

# Generative discovery of partial differential equations by learning from math handbooks

Received: 8 July 2025

Accepted: 8 October 2025

Published online: 21 November 2025

 Check for updates

Hao Xu<sup>1,2</sup>, Yuntian Chen<sup>1,3</sup> , Rui Cao<sup>4,5</sup>, Tianning Tang<sup>6,7</sup>, Mengge Du<sup>8</sup>, Jian Li<sup>3</sup>, Adrian H. Callaghan<sup>5</sup> & Dongxiao Zhang<sup>1,9</sup> 

Data-driven discovery of partial differential equations (PDEs) is a promising approach for uncovering the underlying laws governing complex systems. However, purely data-driven techniques face the dilemma of balancing search space with optimization efficiency. This study introduces a knowledge-guided approach that incorporates existing PDEs documented in a mathematical handbook to facilitate the discovery process. These PDEs are encoded as sentence-like structures composed of operators and basic terms, and used to train a generative model, called EqGPT, which enables the generation of free-form PDEs. A loop of “generation–evaluation–optimization” is constructed to autonomously identify the most suitable PDE. Experimental results demonstrate that this framework can recover a variety of PDE forms with high accuracy and computational efficiency, particularly in cases involving complex temporal derivatives or intricate spatial terms, which are often beyond the reach of conventional methods. The approach also exhibits generalizability to irregular spatial domains and higher dimensional settings. Notably, it succeeds in discovering a previously unreported PDE governing strongly nonlinear surface gravity waves propagating toward breaking, based on real-world experimental data, highlighting its applicability to practical scenarios and its potential to support scientific discovery.

Partial differential equations (PDEs) are important tools for describing complex dynamic processes in nature and constitute one of the cornerstones of scientific research. For a long time, PDEs have been developed on the basis of first-principle derivation, which typically relies on manual effort. In recent years, with improvements in data accessibility and computational science, discovering PDEs directly from high-fidelity observations has become a promising way to identify potential governing laws in complex systems of diverse fields<sup>1,2</sup>. The discovered PDEs possess better interpretability and generalizability than black-box models, and are able to reveal

undisclosed mechanisms and insights, such as conservation laws and symmetries<sup>3</sup>.

The identification of underlying PDEs from data typically involves two key aspects: determining the relevant terms; and capturing the relationships among them. In the literature, sparse regression techniques are introduced to address this by constructing a candidate library based on a priori selection of potential terms, and subsequently identifying a parsimonious structure through  $L_0$  or  $L_1$  normalization<sup>4,5</sup>. Despite its computational efficiency and straightforward implementation, sparse regression faces limitations in practice since it is

<sup>1</sup>Zhejiang Key Laboratory of Industrial Intelligence and Digital Twin, Eastern Institute of Technology, Ningbo, Zhejiang, PR China. <sup>2</sup>Department of Electrical Engineering, Tsinghua University, Beijing, PR China. <sup>3</sup>Ningbo Institute of Digital Twin, Eastern Institute of Technology, Ningbo, Zhejiang, PR China. <sup>4</sup>College of Oceanic and Atmospheric Sciences, Ocean University of China, Qingdao, PR China. <sup>5</sup>Department of Civil and Environmental Engineering, Imperial College London, London, UK. <sup>6</sup>Department of Engineering Science, University of Oxford, Oxford, UK. <sup>7</sup>Department of Mechanical and Aerospace Engineering, University of Manchester, Manchester, UK. <sup>8</sup>College of Engineering, Peking University, Beijing, PR China. <sup>9</sup>Institute for Advanced Study, Lingnan University, Tuen Mun, Hong Kong. ✉e-mail: [ychen@eitech.edu.cn](mailto:ychen@eitech.edu.cn); [dzhang@eitech.edu.cn](mailto:dzhang@eitech.edu.cn)

often infeasible to enumerate all possible candidate terms, given the diversity of nonlinear interactions and composite expressions in derivative terms. When more candidate terms are included, the iterative elimination of terms with small coefficients may fail to converge to a correct equation. While weak-form variants have been proposed to improve stability<sup>6–8</sup>, these are often constrained to polynomial or high-order derivatives, and struggle to handle more intricate terms, such as  $u_{xxt}$  or  $xu_t$ . In essence, the effectiveness of sparse regression is closely tied to the completeness and design of the candidate library—yet constructing such a library remains a challenge, particularly for systems with complex dynamics. In response to these limitations, some symbolic regression-based approaches, including genetic algorithms<sup>9,10</sup> and tree-based search strategies<sup>11–13</sup>, are proposed to discover PDEs in a free-form manner. While this alleviates the dependence on predefined libraries and substantially expands the search space, it also incurs optimization challenges, particularly under high-noise conditions or in multivariable systems.

Recent advances in generative models have introduced new opportunities for data-driven equation discovery<sup>14,15</sup>. Current efforts can be broadly categorized into three approaches. The first is “prompt-based discovery,” which leverages general-purpose large language models (LLMs), such as GPT-4 or Gemini<sup>16–18</sup>. They can infer potential candidates based on prompt templates that encode problem context and inquires. However, such approaches rely heavily on human involvement for prompt design and output evaluation, and they typically lack the ability to directly process observational data, resulting in high cost and limited efficiency. The second is “data-to-sequence discovery,” in which generative models are trained to directly map from data to equation structures<sup>19,20</sup>. While inference is fast, this requires large-scale paired datasets of PDEs and their solutions for training. Given the high computational cost of PDE simulation, such models are currently limited in scope and accuracy, and often restricted to a narrow range of canonical equations. The third is “sequence optimization discovery,” in which generative models produce candidate equations that are subsequently optimized using reinforcement learning, such as risk-seeking policy gradient<sup>21–24</sup>. However, this approach inherits the difficulties of symbolic regression. In PDE discovery, the primary source of complexity arises not from mathematical nesting, but from intricate combinations of differential operators. Prefix tree-based optimization strategies often introduce excessive search complexity and spend computational resources on physically implausible terms, which results in poor robustness and efficiency, particularly in noisy, high-dimensional scenarios.

Taken together, these challenges reveal a common limitation across both traditional sparse regression and emerging LLM-based approaches: a purely data-driven search process often faces a trade-off between search space size and optimization efficiency. While recent methods have improved optimization techniques, they still operate largely independently of domain-specific prior knowledge. In this work, we propose a different strategy. Rather than focusing solely on optimization, we explore how existing mathematical and scientific knowledge can be embedded into the equation discovery process. Our central idea is inspired by the notion that “stones from other hills may polish one’s own jade”—namely, that general knowledge of PDE structures, summarized from established scientific literature, can guide the discovery of new equations in specific contexts.

Specifically, we present a framework that integrates knowledge-guided and data-driven strategies for PDE discovery (Fig. 1). Two techniques are introduced, including generative representation of equation (GRE) and scientifically augmented training (SAT). In the GRE, a structured equation encoding scheme is proposed, wherein equations are parsed into vocabularies composed of operators (e.g., arithmetic symbols) and fundamental physical terms. These units are combined to form sequence representations of free-form equations.

In the SAT, a generative model, called EqGPT, is trained to learn co-occurrence patterns among PDE terms from a dataset of PDEs collected from a mathematical handbook<sup>25</sup>. This enables the autonomous generation of free-form candidate equations directly from physically meaningful tokens, while implicitly filtering out mathematically implausible expressions, thereby improving both the efficiency and the relevance of the search process without relying on brute-force enumeration. Through a knowledge-guided loop of “generation–evaluation–optimization”, PDEs that are both consistent with observed data and aligned with domain knowledge can be identified autonomously.

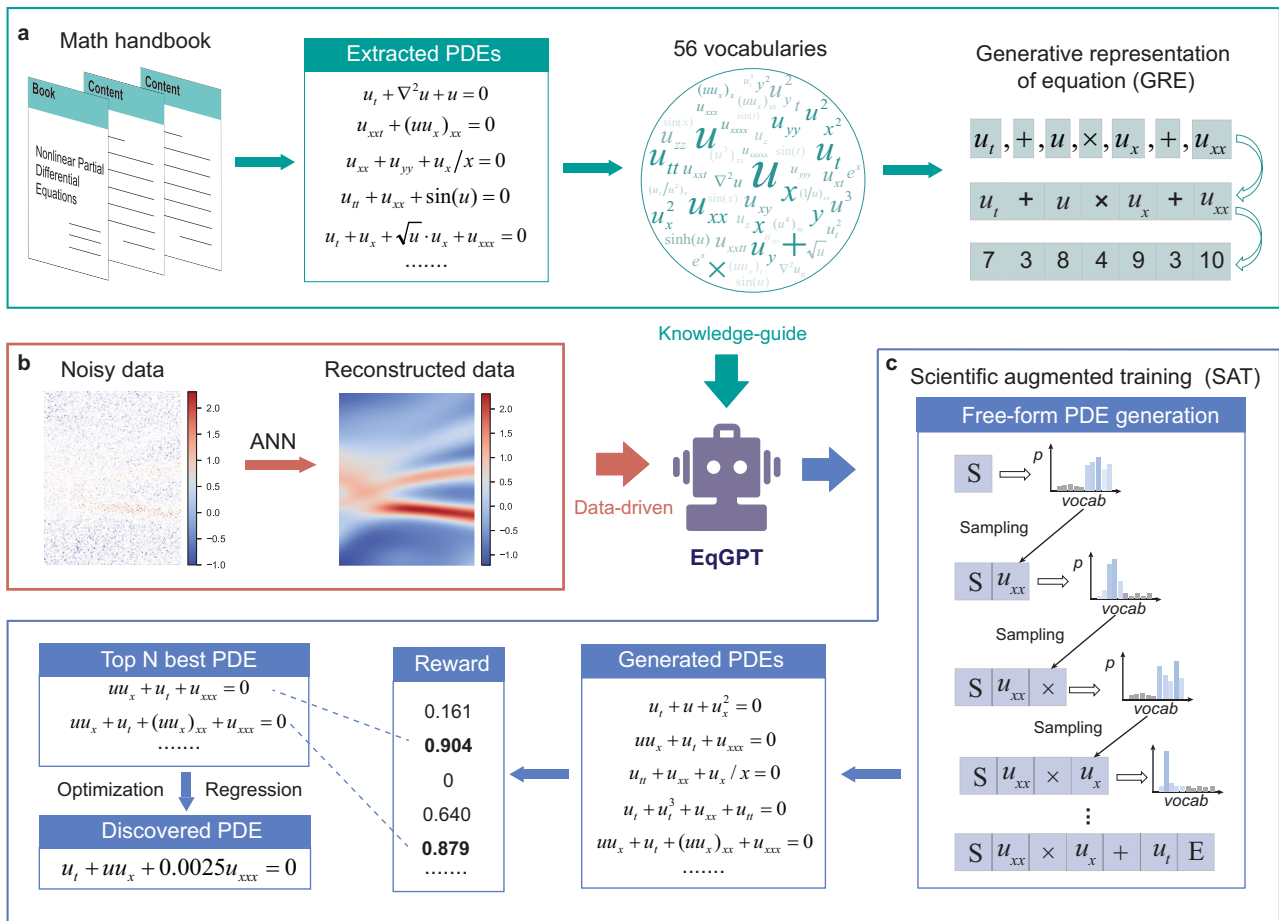
Proof-of-concept experiments validate the framework’s ability to rediscover canonical PDEs in irregular computational domains and to recover multi-variable spatiotemporal equations in high-dimensional settings. Comparative analysis with existing baseline approaches demonstrates its improved accuracy and robustness. Furthermore, we present a practical case study, in which the proposed approach successfully uncovers a previously unreported governing equation for highly nonlinear surface gravity waves propagating toward breaking from real-world experimental data. This example illustrates the proposed framework’s practical utility and provides concrete evidence that it can contribute to scientific understanding by revealing new relationships.

