominated by the symmetric part of the velocity gradient tensor. Consequently, identifying S_{ij}-related terms in this case is relatively straightforward. In contrast, the diagonal components of the pressure tensor tensor are composed of contributions from shear stress term, compressible term (linked to $\nabla \cdot \boldsymbol{u}$) and static pressure p. Among these, the compressible term typically has the smallest magnitude in the current case (see figure 7).

In this case, the problem can be simplified by subtracting the pressure to isolate the trace-free parts $\sigma_{ij} = p_{ij} - p\delta_{ij}$. However, to rigorously evaluate whether SITE can discover small-magnitude terms amid much larger dominant contributions, we deliberately choose to identify the complete constitutive relation of p_{ij}. The results in § 4.2 have shown that SITE is capable of reliably identifying the compressible term under such challenging conditions. This highlights the framework's robustness to large disparities in variable magnitudes.

This robustness arises from SITE's design principles that explicitly address magnitude disparities without distorting physical significance. Compared with sparse regression-based methods which relies on standardized feature spaces and regularization to suppress terms with small numerical contributions, SITE operates directly in the native physical scale and preserves candidate terms through evolutionary mechanisms. Subexpressions such as μD_{ii} are retained and re-evaluated across generations, even if their immediate numerical contribution is minimal.

In addition to the overall magnitude disparity between different terms, another challenge specific to tensor equation identification lies in the magnitude disparity of different components within a single tensor. Unlike scalar regression tasks where each feature corresponds to a single physical quantity, tensor modelling requires treating entire tensor structures as regression targets. However, the magnitudes of different components within a tensor (e.g. the diagonal and off-diagonal terms in p_{ij}) can vary significantly. This disparity can bias the learning process if not properly addressed. In SITE, this issue is mitigated by the design of optimization objective. The loss function (2.6) incorporates a normalization scheme that rescales each component individually. This ensures that each tensor component contributes equally to the total loss, regardless of its absolute magnitude, thereby improving the robustness of coefficient identification.

5. Conclusion

This study introduces SITE, a symbolic regression framework tailored to discover governing equations involving second-order tensors. The SITE addresses key limitations

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

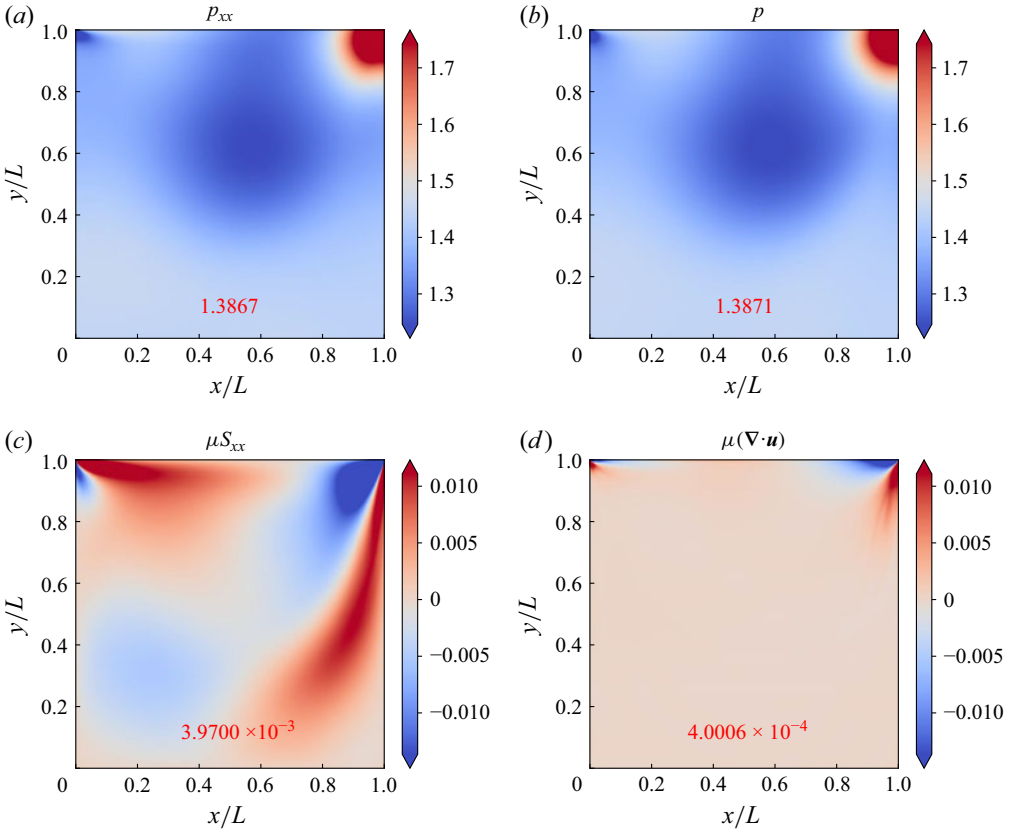


Figure 7. Distributions of the constitutive terms in the xx-direction of the compressible cavity flow case. (*a*) Pressure tensor component p_{xx}, (*b*) static pressure p; these two quantities have similar magnitudes and are shown with a shared colour scale. (*c*) Shear viscosity contribution μS_{xx}, (*d*) compressible contribution $\mu(\nabla \cdot \boldsymbol{u})$; these two quantities have similar magnitudes and are shown with a shared colour scale. The average absolute values over the entire domain are indicated in red text within each panel. Quantitatively, $\mu(\nabla \cdot \boldsymbol{u})$ is roughly four orders of magnitude smaller than the dominant stress terms, posing a significant challenge for data-driven discovery.

of conventional evolutionary approaches by enabling the identification of physically interpretable tensor equations with enhanced numerical efficiency.

To implement symbolic regression for tensors, SITE leverages a host–plasmid structure and incorporates a genetic information retention strategy to enhance evolutionary robustness. A DHC is enforced to ensure physical validity, while a seed injection strategy is adopted to overcome dimensional stagnation under strict constraints. In addition, a TLR technique replaces traditional random mutation–based coefficient optimization, enabling high efficiency and stability.

We validate SITE using two benchmark problems – including the Maxwell stress tensor and the Reynolds-stress transport equation – demonstrates its ability to recover ground truth equations from noisy data. Beyond synthetic tests, SITE is applied to DSMC-generated molecular data for lid-driven cavity flows, successfully identifying constitutive relations under both compressible and incompressible conditions. Notably, SITE also proves robustness in identifying small-magnitude terms, highlighting its sensitivity to meaningful physical features.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

Despite these encouraging results, SITE still has some limitations that warrant further investigation. First, SITE cannot automatically introduce unknown dimensional physical parameters if they are not provided in the candidate terminal set. In cases studied here, parameters such as permittivity or viscosity are explicitly included, allowing SITE to recover the correct expression structures. However, for problems where the relevant dimensional parameters are unknown *a priori*, the framework cannot independently infer or create such quantities, which may restrict its applicability. Second, the current SITE framework is restricted to discovering governing equations composed of algebraic and differential tensor functions that are explicitly included in the candidate library. It does not yet support non-local operations involving integration operators, convolution or historically correlated kernels, which are indispensable in certain materials and fluid systems. Finally, the generation of genes follows a random process without directional guidance, which can limit the efficiency of discovering meaningful structures, particularly as the candidate libraries become larger.

In summary, SITE provides a general, flexible and interpretable framework for tensor equation discovery in high-dimensional physical systems. While this work focuses on second-order tensors, future extensions may explore the incorporation of higher-order tensors and more generalized symbolic operators, such as the non-local operators mentioned above. Furthermore, reinforcement learning and large language models could be incorporated into SITE to provide valuable guidance in the generation and evolution of population. Such advancements could significantly broaden the framework's applicability both in science and engineering.