## Results

### The proposed knowledge-guided generative framework for PDE discovery

The research framework, as depicted in Fig. 1, consists of three key workflows, including knowledge-guided pathway, data-driven pathway, and combined optimization approach. In the knowledge-guided pathway, the scientific knowledge of existing PDEs is incorporated by generative representation of equations (GRE). Therefore, a dataset of 221 PDE structures is extracted from the scientific handbook *Nonlinear Partial Differential Equations*<sup>25</sup>, which contains canonical PDEs in science and engineering. Through 56 vocabularies obtained by tokenizing PDEs into operators and basic terms (listed in Table S1), PDEs can be converted into sentences (Fig. 1a). Considering that the position of the terms in the equation does not affect the structure or meaning of the equation itself, we performed data augmentation by swapping the positions of the terms in the equations, ultimately generating 7072 sentences. The encoded PDE structures are utilized to train the generative model, called EqGPT. Here, the architecture of the GPT-2 model is adopted<sup>26</sup>. Notably, EqGPT functions as both a knowledge learner and an equation generator, with its architecture extending beyond the constraints of GPT-2. The focus of this work is not on introducing a new neural architecture, but rather on illustrating how domain-specific knowledge, especially PDE structures, can be effectively embedded through pretraining to facilitate the equation discovery process. In the training process, the feature distribution of the PDE dataset and the co-occurrence probability of vocabularies are learned by predicting each token from the preceding token(s). The generative model is trained for 100 epochs in this work. The model can be subsequently utilized to generate new PDE structures. In the data-driven pathway, considering that observation data may be sparse and noisy, a fully connected artificial neural network (ANN) is trained with the observation data to function as a surrogate model, which is essentially a data generator that can smooth the data through interpolation. The surrogate model can not only generate abundant predictions on grids (meta-data), but also calculate derivatives through automatic differentiation in back propagation<sup>27</sup>. This facilitates the evaluation of discovered PDEs and provides robustness to high levels of noise.

For the optimization of the most suitable PDEs, the knowledge-guided and data-driven pathways are combined to form a closed loop of “generation–evaluation–optimization”. In the generation process, the start token (S) is first input into the model, and the next token is



**Fig. 1 | Overview of the proposed knowledge-guided generative framework for PDE discovery.** **a** The construction of knowledge guidance, including extracting existing PDEs from a math handbook, which is tokenized into 56 vocabularies (basic terms) and converted to sentences through generative representation of

equations (GRE). **b** The data-driven workflow, in which sparse and noisy data are reconstructed by an artificial neural network (ANN). **c** The loop of “generation–evaluation–optimization”, including knowledge-guided generation of free-form PDEs, data-driven evaluation, and model fine-tuning.

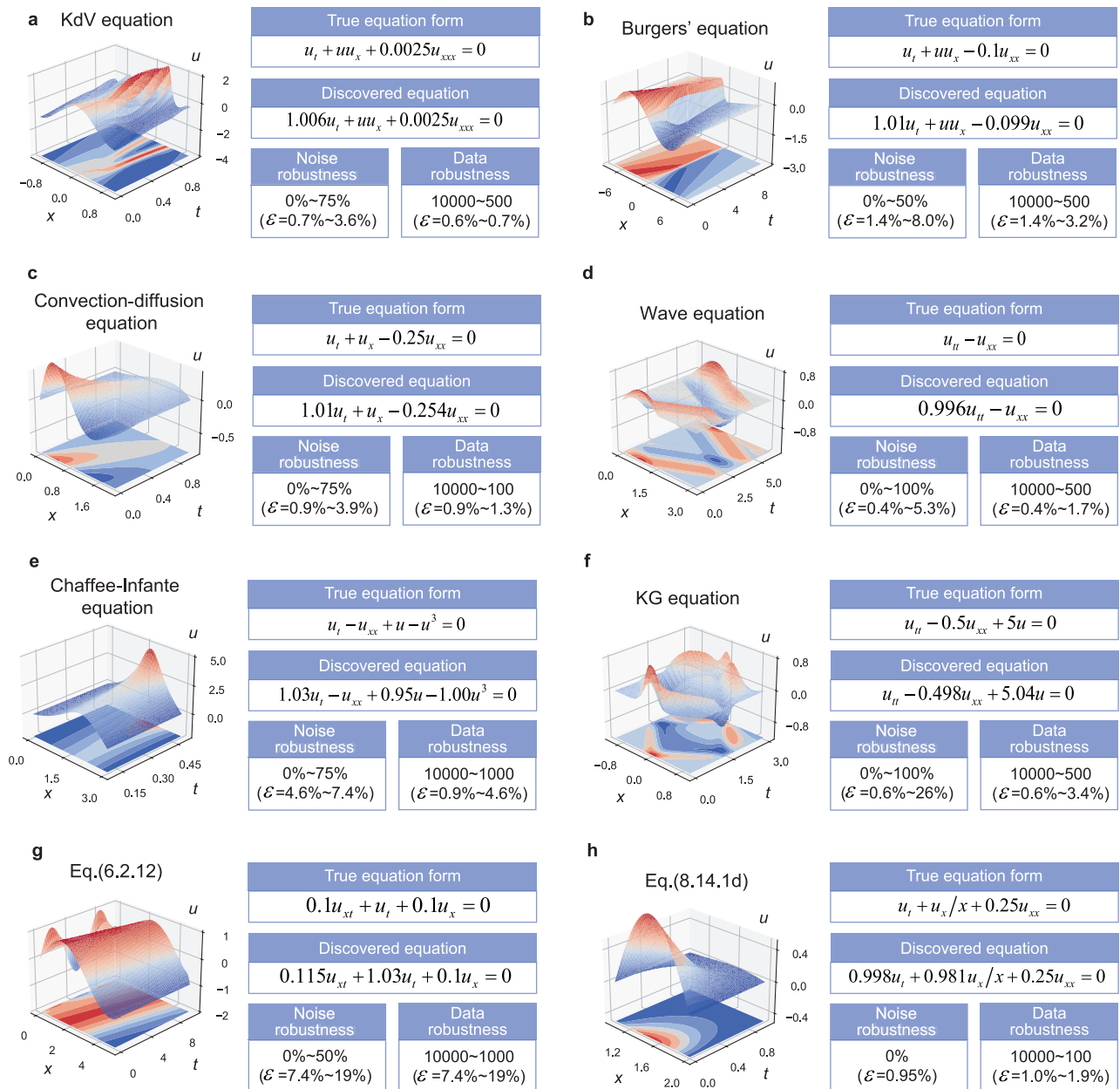
sampled from the predicted probability distribution. The model repeatedly samples tokens from the learned distribution until the end token is sampled, indicating the completion of a new PDE structure (Fig. 1c). Here, the probabilistic sampling and random sampling are combined to guarantee the variety of generative PDEs. Therefore, EqGPT can generate infinitely many equation structures by combining operators with physically meaningful terms from its token vocabulary. Benefitted from the proposed GRE technique for PDEs, EqGPT can generate diversified free-form PDEs. A reward function is defined to evaluate the generated PDEs:

$$\text{Reward} = \alpha \cdot R^2 = (1 - \alpha_0 \log_{10} N_{\text{term}}) \cdot \left( 1 - \frac{\sum (\theta_{LHS} - \theta_{RHS} \cdot \vec{\xi})^2}{\sum (\theta_{LHS} - \bar{\theta}_{LHS})^2} \right) \tag{1}$$

where the reward is composed of two components: the penalty term  $\alpha$  and the determination coefficient  $R^2$ . The former is defined to measure the sparsity, which penalizes the number of terms in the structure to obtain a parsimonious PDE, and the latter is employed to measure the consistency with observation data through the  $R^2$  of the regression. Notably, EqGPT generates the structural form of an equation (e.g.,  $u_x + u_t + u \times u_x = 0$ ), while the coefficients of terms are determined through regression. To perform least-squares regression, we move the first term to the left-hand side (LHS) and treat the remaining terms as the right-hand side (RHS) terms (e.g.,  $-u_x = u_t + u \times u_x$ ).

Therefore, the LHS term is designated as the negative of the first term. Using the surrogate model trained on observation data, automatic differentiation is employed to compute the numerical values of each PDE term at the  $N_{\text{meta}}$  spatiotemporal meta-data points. Thus, the value of LHS term is represented as a vector  $\theta_{LHS}$  with the size of  $N_{\text{meta}} \times 1$ , while the values of RHS term is represented as a matrix  $\theta_{RHS}$  of size  $N_{\text{meta}} \times (N_{\text{term}} - 1)$ .  $N_{\text{term}}$  is the total number of terms in the structure. We then solve a least-squares regression problem,  $\theta_{LHS} \approx \theta_{RHS} \vec{\xi}$ , where  $\vec{\xi}$  is the coefficient vector obtained from regression with the size of  $(N_{\text{term}} - 1) \times 1$ . Through the observed and fitted LHS term, the  $R^2$  can be calculated to represent their agreement. In the equation,  $\bar{\theta}_{LHS}$  is the mean value of  $\theta_{LHS}$ . Overall, the higher is the reward, the better is the generated PDE.

After the reward is calculated, the top 10 equations with the highest rewards are selected to fine-tune the EqGPT model (Fig. 1c). In other words, they are utilized as the training data to further adjust the model parameters with a smaller learning rate,  $10^{-5}$ . The fine-tuning epoch is 5. The fine-tuned model then generates new PDE structures and updates the records of the top 10 structures, which constitute an optimization cycle. Initially, EqGPT is only trained on mathematical handbooks, which ensures that its generations are mathematically valid but not necessarily consistent with the observed data. The fine-tuning strategy serves to adapt the generative model by learning from the top-ranked equations that best fit the data. In this way, EqGPT increases the likelihood of sampling terms that are more consistent with the underlying dynamics, since equations with higher rewards are more likely to contain the correct components. This mechanism is



**Fig. 2 | The performance of the proposed generative framework in discovering canonical PDEs with sparse and noisy data. a** Korteweg-De Vries (KdV) equation. **b** Burgers' equation. **c** Convection-diffusion equation. **d** Wave equation.

**e** Chaffee-infante equation. **f** Klein-Gordon (KG) equation. **g** Eq. (6.2.12) in the math handbook<sup>25</sup>. **h** Eq. (8.14.d) in the math handbook.  $\mathcal{E}$  is the coefficient error.

demonstrated in Table S2 by an example for better understanding. In the first epoch, the top-ranked equations may already include some correct terms but still deviate from the true structure, whereas after fine-tuning, EqGPT increasingly favors higher-reward terms and successfully recovers the correct PDE by the third epoch. Since the number of training data for fine-tuning is tiny (only 10), the fine-tuning process is fast (only several seconds). When the optimization cycle reaches the maximum number of epochs (5 in this work), the structure with the highest reward is the optimal PDE. On this basis, the coefficients of the best PDE structure are regressed and optimized to obtain the final PDE form. Additional details about the framework can be found in the Methods.