Acknowledgements. The authors gratefully acknowledge Professor R.D. Sandberg and his team at the University of Melbourne for their kind assistance with running M-GEP tests, which provided valuable reference for the development of this work. The authors thank P. Luan and Q. Ma for their help with data generation using OpenFOAM.

Funding. This work was supported by the National Natural Science Foundation of China (grant nos. 12272028 and 92371102).

Declaration of interests. The authors report no conflict of interest.

Author contributions. T.C., W.M. and J.Z. contributed to the ideation and design of the research; T.C. and H.Y. generated the datasets; T.C. performed the research; T.C., H.Y. and J.Z. wrote the paper.

Data availability statement. The datasets and source codes used in this work are available on GitHub at https://github.com/PistilReaper/SITE.

Appendix A. Genetic operators

Within the evolutionary process of SITE, both the host population and the plasmid population are subjected to genetic operator applications. More specifically, the genetic operations on the host and plasmid populations are decoupled. First, the host population undergoes modifications and crossovers. After completing these operations on the host population, the plasmid population then undergoes modifications. Modification (Algorithm 1) is divided into mutation and transposition, where transposition is further subdivided into gene fragment inversion, insertion sequence transposition, root insertion sequence transposition and whole gene transposition. Crossover (Algorithm 2) is divided into single-point crossover, double-point crossover and whole gene crossover. These genetic operators are applied to the offspring with certain probabilities.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

Algorithm 1 Modification

```
Input: population, operator, p_b
function APPLY_MODIFICATION (population, operator, p_b)
    N ← length(population)                          ▷ Get the population size
    for i ← 0 to N − 1 do
        if random() < p_b then
            population[i] ← operator(population[i])
            delete fitness of population[i]          ▷ Invalidate fitness
        end if
    end for
    return population
end function
```

Algorithm 2 Crossover

```
Input: population, operator, p_b
function APPLY_CROSSOVER (population, operator, p_b)
    N ← length(population)                          ▷ Get the population size
    for i ← 1 to N − 1 step 2 do                    ▷ Iterate in pairs
        if random() < p_b then
            (population[i], population[i + 1]) ←
    operator (population[i], population [i + 1])
            delete fitness of population [i]
            delete fitness of population [i + 1]
        end if
    end for
    return population
end function
```

In this work, the probabilities of applying each genetic operator refer to Ferreira (2001), listed in table 7. It is important to note that, unlike other operators, the mutation operator's probability listed in the table is not its probability of being applied to an entire individual. In GEP, the mutation operator is applied to every offspring (i.e. the per-individual application probability is 1). Then, the operator traverses all gene elements in a chromosome, and each gene element is changed with the probability specified in the table (effectively, this is the calling probability for each gene element). As a result, even after the mutation operator is applied, an individual may remain unchanged if none of its gene elements are mutated. Additionally, genetic operators that operate directly on genes only become effective when the number of genes is greater than one.

In practical tests, SITE is not sensitive to the application probability of the genetic operators. Therefore, when choosing these probabilities, you can adjust them upward or downward according to whether you want to enhance exploration or achieve faster convergence.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

Genetic operator	Object	Probability
Mutation	Host	0.2
	Plasmid	0.05
Inversion	Host	0.2
	Plasmid	0.1
IS transposition	Host	0.2
	Plasmid	0.1
RIS transposition	Host	0.2
	Plasmid	0.1
Gene transposition	Host	0.2
	Plasmid	0.1
Single-point crossover	Host	0.2
Double-point crossover	Host	0.2
Gene crossover	Host	0.2

Table 7. Probabilities for genetic operators being invoked.

Hyperparameter	Case 1	Case2 (clean / noisy)	Case 3
The maximum number of generations	500	1000 / 3000	500
Loss threshold	10^{-6}	10^{-6} / 0.2	10^{-6}
Number of genes in a host chromosome	4	6	4
Number of genes in a plasmid chromosome	1	1	1
Head length of the host gene	5	5	5
Head length of the plasmid gene	10	10	10
Number of individuals in the population	1600	1600	1600
The number of elites	1	1	1
Tournament size	200	200	200
The number of seed individuals	100	100	100

Table 8. Values of hyperparameters used in SITE. Case 1, Case 2 and Case 3 represent three different cases in this study (Case 1, Maxwell stress tensor; Case 2, Reynolds stress transport equation; Case 3, molecular simulation-based constitutive relation identification).

Appendix B. Hyperparameters

There are several key hyperparameters that govern SITE's evolutionary process. Table 8 summarizes the values used in this study, which were selected based on prior benchmarking.

The maximum number of generations together with the loss threshold determines when the evolutionary process terminates. These two parameters provide complementary stopping criteria: one sets a hard upper bound on search effort, while the other defines a tolerance for convergence.

The number of host genes specifies the maximum number of independent tensor terms that can appear in the final governing equation, while the number of plasmid genes controls the complexity of embedded scalar expressions. The head lengths of host and plasmid genes regulate the richness of symbolic structures that can be generated, with larger head lengths allowing for more diverse combinations of candidate elements.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

The remaining parameters are at the population evolution level. These settings generally do not have a significant impact on SITE's optimization process. Compared with classical GEP settings, we have moderately increased the tournament size, which can enhance the convergence efficiency.

Among all hyperparameters, the first three parameters are case-dependent. For the benchmark problems (Case 1 and Case 2), these values are adjusted according to the complexity of the task, with more generations and host genes allocated to the Reynolds-stress transport equation than to the Maxwell stress tensor. For the cavity-flow case based on molecular simulation (Case 3), a relatively small number of host genes and generations are sufficient because the flow conditions are not highly complex. However, since this case involves statistical noise and does not have a known ground truth, we avoided imposing a loose loss threshold. Instead, we chose a very small threshold (10^{-6}) and allowed the algorithm to evolve up to the maximum generation limit, in order to objectively observe the best equation that SITE can discover.

Appendix C. Construction of the terminal library for constitutive relation identification

In constructing the candidate terminal library for the compressible cavity flow case, our objective was to avoid imposing any strong prior modelling assumptions regarding the form of the constitutive relation. Instead, we aimed to provide a physically comprehensive yet minimally biased set of inputs that could enable the discovery of the constitutive relation purely from data.

We initially included both the symmetric strain-rate tensor S_{ij} and the antisymmetric rotation-rate tensor Ω_{ij} in the terminal library for tensors. However, preliminary experiments showed that Ω_{ij} was consistently absent in the expressions discovered by SITE. This result aligns with physical analysis: Ω_{ij} represents local rigid-body rotation, which does not contribute to momentum transport in isotropic fluids. As a result, the final terminal library for tensor is set to $\{S_{ij}, \delta_{ij}\}$.

For the scalar terminal library, we selected $\{D_{ii}, p, \mu\}$. Here, $D_{ii} = \nabla \cdot u$ is the divergence of the velocity field and serves as a fundamental scalar invariant of the velocity gradient tensor D_{ij}. The static pressure p and dynamic viscosity μ are intrinsic thermodynamic and transport properties of the flow, respectively. These scalar quantities enable SITE to capture both shear and compressible effects without presuming linear or Newtonian behaviour.

Appendix D. Candidate equations for constitutive relation identification and construction of the reported form

In § 4, the true governing constitutive equation is unknown. Therefore, the evaluation of symbolic regression results cannot rely on an exact comparison with a known target equation. To address this, we conducted five independent discovery runs under each of the two flow conditions (lid velocities of 50 and 337 m s^{-1}). In each run, 200 randomly selected data points (see figure 6) were used as training data, and all 57 600 points in the same region were employed as the test data. This allows for an objective error-based evaluation, even in the absence of ground truth equations. The test loss is calculated by (2.6).