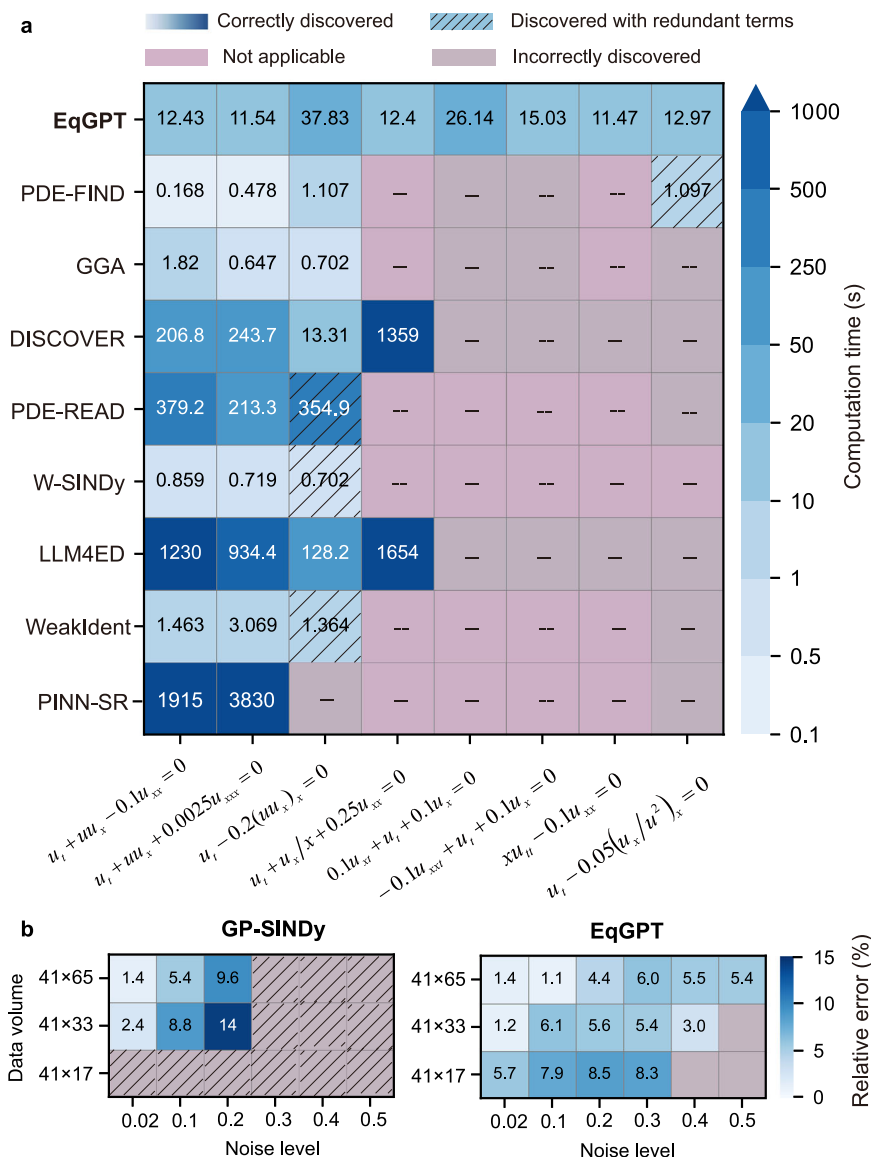
#### Discovery of canonical PDEs with sparse and noisy data

For proof-of-concept, 8 canonical PDEs from different scientific fields are employed to examine the performance of the proposed framework

with sparse and noisy data. The true equation form, discovered equation form, noise robustness, and data robustness for discovering each PDE are illustrated in Fig. 2. Here, the discovered equation represents the outcome of the framework applied to 10,000 data points. The noise is Gaussian noise, which is defined as:

$$\tilde{u} = \gamma \cdot \text{std}(u) \cdot N(0, 1) + u \quad (2)$$

where  $u$  represents the clean data;  $\tilde{u}$  represents the noisy data;  $N(0,1)$  represents the standard normal distribution; and  $\gamma$  represents the noise level. To assess the noise robustness, the noise level is systematically elevated by 25% increments until the identification of the correct PDE form becomes unattainable. To examine the data robustness, the number of training data is decreased progressively as 10000, 5000, 2500, 1000, 500, and 100 until the correct PDE form fails to be discovered. If the equation form is correctly identified, the coefficient



**Fig. 3 | The comparison between the proposed method and existing methods on the discovery of canonical PDEs. a** Performance of different methods on discovering eight canonical PDEs from clean data. The color scale indicates the

computational cost. **b** Performance of EqGPT and GP-SINDy method on discovering Burgers’ equation under sparse and noisy data. The heatmaps report the relative error of discovered coefficients under different data volumes and noise levels.

error is employed to measure the accuracy of the discovered PDE, which is defined as:

$$\varepsilon = \frac{1}{N_{term}} \sum_{i=1}^{N_{term}} \left| \frac{\xi_i - \xi_i^{true}}{\xi_i^{true}} \right|, \quad (3)$$

where  $N_{term}$  is the number of PDE terms;  $\xi_i$  is the coefficient of the discovered PDE terms; and  $\xi_i^{true}$  is the coefficient of true PDE terms. Notably, in the experiments, the target PDE is deleted from the dataset to ensure that the EqGPT model has never seen the target PDE, which guarantees the validity of the experiments. However, experiments have shown that the discovery of PDEs with and without target PDEs in the PDE dataset does not significantly influence the ultimate results (Supplementary Information S1. 2).

Upon analysis of the results, we can derive several key insights. First, the proposed framework can discover correct PDEs accurately with more than 50% noise and fewer than 1000 datapoints in most cases (Fig. 2a–f). This robustness is greater than that of existing mainstream PDE discovery algorithms<sup>28–30</sup>, which confirms the

general ability of our proposed framework to discover PDEs under sparse and noisy data; this is attributed to the surrogate model utilized in the proposed framework, since it can reconstruct the underlying process from sparse and noisy observations, which smooths the noise and generates sufficient meta-data on grids for PDE discovery. Additional experiments confirm the robustness of the proposed method under conditions of both data sparsity and high noise levels, as detailed in Supplementary Information S1. 5. Moreover, the EqGPT model can generate proper structures, which simplifies the optimization process and facilitates the identification of the correct structures.

Notably, the proposed framework can discover PDEs with broader forms, such as uncommon derivatives and terms with fractions, which were previously difficult to discover. For example, the PDE in Fig. 2g contains a rarely seen derivative term  $u_{xt}$ , which was not discovered in preceding works. Without any prior knowledge, existing knowledge discovery algorithms have difficulty in handling such situations since they require predetermined terms for regression and usually assume that the left-hand side term is  $u_t$  or  $u_{tt}$ . In contrast, the current

framework successfully identifies this PDE, as it learns the term  $u_{xt}$  when training the generative model, although its origin could be traced to other equations, as well.

Figure 2h provides another PDE that involves fraction terms,  $u_x/x$ . It is also difficult to discover by sparse regression or genetic evolution algorithms since they cannot generate free-form fraction terms. The proposed framework is able to handle this problem in a straightforward manner because of the operator “/” defined in the vocabularies. Furthermore, the proposed framework remains stable and accurate even with 100 data points for this complex situation, which confirms its ability to handle broader forms of PDE. Moreover, the framework can also recognize other mathematically equivalent forms beyond simple permutations, as demonstrated in Supplementary Information S1. 4.

### Comparison against existing PDE discovery algorithms

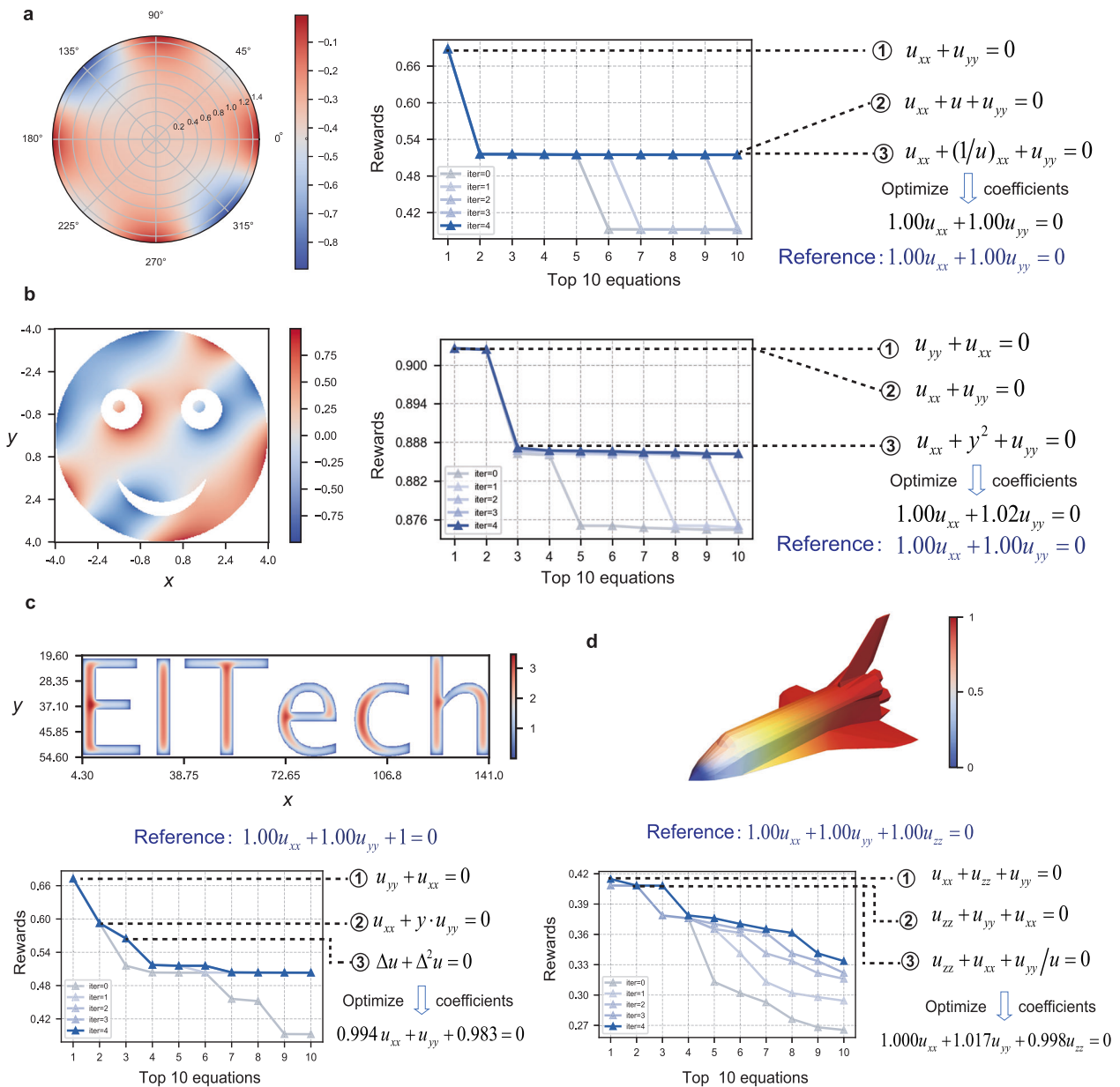
To better elucidate the performance of our proposed framework, a comparative analysis was conducted against existing PDE discovery algorithms. First, we focus on comparing the discovery scope of PDEs and discovery efficiency. Acknowledging the diversity of existing PDE discovery methods, several representative algorithms are selected for comparison: PDE-FIND<sup>5</sup>, the generalized genetic algorithm (GGA)<sup>31</sup>, DISCOVER<sup>11</sup>, PDE-READ<sup>28</sup>, W-SINDy<sup>7</sup>, LLM4ED<sup>16</sup>, WeakIdent<sup>8</sup>, and PINN-SR<sup>32</sup>. Here, PDE-FIND is essentially the PDE-specific version of SINDy. Eight canonical PDEs are examined, which involves typically-used baselines as well as complex-form PDEs recorded in the math handbook<sup>25</sup>. For each case, 10,000 discrete data with no noise are sampled for PDE discovery. To highlight the inherent differences between algorithms, the PDE discovery process was conducted on the metadata generated by the trained surrogate model, thereby avoiding the influence of data processing. For a fair comparison, the candidate library in sparse regression-based methods includes the combination of polynomial terms and high-order derivatives, and all relevant vocabularies consistent with those employed in the proposed method. This results in a comprehensive candidate library comprising 43 candidate terms. For EqGPT, the correct PDEs are excluded from the PDE dataset during the training process, thereby avoiding data leakage. EqGPT must rely solely on its learned representations of general PDE structures and term interactions to generate and optimize toward the correct form. The details of experimental settings are provided in Methods. The ability of existing methods and our proposed framework to discover these PDEs and corresponding time overhead are provided in Fig. 3a. The discovery outcomes are categorized into four classes. Results are deemed correctly discovered when the identified equation exactly matches the true form. If the correct terms are present but accompanied by additional terms, the result is classified as discovered with redundant terms. When the identified equation fails to include all of the true terms, it is considered incorrectly discovered. Finally, if a method is technically unable to address a given type of equation, the case is marked as not applicable.