Table 9 and table 10 present the explicit candidate equations obtained from the five independent runs under the 50 and 337 m s^{-1} conditions, respectively, along with their predictive test errors. Although the identified equations are not exactly the same, clear

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

	Equation form	Test error
Eq. 1	$p_{ij} = -1.908\mu S_{ij} + 1.000 p\delta_{ij} + (1.060 \times 10^{-6}) \dfrac{\mu}{D_{mm}} S_{ik}S_{kj}$	4.094 %
Eq. 2	$p_{ij} = -1.899\mu S_{ij} + 1.000 p\delta_{ij}$	4.099 %
Eq. 3	$p_{ij} = -1.933\mu S_{ij} + 1.000 p\delta_{ij} - 3.865\mu D_{kk}\delta_{ij}$	4.155 %
Eq. 4	$p_{ij} = -1.895\mu S_{ij} + 1.000 p\delta_{ij} - (2.600 \times 10^{-7}) \dfrac{\mu}{D_{mm}} S_{ik}S_{kj} + 2.000\mu D_{kk}\delta_{ij}$	4.103 %
Eq. 5	$p_{ij} = -1.908\mu S_{ij} + 1.000 p\delta_{ij} + (1.370 \times 10^{-6}) \dfrac{\mu}{D_{mm}} S_{ik}S_{kj} - 1.000\mu D_{kk}\delta_{ij}$	4.094 %

Table 9. Candidate equations obtained from five independent runs under the 50 m s^{-1} cavity flow condition, together with their predictive errors on the test dataset. The indices i, j, k and m denote tensor indices.

	Equation form	Test error
Eq. 1	$p_{ij} = -1.959\mu S_{ij} + 0.676\mu D_{kk}\delta_{ij} + 1.000 p\delta_{ij}$	1.593 %
Eq. 2	$p_{ij} = -1.953\mu S_{ij} + 0.664\mu D_{kk}\delta_{ij} + 1.000 p\delta_{ij}$	1.476 %
Eq. 3	$p_{ij} = -1.963\mu S_{ij} + 0.675\mu D_{kk}\delta_{ij} + 1.000 p\delta_{ij}$	1.674 %
Eq. 4	$p_{ij} = -1.961\mu S_{ij} + 0.681\mu D_{kk}\delta_{ij} + 1.000 p\delta_{ij}$	1.633 %
Eq. 5	$p_{ij} = -1.946\mu S_{ij} + 0.721\mu D_{kk}\delta_{ij} + 1.000 p\delta_{ij} - (5.375 \times 10^{-7}) \dfrac{\mu}{D_{mm}} S_{ik}S_{kj}$	1.351 %

Table 10. Candidate equations obtained from five independent runs under the 337 m s^{-1} cavity flow condition, together with their predictive errors on the test dataset. The indices i, j, k and m denote tensor indices.

structural similarities are observed. Several terms appear consistently across different runs, indicating that they are statistically robust features of the underlying relationship. By contrast, other terms appear sporadically and contribute marginally to accuracy.

Specifically, for the nearly incompressible flow condition, all candidate equations consistently contain the shear stress term and the static pressure term. However, the inclusion of the compressible term is unstable, and its coefficients fluctuate significantly across different runs. This behaviour is expected since the velocity divergence is close to, but not exactly, zero in this regime. The appearance of the compressible term is generally attributable to its coupling with other dominant terms within the same symbolic structure, rather than representing a truly independent physical contribution. For the compressible flow condition, all candidate equations robustly include the shear stress term, the compressible term and the static pressure term.

In addition, certain candidate equations contain higher-order terms. These higher-order contributions typically have small coefficients and reduce the test error only marginally. From a theoretical perspective, such terms can be interpreted as artefacts of local non-equilibrium effects. In particular, for high-speed compressible flows, the inclusion of higher-order corrections may capture weak deviations from continuum-level constitutive assumptions, thereby slightly improving predictive accuracy. Nonetheless, their limited impact on the loss function suggests that these effects remain secondary compared with the dominant physical terms.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

References

Bakarji, J., Callaham, J., Brunton, S.L. & Kutz, J.N. 2022 Dimensionally consistent learning with Buckingham Pi. *Nat. Comput. Sci.* **2** (12), 834–844.