The result reveals notable differences in the discovery scope across various methods. Sparse regression-based methods, like PDE-FIND, W-SINDy, and WeakIdent, have the highest computational efficiency. However, we find that they are usually not applicable to free-form terms, since the candidate library is practically restricted to a finite set of polynomial derivative combinations and cannot accommodate infinite free-form expressions, such as  $xu_{tt}$  and  $u_x/x$ . Expanding the candidate library to include broader forms would simultaneously increase the difficulty of sparse optimization and hyperparameter tuning, making it challenging to preserve parsimony and leading to the appearance of redundant terms. While W-SINDy and WeakIdent employ weak-form formulations to enhance robustness and accuracy, their reliance on FFT-based techniques makes them difficult to apply when PDEs are posed on complex or non-squared domains. In contrast, EqGPT can generate free-form terms by flexibly combining

operators and variables without the need of predefined candidate libraries, which is essential for identifying new PDE structures in practice. By contrast, symbolic regression methods can handle free-form terms<sup>33,34</sup>, however, the computational cost is extremely high, nearly 20 min for Eq. (8.14.1d) with the DISCOVER method. The low search efficiency also influences its ability to address more intricate terms. Although genetic algorithm-based methods, like GGA, can discover PDEs under simple compound form with higher computational efficiency, they are also incapable of handling complex terms that cannot be generated by cross-over and mutation. From the comparison, the LLM-based methods usually take a much larger computational time, which reflects the low efficiency of calling the general LLM by giving prompts. Meanwhile, the integration of PINNs with PDE discovery has emerged as a promising direction, demonstrating notable robustness and flexibility<sup>32,35,36</sup>. However, such approaches typically rely on an alternating direction optimization (ADO) scheme, as in PINN-SR, which leads to a high computational cost. Meanwhile, the success of the procedure depends on favorable initialization, which may fail to converge in complex problems. In contrast, EqGPT leverages a generative model fine-tuned on mathematical handbooks to directly propose free-form PDE structures, which allows it to generate plausible candidates efficiently and to converge rapidly, thereby offering superior flexibility and robustness. Generally, most existing approaches that fix the temporal derivative  $u_t$  as the LHS term struggle to accommodate mixed partial derivatives such as  $u_{xt}$  and  $u_{xxt}$ . In contrast, EqGPT does not require a pre-determined LHS term, which can adaptively adjust the equation structure during optimization.

Next, we focus on the stability and accuracy of EqGPT compared with existing methods under sparse and noisy data. For comparison, we select GP-SINDy<sup>37</sup>, which combines Gaussian processes with SINDy and has been shown to achieve higher robustness than most existing methods. We consider Burgers' equation with a diffusion coefficient of 0.5 as an example. Since GP-SINDy requires grid data, we construct datasets by subsampling to obtain grids of size  $41 \times 65$ ,  $41 \times 33$ , and  $41 \times 17$  (denoted as  $n_x \times n_y$ ), corresponding to 2665, 1353, and 697 data points, respectively. For each dataset, Gaussian noise with noise levels = 0.02, 0.1, 0.2, 0.3, 0.4, and 0.5 is added. In the GP-SINDy experiments, we include derivatives up to third order and polynomial terms up to second order, resulting in 15 candidate terms in the library. Detailed experimental settings are provided in Methods. Accuracy is evaluated for correctly discovered equations by the mean relative error of the recovered coefficients defined in Eq. (3). The results are presented in Fig. 3b. EqGPT demonstrates high stability and accuracy across sparse and noisy scenarios even with only 697 data points, which remains robust to noise levels up to 0.3. By contrast, GP-SINDy exhibits robustness to noise levels up to 0.2 when using 2665 or 1353 data points, which already exceeds many comparable algorithms<sup>5,38</sup>. However, it fails to maintain sparsity under higher noise, leading to incorrect results. With as few as 697 data points, GP-SINDy cannot tolerate any noise. Furthermore, even at the same noise levels, EqGPT achieves lower relative coefficient errors than GP-SINDy, owing to its flexible regression process and the denoising and optimization capability of physics-informed neural network. Notably, GP-SINDy can achieve improved performance with a smaller candidate library, when limited to second-order derivatives. However, this restriction reduces its ability to address diverse PDE structures and limits its practical applicability. In contrast, EqGPT is able to operate within a broad search space and still accurately recover equations from sparse and noisy data.

In the comparison against existing methods, it is found that EqGPT can achieve a balance between the search scope and optimization efficiency. For baseline PDEs, the time overhead of EqGPT may be slightly higher than the sparse regression and genetic algorithms. However, the total time overhead is acceptable only with tens of seconds. Meanwhile, EqGPT presents stability and efficiency on the



**Fig. 4 | Discovery of Poisson's equation within complex computed regions.** **a** The dataset on a disk region (left) and the rewards of the top 10 equations generated from the EqGPT model in each optimization epoch (right). **b** The dataset on a "smiley face region" (left) and the rewards of the top 10 equations in each epoch (right). **c** The dataset on a discontinuous glyph boundary region of "EITech"

(upper) and the rewards of the top 10 equations in each epoch (lower). **d** The dataset on the region of the 3-dimensional space shuttle (upper) and the rewards of the top 10 equations in each epoch (lower). A deeper blue color indicates a larger epoch. The top 3 structures in the final epoch are listed.

discovery of PDEs with complex temporal terms and intricate spatial terms. It also demonstrates superior stability and accuracy under sparse and noisy data. This capability is attributed to the integration of prior knowledge from established PDEs and the proposed GRE strategy, which enables the model to generate compact and plausible free-form PDEs, thereby facilitating more effective optimization during the discovery process.

**Identification of nonlinear dynamic systems in complex computed regions**

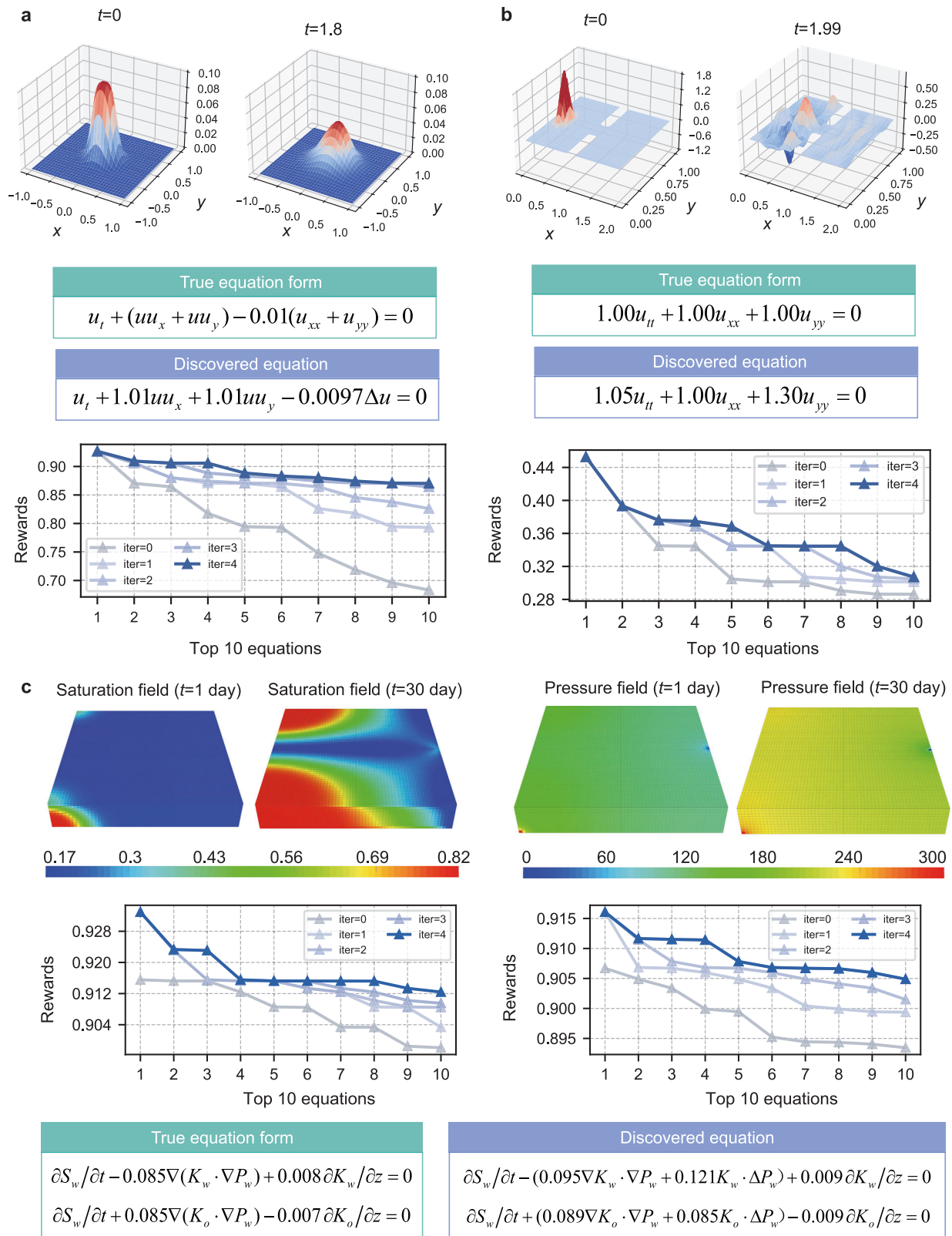
Boundary conditions and computed regions play important roles in simulating the behavior of dynamic systems. In real-world scenarios, the computed regions are often complex and irregular. However, previous studies have not explored the discovery of PDEs within complex regions. Typically, earlier research focused on cases

with regular regions, such as rectangular regions. Nevertheless, complex computed regions with irregular boundaries can amplify the challenge of discovering PDEs since the data complexity is escalated and the data continuity is reduced. In this analysis, we explore several scenarios featuring irregular computed regions to assess the performance of the proposed framework. Detailed information on these scenarios is provided in Supplementary Information S1.1. Poisson's equation with different boundary conditions is employed, which is written as:

$$\nabla^2 u = f. \tag{5}$$

If  $f = 0$ , the equation is also called the Laplacian equation.

Initially, a simple disk region with radius  $r = 1.5$  is adopted. The Dirichlet boundary condition is  $u(x,y) = x \sin(xy)$ . The solution dataset



**Fig. 5 | Discovery of high-dimensional PDEs via the proposed generative framework.** **a** For the 2-dimensional (2D) Burgers' equation, the surface of  $u(x,y)$  when  $t = 0$  and  $t = 1.8$ , the true PDE, the discovered PDE, and the rewards of the top 10 equations in each epoch. **b** For the vibration of an H-shaped elastic membrane, the physical field when  $t = 0$  and  $t = 1.99$ , the true PDE, the discovered PDE, and the

rewards of the top 10 equations in each epoch. **c** The saturation field and the pressure field of water when  $t = 1$  day and 30 day in the 3-dimensional oil–water two-phase displacement process. The true PDE, the discovered PDE, and the rewards of the top 10 equations when discovering the governing equation of water (left) and oil (right) phases in each epoch.

contains 40,200 observation data points on grids, as illustrated in Fig. 4a. The proposed framework is employed to discover the governing equation for the proof-of-concept. Notably, the target equation is deleted from the PDE dataset when training the generative model. Here, 30,000 discrete data points are randomly selected to train the surrogate model. The rewards of the top 10 equations in each optimization epoch are depicted in Fig. 4a. The reward of the best structure is dominantly higher than that of the other structures. The correct PDE structure has been discovered since the first epoch, which reflects the high optimization efficiency of the method. Furthermore, the precise alignment of the ultimately discovered PDE with the actual PDE attests to the method’s remarkable accuracy.

Then, an irregular region, called the “smiley face region”, is investigated. This region is carved out of the disk region with a radius of 2 to form a region of the smiley face. The Dirichlet boundary condition is as follows:  $u(x,y)=\sin(x+y)$ . The solution dataset contains 44,711 observation data points on grids, as illustrated in Fig. 4b. Similarly, 30,000 discrete data points are randomly selected for PDE discovery. The rewards of the top 10 equations in each optimization epoch are depicted in Fig. 4b. The results of EqGPT are almost unaffected by this irregular region, exhibiting only a minor variation in the coefficients of the discovered PDE.

For a more difficult scenario, a computed region that consists of discontinuous glyph boundaries is considered. As illustrated in Fig. 4c, Poisson’s equation with  $f=-1$  is solved in the “EITech” region. For each alphabet, the zero Dirichlet boundary condition, where  $u(x,y)=0$ , is adopted. This complex scenario is simulated via *Mathematica*<sup>39</sup>, and a dataset of 48,367 data points is obtained. A total of 45,000 discrete data points are randomly selected for PDE discovery, and the results are provided in Fig. 4c. The proposed framework can accurately identify the correct PDE structure and estimate the coefficients with overall precision. Although there is a relatively large deviation in the constant term  $f$  of the PDE, the coefficient error remains below 2%. This finding indicates that the proposed framework can handle scenarios with complex discontinuous regions.

Finally, a 3-dimensional (3D) scenario is investigated, in which the simulated region is a 3D space shuttle (Fig. 4d). The high-dimensional region is highly irregular and complex, which poses a great challenge to PDE discovery. Here, the simulation is conducted via *Mathematica*, and the Dirichlet boundary condition is as follows:  $u(x,y,z)=1$  if  $z \leq 1.3$  and  $u(x,y,z)=0$  if  $x \leq -7$ . The dataset contains 438,960 data points, of which 400,000 data points are employed for PDE discovery. The results are illustrated in Fig. 4d. The optimization process, while challenging, is ultimately effective. This is demonstrated by the gradual increase in the rewards of the top PDE structures with optimization epochs. The ultimately discovered PDE is closer to the true PDE, with a coefficient error of 0.63%. This substantiates the accuracy and stability of the proposed framework, even when confronted with high-dimensional complex regions.

### Extension to complex PDEs in high-dimensional space

The proposed framework is readily scalable to higher dimensions, given that the PDE dataset encompasses PDEs that range from one to three dimensions. Moreover, differential operators, such as the divergence operator ( $\nabla$ ), Laplacian operator ( $\Delta$ ), and even the BiLaplacian operator ( $\Delta^2$ ), are considered and can be generated via the EqGPT model. These operators are applicable for all dimensions, which improves the flexibility of the proposed framework. By specifying the target dimension, EqGPT can generate proper PDEs in the respective dimension. Here, experiments are conducted to examine the extension of the EqGPT framework to high-dimensional scenarios. Details of the utilized dataset can be found in Supplementary Information S1.1.

First, the discovery of Burgers’ equation in a 2-dimensional space is investigated, as depicted in Fig. 5a. This case serves as a typical benchmark for assessing the performance in uncovering high-

dimensional PDEs. The true PDE form is written as follows:

$$u_t + (uu_x + uu_y) - 0.01(u_{xx} + u_{yy}) = 0 \tag{6}$$

The dataset consists of the grid data of 101 spatial observation points in the domain  $x \in [-1,1]$ , 51 spatial observation points in the domain  $y \in [-1,1]$ , and 100 temporal observation points in the domain  $t \in [0,2)$ . The data size is 515,100. For PDE discovery, 200,000 data points are randomly selected to construct the surrogate model. The results of the proposed generative framework are illustrated in Fig. 5a. Evidently, it successfully discovers the correct PDE with a high accuracy of coefficients. Interestingly, the Laplacian operator ( $\Delta u$ ) occurs in the discovered PDE, which diminishes the number of terms and leads to a more parsimonious PDE. The optimization cycle in the framework is also efficient, where the rewards of the top structures in the final epoch are much higher than those in the initial epoch. Additional experiments indicate that the surrogate model performs reliably in high-dimensional settings and remains robust under data noise levels of up to 15% (Table S3).

We then consider a more difficult high-dimensional scenario, which has never been discovered in prior works. The physical scenario is the vibration of an H-shaped elastic membrane, where the governing equation can be written as:

$$u_{tt} + u_{xx} + u_{yy} = 0. \tag{7}$$

The initial conditions are  $u(0,x,y) = 2e^{-125(x-0.25)^2 + (y-0.5)^2}$  and  $u_t(0,x,y) = 0$ . The boundary condition is a zero Dirichlet boundary,  $u(t,x,y) = 0$ . The dataset contains 3,774,981 data points, where 300,000 data points (7.9%) are randomly selected for PDE discovery. As illustrated in Fig. 5b, the physical field exhibits intricate temporal dynamics, posing significant challenges for PDE discovery. Despite this, the proposed framework remains successful in identifying the PDE structure, although there are some deviations in the coefficients. Considering that only 7.9% of the total data points are utilized, the results are acceptable.

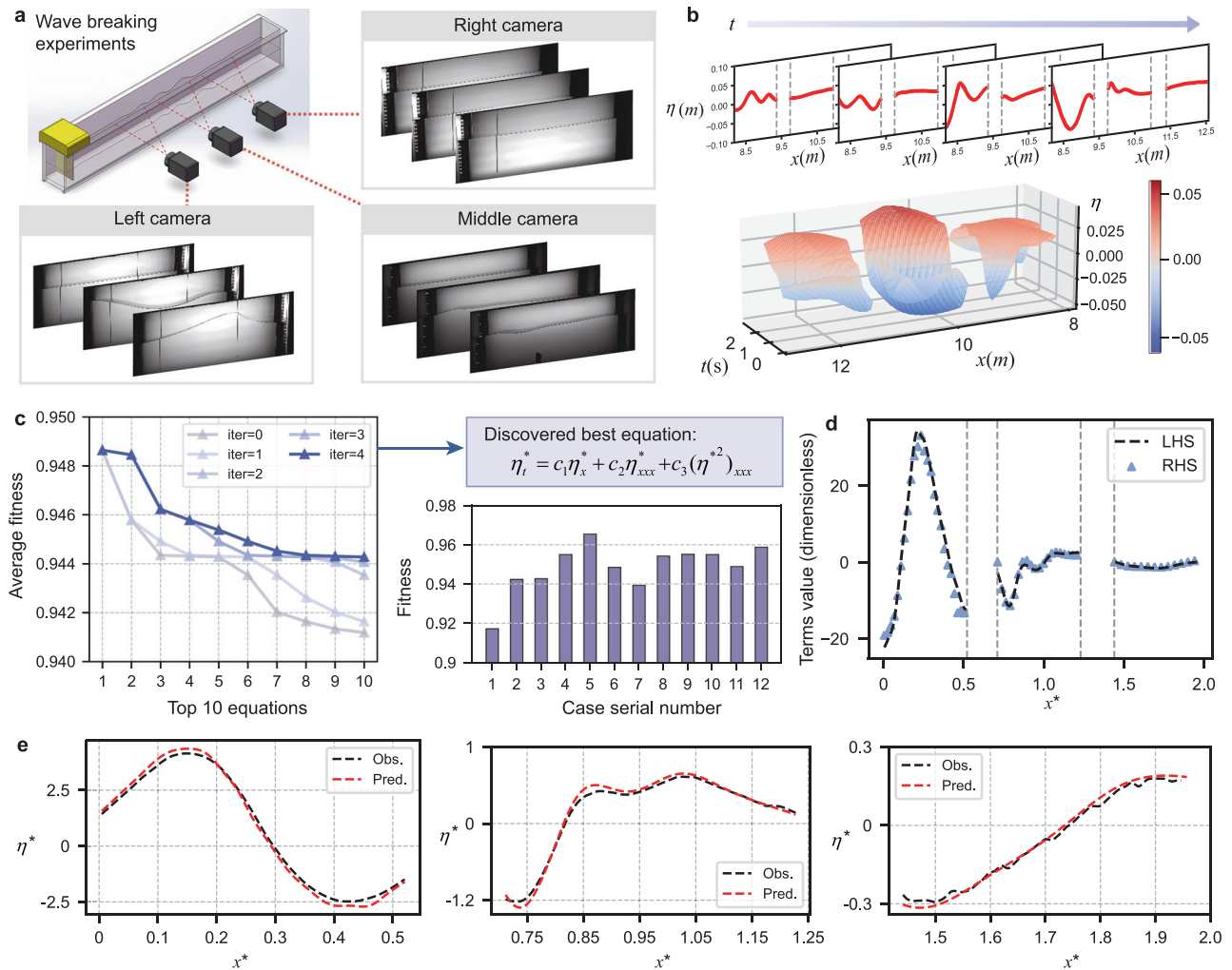
Finally, we consider a practical case study involving a three-dimensional multiphase flow system with multiple state variables. The scenario simulates an oil–water two-phase displacement process, where water is injected at the two corners on the left boundary, and oil is produced from the center of the right boundary (Fig. 5c). This physical process is governed by a system of coupled PDEs that separately describe the dynamics of the oil and water phases, which are written as:

$$\phi \frac{\partial S_w}{\partial t} - \alpha \nabla [K_w \cdot \nabla (P_w - \rho_w \mathbf{g})] = 0 \tag{8}$$

$$\phi \frac{\partial S_o}{\partial t} - \alpha \nabla [K_o \cdot \nabla (P_o - \rho_o \mathbf{g})] = 0 \tag{9}$$

where  $\phi=0.1$  denotes the porosity; the coefficient  $\alpha = 0.0085$ ;  $S_w$  and  $S_o$  are saturation fields for water and oil, respectively;  $P_w$  and  $P_o$  are pressure fields;  $K_w$  and  $K_o$  are the effective permeability fields that vary spatially and temporally; the gravitational acceleration  $\mathbf{g}$  acts only in the vertical ( $z$ ) direction and is set to 0.098; the water density  $\rho_w$  is set to be 1; the oil density  $\rho_o$  is 0.8; the two equations are coupled through the saturation constraint,  $S_w + S_o = 1$ ; and under the assumptions of incompressibility and negligible capillary pressure, the oil phase pressure equals the water phase pressure, i.e.,  $P_o = P_w$ . Therefore, the PDE system can be simplified as:

$$\frac{\partial S_w}{\partial t} - 0.085 \nabla (K_w \cdot \nabla P_w) + 0.008 \frac{\partial K_w}{\partial z} = 0 \tag{10}$$



**Fig. 6 | Discovery of previously undisclosed governing equations for highly nonlinear surface gravity waves propagating toward breaking from real-world experimental data.** **a** Experimental setup and the original records from the high-resolution, charge-coupled device cameras. The diagram is not to scale. **b** The visualization of the recognized water-air boundary. The blank areas between the gray dashed lines represent regions that could not be captured due to obstruction

by columns in the tank. **c** The discovered equation, the visualization of the optimization process, and the fitness in 12 independent experiments. **d** The comparison between the left-hand-side (LHS) term and the regressed right-hand-side term (RHS) of the discovered equation. **e** The observation and posterior prediction from the discovered governing equation in different regions.

$$\frac{\partial S_w}{\partial t} + 0.085 \nabla(K_o \cdot \nabla P_w) - 0.007 \frac{\partial K_o}{\partial z} = 0 \quad (11)$$

This physical system involves two primary state variables ( $S_w$  and  $P_w$ ) and two dependent state variables, as the effective permeabilities of the water and oil phases are correlated with the corresponding saturation fields. This reflects the complex and high-dimensional characteristics of real-world subsurface flow systems. Datasets are generated from the unconventional oil and gas simulator (UNCONG)<sup>40</sup>. The spatial grid dimensions are  $n_x = 50$ ,  $n_y = 50$ , and  $n_z = 8$ , with the simulation time ranging from day 0 to day 30, discretized into 30 time steps. Each physical variable yields 600,000 data points. During PDE discovery, potential terms associated with  $S_w$ ,  $K_w$ , and  $K_o$  are incorporated into the vocabulary, while terms related to  $P$  are constructed analogously to the single-variable setting. Spatial symmetry in the  $x$  and  $y$  directions is utilized to constrain the search space for the generated equations. All available data are used for the discovery of governing equations. Additional details about the experimental setting are provided in Supplementary Information S1.1. The PDE discovery process involves 600 populations with 5 optimization epochs. The resulting discovered PDEs are

written as:

$$\frac{\partial S_w}{\partial t} - (0.095 \nabla K_w \cdot \nabla P_w + 0.121 K_w \cdot \Delta P_w) + 0.009 \frac{\partial K_w}{\partial z} = 0 \quad (12)$$

$$\frac{\partial S_w}{\partial t} + (0.089 \nabla K_o \cdot \nabla P_w + 0.085 K_o \cdot \Delta P_w) - 0.009 \frac{\partial K_o}{\partial z} = 0 \quad (13)$$

Notably, the identified equations correspond to the expanded form of the ground-truth PDEs (Fig. 5c). The true governing equations include nested differential operators, which are difficult to recover directly without explicit prior knowledge. However, the proposed method effectively reconstructs these structures by generating free-form interaction terms. This result suggests that the framework is capable of unveiling underlying relationships among multiple state variables, even in three-dimensional space with time. Although the numerical coefficients of the discovered equation show slight deviations from the ground truth, the discrepancies remain within acceptable bounds given the complexity of the physical processes involved.