Bird, G.A. 1994 Molecular Gas Dynamics and the Direct Simulation of Gas Flows. Oxford University Press.

Bongard, J. & Lipson, H. 2007 Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **104** (24), 9943–9948.

Brunton, S.L., Noack, B.R. & Koumoutsakos, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52** (2020), 477–508.

Brunton, S.L., Proctor, J.L. & Kutz, J.N. 2016 Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl Acad. Sci. USA* **113** (15), 3932–3937.

Byrd, R.H., Lu, P., Nocedal, J. & Zhu, C. 1995 A limited memory algorithm for bound constrained optimization. *Siam J. Sci. Comput.* **16** (5), 1190–1208.

Chen, Y., Luo, Y., Liu, Q., Xu, H. & Zhang, D. 2022 Symbolic genetic algorithm for discovering open-form partial differential equations (SGA-PDE). *Phys. Rev. Res.* **4** (2), 023174.

Chen, Z., Liu, Y. & Sun, H. 2021 Physics-informed learning of governing equations from scarce data. *Nat. Commun.* **12** (1), 6136.

Cohen, B.G., Beykal, B. & Bollas, G.M. 2024 Physics-informed genetic programming for discovery of partial differential equations from scarce and noisy data. *J. Comput. Phys.* **514**, 113261.

Cranmer, M. 2023 Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl. arXiv:2305.01582.

Du, M., Chen, Y., Nie, L., Lou, S. & Zhang, D. 2024 Physics-constrained robust learning of open-form partial differential equations from limited and noisy data. *Phys. Fluids* **36** (5), 057123.

Ferreira, C. 2001 Gene expression programming: a new adaptive algorithm for solving problems. *Complex Syst.* **13** (2), 87–129.

Fukami, K., Goto, S. & Taira, K. 2024 Data-driven nonlinear turbulent flow scaling with Buckingham Pi variables. *J. Fluid Mech.* **984**, R4.

Goldberg, D.E. & Richardson, J. 1987 Genetic algorithms with sharing for multimodal function optimization. In *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, vol. 4149, pp. 414–425. Lawrence Erlbaum. Lawrence Erlbaum.

Golden, M., Grigoriev, R.O., Nambisan, J. & Fernandez-Nieves, A. 2023 Physically informed data-driven modeling of active nematics. *Sci. Adv.* **9** (27), eabq6120.

Gurevich, D.R., Golden, M.R., Reinbold, P.A.K. & Grigoriev, R.O. 2024 Learning fluid physics from highly turbulent data using sparse physics-informed discovery of empirical relations (SPIDER). *J. Fluid Mech.* **996**, A25.

Hadjiconstantinou, N.G., Garcia, A.L., Bazant, M.Z. & He, G. 2003 Statistical error in particle simulations of hydrodynamic phenomena. *J. Comput. Phys.* **187** (1), 274–297.

Jasak, H., Jemcov, A. & Tukovic, Z. 2007 Openfoam: A C++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, vol. 1000, pp. 1–20, Dubrovnik, Croatia.

Keijzer, M. 2003 Improving symbolic regression with interval arithmetic and linear scaling. In *European Conference on Genetic Programming*, pp. 70–82. Springer.

Kingma, D.P. & Ba, J. 2017, Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Koza, J.R. 1994 Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **4**, 87–112.

Launder, B.E., Reece, G., Jr & Rodi, W. 1975 Progress in the development of a Reynolds-stress turbulence closure. *J. Fluid Mech.* **68** (3), 537–566.

Lennon, K.R., McKinley, G.H. & Swan, J.W. 2023 Scientific machine learning for modeling and simulating complex fluids. *Proc. Natl Acad. Sci. USA* **120** (27), e2304669120.