### Discovery of new PDEs from real-world experimental data

PDE discovery has long been anticipated to uncover previously undisclosed governing equations from actual experimental data. Therefore, we aim to evaluate the proposed method's ability to discover new equations and principles from real-world experimental data. As a practical example, we consider the discovery of the governing equation for highly nonlinear surface gravity waves propagating toward breaking. Despite the ubiquity of wave breaking in natural environments, its detailed physical and statistical mechanisms remain incompletely understood, making the modeling of such processes a significant challenge. To this end, we focus here on describing the evolution of unidirectional, irregular focused wave groups as they approach the onset of breaking. Particularly, there is currently no theoretical governing equation based on surface elevation at the air-water interface to describe this process towards an irreversible process, which hinders a deeper understanding and necessitates the data-driven discovery of potential governing equations.

The real-world data employed in this study are subsets from two experimental campaigns, namely BUBER and EURUS. Briefly, both campaigns were carried out in a 27.2-meter-long, glass-walled wave tank located at the Department of Civil and Environmental Engineering, Imperial College London. Breaking wave groups were generated using a bottom hinged wave paddle at one end of the tank, following the so-called dispersive-focusing technique and characterized by JONSWAP-type wave spectra<sup>41,42</sup>. Three high-resolution, charge-coupled device (CCD) cameras were placed outside of the flume to capture the propagation process over a spatial domain spanning approximately 8 meter to 12.5 meter (Fig. 6a), with a frame rate of 20 Hz. The spatial surface elevation was reconstructed at different time instants before the onset of breaking by employing the image-processing technique developed by Cao et al.<sup>43</sup> on a frame-by-frame basis, allowing for the air-water boundary to be identified (Fig. 6b). Due to visual occlusion caused by structural columns of the wave tank, the fields of view of the three cameras were spatially separated and did not overlap. As a result, Fig. 6b contains three distinct, non-contiguous spatial regions. Nevertheless, no explicit interpolation or reconstruction was applied to address the spatial gaps, as this was deemed beyond the scope of the present research. A total of 12 independent experiments were conducted with different initial conditions of the wave paddle, including the peak enhancement factor and peak wave period of the JONSWAP-type spectra, as well as the total amplitude summed across all underlying wave components. The total datapoints are 4,583,721 with approximately 380,000 for each experiment. Additional details about the experimental setting are provided in Supplementary Information S1.1. In the PDE discovery process, the data were non-dimensionalized:

$$x^* = (x - x_0)/\lambda; t^* = t/T_p; \eta^* = \eta/\lambda \quad (14)$$

where  $x_0$  is the beginning coordinate of the observed region;  $\lambda$  is the initial wavelength;  $T_p$  is the initial peak period; and  $\eta$  is the surface elevation. The surrogate models are constructed for data from each experiment, and standard EqGPT is applied to discover the underlying governing equation. For generated PDEs, the mean of reward calculated on the data from each independent experiment is utilized for evaluation. The optimization process is shown in Fig. 6c. The final discovered equation is:

$$\eta_t^* = c_1 \eta_x^* + c_2 \eta_{xxx}^* + c_3 (\eta^2)_{xxx}. \quad (15)$$

For different initial conditions, the coefficients may be slightly different (Table S4). As shown in Fig. 6c, this PDE performs well across all experiments. It is discovered that the left-hand side term ( $\eta_t^*$ ) and the regressed right-hand side terms of the PDE align well at the time of

wave breaking onset (Fig. 6d). The time-marching posterior prediction by the discovered PDE also closely matches the observed data, demonstrating its accuracy in representing this physical process (Fig. 6e). Notably, our method successfully uncovers governing equations under partial information since real-world experimental data are incomplete and contain missing regions.

From the analysis of fluid mechanics, the discovered PDE is physically reasonable, which incorporates the  $\eta_x^*$  term as the leading order propagation term. This is consistent with the formal mathematical analysis, which shows that the same leading order term with  $\eta_x^*$  can be derived for non-breaking water waves under the potential flow framework<sup>44</sup>. This indicates that the evolution of bulk water towards the breaking onset still mostly follows the classic wave theory, which agrees well with the observation that wave breaking is a strong but localized behavior. Apart from the classic term, we further observed a new functional form as  $(\eta^2)_{xxx}$ , where the triple partial  $x$  derivative can be found in the classic shallow water equation – KdV equation. The new form, however, instead of being a linear dispersive term, can give rise to harmonics at higher frequencies as the wave evolves towards a breaking event. This is superficially similar to the wave-breaking behaviors observed in previous studies, where the breaking events are found to be associated with higher energies at higher frequencies<sup>45,46</sup>.

Interestingly, this term occurs in the PDE dataset collected from the math handbook within a chapter introducing compactons, which was originally concerned with mathematical properties rather than physical applications. This finding reflects the effectiveness of our proposed strategy: the broad learning of PDE forms across diversified domains is beneficial for discovering new equations in specific fields. Absent such cross-domain knowledge, such a higher-order nonlinear term would be difficult to conceive and include in traditional methods' candidate libraries. This example demonstrates the importance of knowledge-guidance, and confirms the potential of our method to discover new principles and advance scientific understanding in real-world problems.

### Discussion

In this study, we explored a knowledge-guided approach for the discovery of partial differential equations (PDEs), representing a step toward integrating data-driven methodologies with domain-informed generation. This framework addresses a longstanding dilemma of balancing both search space and optimization efficiency in purely data-driven approaches. By incorporating prior knowledge extracted from mathematical handbooks, the proposed method autonomously generates and optimizes free-form PDEs with improved accuracy and robustness, even in challenging scenarios involving sparse or noisy data, complex computed regions, and high-dimensional domains. The practical example of discovering a new governing equation from real-world experiments demonstrates the practical utility of the method and underscores the core principle of this work: “stones from other hills may polish one's own jade”, i.e., extensive learning from existing equations can facilitate the identification of new governing laws. Further ablation studies confirm the positive impact of increasing the number and diversity of embedded PDE forms on model performance (Supplementary Information S1.3).

In the context of PDE discovery, the generation and optimization of potential equations are central aspects. Sparse regression restricts the search space to a predefined candidate library to improve search efficiency, limiting the diversity of admissible terms, whereas tree-based symbolic regression expands expressiveness by constructing free-form structures, but at the expense of optimization efficiency. To balance flexibility and efficiency, we adopt a generative approach in this work. Despite the structural diversity of PDEs, their constituent elements exhibit quantifiable regularities. Building on this observation, we introduce the generative representation of equations (GRE) strategy, which reformulates PDE structures as sentence-like sequences of

operators and basic terms. By embedding known equation forms into the learning process, EqGPT not only identifies plausible terms but also captures their co-occurrence patterns, thereby guiding discovery in a more structured manner. Notably, through combinations of operators and basic terms, EqGPT can generate infinitely many PDEs while preserving mathematical validity. Traditional optimization approaches include sparse optimization, which offers computational efficiency but relies on the predetermined candidate library, and genetic programming, which provides greater flexibility but incurs substantial computational cost. In contrast, EqGPT leverages the strengths of generative modeling by fine-tuning on the highest-rewarded sequences from each optimization epoch, thereby steering subsequent sampling toward equations that better match the data. Conceptually, this fine-tuning strategy is inspired by reinforcement learning, but it achieves superior computational efficiency and stability. Notably, although previous studies have also attempted to construct generative models for symbolic regression<sup>12,21,47</sup>, these models largely focused on algebraic expressions and lacked the capacity to handle differential operators, making them unsuitable for PDE discovery.

Nonetheless, this study has certain limitations. The model's performance is influenced by the quality and coverage of the training data derived from existing PDE sources. Expanding the dataset to include equations from a wider range of scientific domains may further enhance generalizability and reduce domain-specific biases. Future work will focus on enriching the PDE dataset and improving the model's adaptability to broader classes of physical systems. Beyond the current generation-evaluation-optimization loop, we wish to enable EqGPT to condition its generation on physical prompt with simple natural-language descriptors, such as “fluid”, “turbulence”, “force”, and “stable”, thereby moving from a generic start token toward genuinely domain-adaptive discovery. A promising way to incorporate domain knowledge is to extend EqGPT with multimodal information, aligning equations, data, language, and visual representations within a shared latent space. Given the stability, precision, and scalability of the proposed generative framework, it is promising for practical application in PDE discovery within the scientific community.

## Methods

### The construction of the PDE dataset from the math handbook

In this work, a dataset of PDE forms is established from the scientific handbook *Nonlinear Partial Differential Equations*<sup>25</sup>. In this handbook, a variety of common PDEs, which are commonly utilized in science and engineering, are introduced and analysed in detail. Here, PDEs that contain explicit terms are recorded in the PDE dataset. Each PDE in the dataset is recorded solely based on its structure, which means that PDEs with identical structures, but different coefficients, are considered identical within the dataset. Moreover, the equivalent forms of PDEs are differentiated by experts, which avoids repetition and improves the representativity of the dataset. PDEs are recorded in the form of a structure string, which is separated by commas. The conversion from mathematic form to structure string is currently accomplished manually, since humans can easily understand and separate the basic components in the PDE. In this work, 221 PDEs of different structures are recorded in the dataset. These PDEs can be decomposed and summarized into 56 basic components (i.e., vocabularies), including terms and operators, which serve as the foundation for the subsequent implementation of scientific-augmented training (SAT).

Considering that the sequence of terms in PDEs does not affect their essence, a data augmentation technique is adopted in this work to expand the size of the dataset for training models. In data augmentation, the commutative properties of addition and multiplication are leveraged to generate equivalent PDE structures simultaneously. The first method involves swapping the sequence within the terms. For example, the term  $u \times u_x$  equals  $u_x \times u$ , while the encoded vectors are different. The other is swapping the sequence between terms. For

example, the structure  $u \times u_x + u_{xx}$  equals  $u_{xx} + u \times u_x$ . By combining the two methods, all equivalent forms of the PDE can be obtained. Programming can be accomplished by identifying the operator of  $\times$  and  $+$  for exchanging the corresponding sequence. The PDE dataset consists of 221 different PDEs collected from math handbooks. To enable generative representation of equations, each PDE is expressed as a sentence-like structure composed of operators and terms. Since the semantic meaning of a PDE remains invariant under permutation of additive terms, for example,  $u_t + u_x + u_{xxx} = 0$  is equivalent with  $u_{xxx} + u_t + u_x = 0$ , we leverage this property to perform data augmentation. Specifically, for each PDE form, we generate 32 distinct sentence variants by randomly shuffling the order of additive terms. In cases in which a PDE admits more than 32 equivalent permutations, a random subset of 32 is selected; for PDEs with fewer permutations, additional samples are generated through random augmentation to reach a total of 32 sentence representations per equation. As a result, we obtain a training dataset comprising 7072 sentence-style entries that encode mathematical prior knowledge, which is subsequently used to train the EqGPT.

### Generative representation of equations (GRES)

The proposed GRE consists of a pivotal component within the framework, where a specialized method for representing PDEs is adopted. In the featurization procedure, vocabularies are first represented via sequential encoding. In addition, the start token (S) and end token (E) are also encoded to indicate the starting and ending of the sentence, respectively. Specifically, E is encoded to be 1, + is 2,  $\times$  is 3, / is 4, S is 5, and terms of the PDE are encoded sequentially after them. This encoding method separates the operators and terms, which facilitates the subsequent training and generating process. In this way, the PDEs can be converted into vectors. When training the EqGPT model, the PDEs are further transformed into a matrix through one-hot encoding of each vocabulary. The GRE establishes a direct correspondence between equations and vectors, enabling efficient interpretation of the generated vectors. As a result, the output vectors can be quickly decoded back into their corresponding PDE forms.

### The design of the EqGPT generative model

In the proposed PDE discovery framework, a generative model, called EqGPT, is established to learn from the PDE dataset and generate new PDEs. Its architecture is similar to that of GPT-2, which consists of multiple transformer blocks. The input training data are embedded into high-dimensional spaces through the embedding layer, including both token embeddings and positional embeddings. Then, the inherent features and co-occurrence probabilities between vocabularies in the PDE structure are extracted and learned in blocks of the transformer. The transformer is composed of a layer norm and a masked multi-head self-attention layer. The details of the implementation can be found in the literature<sup>26</sup>. The output of the model is the probability distribution of the next word, which has been processed by the Softmax function.

To support the design of EqGPT, we draw upon the theoretical foundations of autoregressive language models, particularly those instantiated by transformer-based architectures, such as GPT-2. At the core of these models lies the principle of next-token prediction conditioned on preceding context, which enables them to capture complex sequential dependencies in tokenized inputs. In the context of PDE discovery, we treat symbolic expressions of differential equations as sequences composed of discrete semantic units. This formulation allows us to repurpose the autoregressive modeling framework for learning the co-occurrence patterns of PDE terms. Specifically, EqGPT learns to model the conditional probability distribution of the next token  $x_i$  given the preceding tokens  $x_1, \dots, x_{i-1}$ , i.e.,  $P(x_i | x_1, \dots, x_{i-1})$ . This co-occurrence learning is facilitated by the self-attention mechanism, which dynamically weights context tokens according to their relevance to the prediction target. Although the EqGPT model is trained on a relatively small dataset compared to its parameter count, this

overparameterization does not necessarily lead to overfitting<sup>48</sup>. On the contrary, large models can exhibit better generalization by adapting their internal representations more flexibly, especially when paired with techniques, such as dropout, weight decay, and appropriate regularization. Moreover, the autoregressive nature of EqGPT enables it to generate new equation structures in a free-form yet syntactically valid manner, by sampling from the learned token distributions.

### Scientifically augmented training (SAT)

The SAT enables the EqGPT model to extract and utilize scientific knowledge from mathematical handbooks, thereby increasing its generative capacity and empowering it to produce PDE structures that adhere to established scientific principles. The sentence of the PDE is separated into different lengths of preceding fragments (tokens) for training. The training loss is the cross-entropy loss of the predicted next word from preceding tokens and the true word. The optimizer is Adam<sup>49</sup>, and the learning rate is  $10^{-4}$ . The batch size is 128, and the number of training epochs is 100. In the generation process, the start token (S) is first input into the trained model, and the next token is sampled from the predicted probability distribution. Here, a combination of probabilistic sampling and random sampling is adopted to improve the diversity of the generation. Specifically, several scientific constraints are imposed to guarantee the validity of the generated PDEs in the SAT process. First, the terms and operators are considered to occur at intervals. Specifically, if the preceding token is an operator ( $\text{index} < 6$ ), the next token must be a PDE term ( $\text{index} \geq 6$ ). Therefore, the probability distribution can be constrained according to the predicted location. For example, when the length of the input is odd, only the probabilities of terms are reserved, whereas the others are forced to be zero. Similarly, when the length of the input is even, only the probabilities of the operators are reserved. In this way, the PDE form is guaranteed to be scientifically proper, which avoids the occurrence of consecutive operators or terms. Furthermore, the predicted probability is tailored according to the input variables or dimensions. For example, if the input variables are  $x$  and  $t$ , the probabilities regarding  $y$  and  $z$  are forced to be zero. By employing this scientifically augmented method, the probability distribution can be automatically calibrated to accommodate problems of different dimensions.

When selecting the next token from the constrained probability distribution, the combination of probabilistic sampling and random sampling is adopted in the SAT. In addition, under a certain probability (20% in this work), the token is chosen by random selection of all possible valid tokens. This treatment enables the model to explore more diverse possibilities. When the end token (E) is sampled or the length of the output reaches a threshold (50 in this work), the generation process stops. The sentence between the start token and the end token is the generated PDE structure. Notably, in the generation process, even for the same input, the outputs can be different. To ensure dimensional consistency with the underlying physical problem, EqGPT employs a token masking strategy during the generation process. Specifically, the vocabulary tokens that are incompatible with the problem's dimensionality are masked out, and the probability distribution is renormalized over the remaining valid tokens. For example, in a one-dimensional time-dependent system, all tokens involving  $y$  or  $z$  derivatives (e.g.,  $u_y$ ,  $u_{zy}$ ) are masked, ensuring that only  $x$  and  $t$ -dependent terms can be generated. Similarly, for two-dimensional systems, all tokens involving  $z$  are excluded. In addition, composite operators such as divergence or Laplacian are automatically adapted to the specified dimensionality. This masking mechanism guarantees that the generated PDE structures remain physically meaningful with respect to the spatial and temporal dimensions of the problem.

### The optimization cycle for PDE discovery

The generative model, after being trained through SAT, is capable of generating a diverse range of valid PDE structures, thereby providing a

broad yet effective search space. An optimization cycle is constructed to fine-tune the generative model to realize the directed generation of PDEs that conform to the observation data. As illustrated in Fig. 1, the observation data, which may be sparse and noisy, are utilized to train a fully connected artificial neural network (ANN) that functions as a surrogate model. The ANN consists of 7 layers, including an input layer, 5 hidden layers, and an output layer. The input neurons are equal to the number of dimensions of the problem, the output neuron is 1, and the number of neurons in the hidden layer is 50. The activation functions are Sin or Rational functions<sup>28</sup>. The training epoch is 50,000, and the early stopping technique is adopted to avoid overfitting. In practice, for sparse and noisy datasets, training typically converged and terminated within 1000 to 2000 epochs, well before reaching the pre-defined limit. During this process, the validation error is monitored, and the epoch exhibiting the lowest validation error is identified as the optimal one.

After the surrogate model has been trained, abundant meta-data are generated on grids from model predictions, where the data noise is smoothed, and derivatives are calculated through automated differentiation. These meta-data are utilized to calculate the reward of the generated PDEs from the generative model. In this work, EqGPT generates 400 PDEs in each optimization cycle. The reward is calculated via Eq. (1). For a candidate PDE structure, for example  $u_{xx} + u_t + uu_x = 0$ , we designate the negative of the first term ( $-u_{xx}$ ) as the LHS, while the remaining terms ( $u_t$ ,  $uu_x$ ) form the RHS. The coefficients are obtained from the least squares regression on the values of LHS and RHS. Importantly, EqGPT generates equation structures with diverse LHS terms (rather than a pre-fixed  $u_t$ ), thereby increasing the degrees of freedom available to the regression. The calculated rewards are then sorted, and the top 10 PDEs with the highest rewards are selected as the inputs for fine-tuning the generative model (Fig. S1). In the fine-tuning process, the learning rate is  $10^{-5}$ , and the training epoch is 5. This means that fine-tuning will only slightly optimize EqGPT, increasing its propensity to generate PDE structures that closely resemble these top PDEs. Afterwards, the fine-tuned EqGPT model continues to generate 400 new PDEs, and the top 10 PDEs are selected similarly. Out of the top 10 equations from both the new and old rounds, totaling 20 equations, 10 equations with the highest rewards are selected to advance to the subsequent optimization round. The optimization cycle stops when the maximum number of iterations is reached, which is set to 5 in this work. The best structure in the last iteration is the discovered optimal PDE. Notably, the current framework can also provide alternative suboptimal structures by sorting the rewards in the final iteration. In terms of the computational cost, the generation process is fast, i.e., only seconds, and each optimization cycle takes 10–30 s on the CPU i9-14900 KF and GPU RTX 4090D.

When the optimal PDE structure has been discovered, the coefficients can be calculated through least squares regression and further optimized by a physics-informed neural network (PINN)<sup>50</sup>. In the optimization of coefficients, the discovered PDE is incorporated into the previously trained surrogate model to construct a PINN. In each training epoch of the PINN, the derivatives are calculated through automatic differentiation, and the coefficients of the PDE are updated via least squares regression. The loss of the PINN can be calculated as:

$$L_{PINN} = \lambda_{Data} MSE_{Data} + \lambda_{PDE} MSE_{PDE} \\ = \frac{\lambda_{Data}}{N} \sum_{i=1}^N (u - u_{pred})^2 + \frac{\lambda_{PDE}}{N} \sum_{i=1}^N (\theta_{LHS} - \vec{\xi} \theta_{RHS})^2 \quad (16)$$

where  $N$  is the number of training data;  $MSE_{Data}$  is the data loss, which is the same as that of the ANN; and  $MSE_{PDE}$  is the mean squared error calculated from the least squared regression of the PDE. In this work,

$\lambda_{Data}$  is set to 1, and  $\lambda_{PDE}$  is set to 0.01. Then, the PINN is optimized by the loss function. The maximum optimization epoch for the PINN is 500, and the calculated coefficients in the final epoch are the optimal coefficients. Through PINN optimization, the accuracy of the discovered PDEs can be further improved<sup>31</sup>.

### The experimental settings

In this work, some experiments are conducted for proof-of-concept. In the discovery of canonical PDEs with sparse and noisy data, the number of meta-data is 10,000, and the input variables are  $x$  and  $t$ . When discovering each PDE, the target structure is deleted from the dataset, and the EqGPT model is retrained by the modified dataset. This guarantees that the true structure has never been trained by the generative model. The penalty strength  $\alpha_0$  in Eq. (1) is 0.2.

In the comparison with existing PDE discovery methods, experiments are conducted on the meta-data generated from the surrogate model. For the sparse-regression based algorithm, the candidate library comprises the combination of 3<sup>rd</sup>-order polynomial terms and up to 4<sup>th</sup>-order derivatives, and all relevant vocabularies consistent with those employed in the proposed method. Considering that the target canonical PDEs are both in one-dimensional (1D) space, the vocabularies relevant to the variable  $y$  and  $z$  are omitted. Therefore, this results in a comprehensive library comprising 43 candidate terms.

$$\Phi = [u, u_x, u_{xx}, u_{xxx}, u_{xxxx}, \dots, u^3 u_{xxx}, u_{xt}, u_{xxt}, x, t, \dots, (uu_x)_x, (u_x/u^2)_x] \quad (17)$$

The maximum number of iterations is 250,  $\lambda$  is 2, and  $d_{tol}$  is 0.1. The  $l_0$  penalty strength ranges from 0 to 1 in increments of 0.05. For the GGA algorithm, the basic genes are defined as  $u$ ,  $u_x$ ,  $u_{xx}$ , and  $u_{xxx}$ , the number of populations is 400, the maximum generation of evolution is 200, the mutation rate is 0.3, and the  $l_0$  penalty strength is  $3 \times 10^{-4}$ . For the DISCOVER algorithm, the considered operators are plus, minus multiply, and divide. The threshold of the reserved expressions is 0.02, the coefficient of entropy loss is 0.03, the coefficient of policy gradient loss is 1, the maximum number of function terms is 6, the total number of generated expressions at each iteration is 500, and the maximum number of iterations is 100. For other algorithms, the parameter setting is the same as the codes provided in the open source.