Li, H., Zhao, Y., Wang, J. & Sandberg, R.D. 2021 Data-driven model development for large-eddy simulation of turbulence using gene-expression programing. *Phys. Fluids* **33** (12), 125127.

Loiseau, J.-C. & Brunton, S.L. 2018 Constrained sparse Galerkin regression. *J. Fluid Mech.* **838**, 42–67.

Lu, L., Zeng, Z., Jiang, Y., Zhu, Y. & Hu, P. 2024 Weak collocation regression method: fast reveal hidden stochastic dynamics from high-dimensional aggregate data. *J. Comput. Phys.* **502**, 112799.

Ma, Q., Yang, C., Chen, S., Feng, K., Cui, Z. & Zhang, J. 2024*a* Effect of thermal fluctuations on spectra and predictability in compressible decaying isotropic turbulence. *J. Fluid Mech.* **987**, A29.

Ma, W., Zhang, J., Feng, K., Xing, H. & Wen, D. 2024*b* Dimensional homogeneity constrained gene expression programming for discovering governing equations. *J. Fluid Mech.* **985**, A12.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press

MASLYAEV, M., HVATOV, A. & KALYUZHNAYA, A. 2019 Data-driven partial derivative equations discovery with evolutionary approach, proceedings. In *Computational Science–ICCS 2019: 19th International Conference, Faro, Portugal, June 12–14, 2019, Proceedings, Part V 19*, pp. 635–641. Springer.

REESE, J.M., GALLIS, M.A. & LOCKERBY, D.A. 2003 New directions in fluid dynamics: non-equilibrium aerodynamic and microsystem flows. *Phil. Trans. R. Soc. Lond. A* **361** (1813), 2967–2988.

REINBOLD, P.A.K., KAGEORGE, L.M., SCHATZ, M.F. & GRIGORIEV, R.O. 2021 Robust learning from noisy, incomplete, high-dimensional experimental data via physically constrained symbolic regression. *Nat. Commun.* **12** (1), 3219.

REISSMANN, M., FANG, Y., OOI, A.S.H. & SANDBERG, R.D. 2025 Constraining genetic symbolic regression via semantic backpropagation. *Genet. Program. Evol. Mach.* **26** (1), 12.

SCHMIDT, M. & LIPSON, H. 2009 Distilling free-form natural laws from experimental data. *Science* **324** (5923), 81–85.

UDRESCU, S.-M. & TEGMARK, M. 2020 Ai Feynman: a physics-inspired method for symbolic regression. *Sci. Adv.* **6** (16), eaay2631.

WANG, H. *et al.* 2023 *Scientific discovery in the age of artificial intelligence*. *Nature* **620** (7972), 47–60.

WANG, L., JI, T., HUANG, Y., ZHOU, H. & XIE, F. 2025 Physics-based discovery of governing equations from scarce and noisy data. *Phys. Rev. E* **111** (6), 065314.

WEATHERITT, J. & SANDBERG, R. 2016 A novel evolutionary algorithm applied to algebraic modifications of the RANS stress–strain relationship. *J. Comput. Phys.* **325**, 22–37.

ZHANG, B., LEI, J., SUN, G., DING, S. & GUO, J. 2025 Cartesian tensor-based sparse regression for data-driven discovery of high-dimensional invariant governing equations. *Phys. Fluids* **37** (7), 077191.

ZHANG, J. & MA, W. 2020 Data-driven discovery of governing equations for fluid dynamics based on molecular simulation. *J. Fluid Mech.* **892**, A5.

ZHAO, Y., AKOLEKAR, H.D., WEATHERITT, J., MICHELASSI, V. & SANDBERG, R.D. 2020 RANS turbulence model development using CFD-driven machine learning. *J. Comput. Phys.* **411**, 109413.

https://doi.org/10.1017/jfm.2025.10710 Published online by Cambridge University Press