In the comparison with GP-SINDy, we include derivatives up to third order and polynomial terms up to second order to establish the candidate library, which is denoted as:

$$\Phi_{GP-SINDy} = \left[ \begin{array}{c} 1, u, u_x, u_{xx}, u_{xxx}, u^2, uu_x, uu_{xx}, uu_{xxx}, u_x^2, u_x u_{xx}, \\ u_x u_{xxx}, u_{xx}^2, u_{xx} u_{xxx}, u_{xxx}^2 \end{array} \right] \quad (18)$$

The noise is added as:  $\sigma = \sigma_{NR} \times \|U\|_F / \sqrt{N_{data}}$ , where  $\sigma_{NR}$  is the noise level.  $\|U\|_F$  represents the Frobenius norm of the clean data matrix  $U$ .  $N_{data}$  is the number of observation data. For EqGPT, the penalty strength  $\alpha_0$  is 0.3.

For the discovery of PDEs in complex regions, the input variables are  $x$  and  $y$  for 2D cases and  $x$ ,  $y$ , and  $z$  for 3D cases. The penalty strength  $\alpha_0$  is 1. Given the irregularity of the regions in these instances, the meta-data are generated at each observation point, excluding the boundaries. For the discovery of PDEs with high dimensions, the input variables are  $x$ ,  $y$ , and  $t$ . The penalty strength  $\alpha_0$  is 1.

### Data availability

The dataset generated in this study has been deposited in the GitHub repository, <https://github.com/woshixuhao/EqGPT/tree/main/data>, the Zenodo DOI is 10.5281/zenodo.17211769<sup>31</sup>.

### Code availability

All of the original code has been deposited at the website, <https://github.com/woshixuhao/EqGPT/tree/main/code>, the Zenodo is <https://doi.org/10.5281/zenodo.17211769>.

### References

- Zanna, L. & Bolton, T. Data-driven equation discovery of ocean mesoscale closures. *Geophys. Res. Lett.* **47**, e2020GL088376 (2020).
- Beetham, S., Fox, R. O. & Capecelatro, J. Sparse identification of multiphase turbulence closures for coupled fluid-particle flows. *J. Fluid Mech.* **914**, A11 (2021).
- Brunton, S. L. & Kutz, J. N. Promising directions of machine learning for partial differential equations. *Nat. Comput. Sci.* **4**, 483–494 (2024).
- Brunton, S. L., Proctor, J. L., Kutz, J. N. & Bialek, W. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci. USA* **113**, 3932–3937 (2016).
- Rudy, S. H., Brunton, S. L., Proctor, J. L. & Kutz, J. N. Data-driven discovery of partial differential equations. *Sci. Adv.* **3**, 1–7 (2017).
- Stephany, R. & Earls, C. Weak-PDE-LEARN: a weak form based approach to discovering PDEs from noisy, limited data. *J. Comput. Phys.* **506**, 112950 (2024).
- Messenger, D. A. & Bortz, D. M. Weak SINDy for partial differential equations. *J. Comput. Phys.* **443**, 110525 (2021).
- Tang, M., Liao, W., Kuske, R. & Kang, S. H. WeakIdent: weak formulation for identifying differential equation using narrow-fit and trimming. *J. Comput. Phys.* **483**, 112069 (2023).
- Xu, H., Chang, H. & Zhang, D. DLGA-PDE: discovery of PDEs with incomplete candidate library via combination of deep learning and genetic algorithm. *J. Comput. Phys.* **418**, 109584 (2020).
- Chen, Y., Luo, Y., Liu, Q., Xu, H. & Zhang, D. Symbolic genetic algorithm for discovering open-form partial differential equations (SGA-PDE). *Phys. Rev. Res.* **4**, 023174 (2022).
- Du, M., Chen, Y. & Zhang, D. DISCOVER: deep identification of symbolically concise open-form partial differential equations via enhanced reinforcement learning. *Phys. Rev. Res.* **6**, 013182 (2024).
- Sun, F., Liu, Y., Wang, J.-X. & Sun, H. Symbolic physics learner: discovering governing equations via Monte Carlo tree search. *arXiv preprint arXiv:2205.13134* (2022).
- Kamienny, P.-A., Lample, G., Lamprier, S. & Virgolin, M. Deep generative symbolic regression with monte-carlo-tree-search. in *International Conference on Machine Learning* 15655–15668 (PMLR, 2023).
- Makke, N. & Chawla, S. Interpretable scientific discovery with symbolic regression: a review. *Artif. Intell. Rev.* **57**, 2 (2024).
- Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process Syst.* **30**, 1–11 (2017).
- Du, M., Chen, Y., Wang, Z., Nie, L. & Zhang, D. LLM4ED: Large Language Models for Automatic Equation Discovery. *arXiv preprint arXiv:2405.07761* (2024).
- Shojaee, P., Meidani, K., Gupta, S., Farimani, A. B. & Reddy, C. K. Llm-sr: Scientific equation discovery via programming with large language models. *arXiv preprint arXiv:2404.18400* (2024).
- Li, Y. et al. Mllm-sr: Conversational symbolic regression base multi-modal large language models. *arXiv preprint arXiv:2406.05410* (2024).
- Ye, Z. et al. Pdeformer: Towards a foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2402.12652* (2024).
- Sun, J., Liu, Y., Zhang, Z. & Schaeffer, H. Towards a foundation model for partial differential equations: multi-operator learning and extrapolation. *Phys. Rev. E* **111**, 035304 (2024).

21. Chen, J. et al. Symbol: Generating flexible black-box optimizers through symbolic equation learning. *arXiv preprint arXiv:2402.02355* (2024).
22. Mundhenk, T. N. et al. Symbolic regression via neural-guided genetic programming population seeding. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, 24912–24923 (2021).
23. Landajuela, M. et al. A unified framework for deep symbolic regression. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 33985–33998 (2022).
24. Valipour, M., You, B., Panju, M. & Ghodsi, A. SymbolicGPT: a generative transformer model for symbolic regression. *arXiv preprint arXiv:2106.14131* (2021).
25. Debnath, L. *Nonlinear Partial Differential Equations for Scientists and Engineers* (Springer, 2005).
26. Radford, A. et al. Language models are unsupervised multitask learners. *OpenAI Blog* **1**, 9 (2019).
27. Xu, H., Chang, H. & Zhang, D. DL-pde: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data. *Commun. Comput. Phys.* **29**, 698–728 (2021).
28. Stephany, R. & Earls, C. PDE-READ: human-readable partial differential equation discovery using deep learning. *Neural Netw.* **154**, 360–382 (2022).
29. Zhang, S. & Lin, G. Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* **474**, (2018).
30. Xu, H., Zeng, J. & Zhang, D. Discovery of partial differential equations from highly noisy and sparse data with physics-informed information criterion. *Research* **6**, 1–13 (2023).
31. Xu, H. & Zhang, D. Robust discovery of partial differential equations in complex situations. *Phys. Rev. Res.* **3**, 033270 (2021).
32. Chen, Z., Liu, Y. & Sun, H. Physics-informed learning of governing equations from scarce data. *Nat. Commun.* **12**, 1–13 (2021).
33. Petersen, B. K. et al. Deep symbolic regression: Recovering mathematical expressions from data via risk-seeking policy gradients. *arXiv preprint arXiv:1912.04871* (2019).
34. Udrescu, S. M. & Tegmark, M. AI Feynman: a physics-inspired method for symbolic regression. *Sci. Adv.* **6**, eaay2631 (2020).
35. Thanasutives, P., Morita, T., Numao, M. & Fukui, K. Noise-aware physics-informed machine learning for robust PDE discovery. *Mach. Learn. Sci. Technol.* **4**, 015009 (2023).
36. Xu, H. & Zhang, D. Robust discovery of partial differential equations in complex situations. *Phys. Rev. Res.* **3**, 1–12 (2021).
37. Meng, Y. & Qiu, Y. Sparse discovery of differential equations based on multi-fidelity Gaussian process. *J. Comput. Phys.* **523**, 113651 (2025).
38. He, Y., Kang, S.-H., Liao, W., Liu, H. & Liu, Y. Robust identification of differential equations by numerical techniques from a single set of noisy observation. *SIAM J. Sci. Comput.* **44**, A1145–A1175 (2022).
39. Wolfram, S. *The Mathematica Book* (Wolfram Media, 2003).
40. Li, X., Zhang, D. & Li, S. A multi-continuum multiple flow mechanism simulator for unconventional oil and gas recovery. *J. Nat. Gas. Sci. Eng.* **26**, 652–669 (2015).
41. Cao, R., Padilla, E. M. & Callaghan, A. H. The influence of bandwidth on the energetics of intermediate to deep water laboratory breaking waves. *J. Fluid Mech.* **971**, A11 (2023).
42. Rapp, R. J. & Melville, W. K. Laboratory measurements of deep-water breaking waves. *Philos. Trans. R. Soc. Lond. Ser. A, Math. Phys. Sci.* **331**, 735–800 (1990).
43. Cao, R., Padilla, E. M., Fang, Y. & Callaghan, A. H. Identification of the free surface for unidirectional nonbreaking water waves from side-view digital images. *IEEE Journal of Oceanic Engineering* (2024).
44. Taylor, P. H., Tang, T., Adcock, T. A. A. & Zang, J. Transformed-FNV: Wave forces on a vertical cylinder—a free-surface formulation. *Coastal Engineering* **189**, 104454 (2024).
45. Liu, P. C. Wavelet spectrum analysis and ocean wind waves. in *Wavelet Analysis and its Applications* Vol. 4 151–166 (Elsevier, 1994).
46. Jinshan, X., Jiwei, T. & Enbo, W. The application of wavelet transform to wave breaking. *Acta Mech. Sin.* **14**, 306–318 (1998).
47. Kim, J. T., Landajuela, M. & Petersen, B. K. Distilling Wikipedia mathematical knowledge into neural network models. *arXiv preprint arXiv:2104.05930*. (2021).
48. Bornschein, J., Visin, F. & Osindero, S. Small data, big decisions: Model selection in the small-data regime. in *International conference on machine learning* 1035–1044 (PMLR, 2020).
49. Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015).
50. Raissi, M., Perdikaris, P. & Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
51. Xu, H. Data and codes for EqGPT. *Zenodo* <https://doi.org/10.5281/zenodo.17211769> (2025).

## Acknowledgements

This work was supported and partially funded by the National Natural Science Foundation of China (grant no. 52288101, 12501744, and 12572266), the China Postdoctoral Science Foundation (grant no. 2024M761535), the China National Postdoctoral Program for Innovative Talents (Grant No. BX20250063), and the National Key Research and Development Program 2024YFF1500600. The experimental results used here were funded by the UK Natural Environment Research Council (grant no. NE/T000309/1) awarded to A.H.C. This work was supported by the High Performance Computing Centers at the Eastern Institute of Technology, Ningbo, and the Ningbo Institute of Digital Twin. The authors express gratitude to Prof. Michael Rogers for his selfless assistance in simulating PDEs via Mathematica.

## Author contributions

H.X., Y.C., and D.Z. conceived the idea, designed the study, and analyzed the results. H.X. developed the algorithm, performed the computations, and generated the results and figures. M.D. participated in the investigation. A.C. obtained funding for and conceived the real-world experiments at Imperial College London. R.C., supervised by A.C., conducted the experiments. R.C., A.C., and T.T. analyzed the discovered equations from fluid mechanisms. J.L. performed the two-phase-flow simulations. Y.C. and D.Z. supervised the project. All of the authors contributed to the writing and editing of the manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41467-025-65114-2>.

**Correspondence** and requests for materials should be addressed to Yuntian Chen or Dongxiao Zhang.

**Peer review information** *Nature Communications* thanks the anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available”.

**Reprints and permissions information** is available at <http://www.nature.com/reprints>

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025