Physics-Guided Discovery of Highly Nonlinear Parametric Partial Differential Equations

Yingtao Luo[1], **Qiang Liu**[2,3], **Yuntian Chen**[4], **Wenbo Hu**[5], **Jun Zhu**[6,*]
[1]Carnegie Mellon University
[2]Institute of Automation, Chinese Academy of Sciences
[3]Eastern Institute for Advanced Study, Yongriver Institute of Technology
[4]Department of Computer Science and Technology, Hefei University of Technology
[5]Dept. of Comp. Sci. & Tech., Institute for AI, BNRist Lab,
Bosch-Tsinghua Joint ML Center, Tsinghua University
yingtaoluo@cmu.edu,qiang.liu@nlpr.ia.ac.cn,
ychen@eias.ac.cn,i@wbhu.net,dcszj@mail.tsinghua.edu.cn

Abstract

Partial differential equations (PDEs) fitting scientific data can represent physical laws with explainable mechanisms for various mathematically-oriented subjects. The data-driven discovery of PDEs from scientific data thrives as a new attempt to model complex phenomena in nature, but the effectiveness of current practice is typically limited by the scarcity of data and the complexity of phenomena. Especially, the discovery of PDEs with highly nonlinear coefficients from low-quality data remains largely under-addressed. To deal with this challenge, we propose a novel physics-guided learning method, which can not only encode observation knowledge such as initial and boundary conditions but also incorporate the basic physical principles and laws to guide the model optimization. We empirically demonstrate that the proposed method is more robust against data noise and sparsity, and can reduce the estimation error by a large margin; moreover, for the first time we are able to discover PDEs with highly nonlinear coefficients.

1 Introduction

Partial differential equations (PDEs) are ubiquitous in many areas, such as physics, engineering, and finance. PDEs are highly concise and understandable expressions of physical mechanisms, which are essential for deepening our understanding of the world and predicting future responses. The discovery of some typical PDEs is considered as milestones of scientific advances, such as the Navier-Stokes equations and Kuramoto–Sivashinsky equations in fluid dynamics, the Maxwell's equations and Helmholtz equations in electrodynamics, and the Schrödinger's equations in quantum mechanics. Nevertheless, there are still a lot of unknown complex phenomena in modern science such as the micro-scale seepage and turbulence governing equations that await PDEs for description.

Traditionally, PDEs are mainly discovered by: 1) mathematical derivation based on physical laws or principles (e.g., conservation laws and minimum energy principles); and 2) analysis of experimental observations. With the increasing dimensions and nonlinearity of the physical problems to be solved, the PDE discovery is becoming increasingly challenging, which motivates people to take advantage of machine learning methods. Pioneering works [1, 2] use symbolic regression to reveal the differential equations that govern nonlinear dynamical systems without using any prior knowledge. More recently, the representative SINDy[3] and STRidge [4] algorithms are proposed assuming that the dynamical

*Corresponding Author

NeurIPS 2022 AI for Science Workshop.

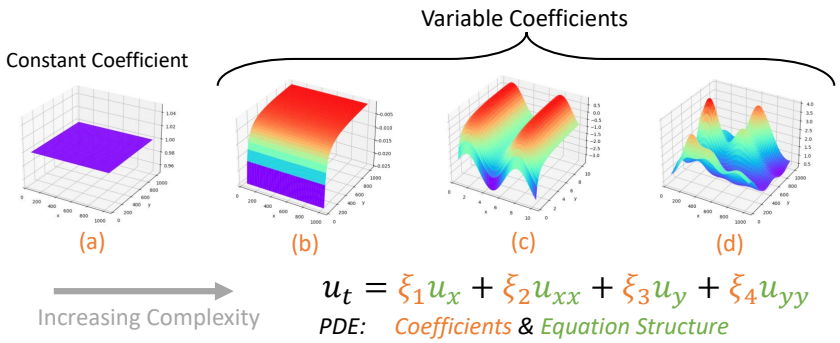

Figure 1: Schematic diagram of PDE coefficients. From left to right, the complexity of coefficient fields increases: (a) a constant value 1, (b) $-1/x$, (c) $-1/x + \sin y$, (d) complex coefficient field described by the Karhunen-Loève expansion of plenty of smooth basis functions [11, 12].

systems are essentially controlled by only few dominant terms. Through sparse regression, feature selection from candidate terms is performed to estimate the PDE model [5, 6]. Further attempts make use of observation knowledge such as boundary conditions of PDEs [7–9] and low-rank property of scientific data [10], which greatly reduce the amount and quality of data needed for PDE discovery.

Although the aforementioned works show promise in discovering PDEs with constant coefficients (PDEs-CC) as shown in Fig. 1(a) and some simple instances of parametric PDEs (PDEs with variable coefficients, PDEs-VC) as shown in Fig. 1(b)-(c), they do not yet suffice to discover more complex PDEs (e.g., PDEs with highly nonlinear coefficients) from scarce and noisier data. An example of highly nonlinear coefficients is the permeability random field [11, 12] shown in Fig. 1(d) for the spatial derivative terms in the PDEs of the seepage problems. Moreover, the PDEs obtained purely based on data-driven methods can only minimize the estimation error, but these methods may not consider the satisfaction of physical principles, such as the conservation of energy, momentum, etc.

To address these challenges, we rethink how the traditional PDE discovery works. Based on physical principles, scientists ensure that a newly discovered PDE aligns with the physical world. For example, the Navier-Stokes (NS) equation originates from the conservation of momentum, thus each term can relate to a certain physical meaning like convective accumulation or viscous momentum. Inspired by this, we propose a physics-guided learning framework that not only uses observation knowledge such as initial conditions and assumed terms for certain problems but also uses basic physical principles that are universal in nature as learning constraints to guide model optimization. Under this framework, a spatial kernel sparse regression model is proposed considering the principles of smoothness (as a first principle in PDEs) and conservation to impose smoothing of adjacent spatial coefficients for discovering PDEs with highly nonlinear coefficients.

Experimental results demonstrate that the proposed method can increase the overall accuracy of PDE estimation and the model robustness by a large margin. In particular, we consider the discovery of PDEs of different structure complexities with comparisons to baselines. Our method can discover the PDE structures of all instances that align well with the existing physical principles, while other baselines yield false equation structures for some complex PDEs with excessively high estimation errors. In summary, our contributions are:

- We propose a novel physics-guided framework for discovering PDEs from sparse and noisy data, which not only encodes observation knowledge but also incorporates physical principles to decrease errors and alleviate data quality issues. We propose a spatial kernel sparse regression model that considers conservation and differentiation principles. It presents excellent robustness in spite of noise compared to previous baselines, and can apply to sparse data in continuous spaces without fixed grids.
- We report experiments on representative datasets of nonlinear systems with comparison to strong baselines. The results show that our method has a lower coefficient estimation error and can discover all the test PDEs with variable coefficients even when the data is extremely noisy while previous baselines cannot. We also show that the discovered PDEs align well with existing physical principles and can reflect physical meanings.

2 Preliminary

2.1 Problem description

A physical field dataset $u(x, y, t)$ is defined with respect to some input coordinates (x, y, t), where $x \in [1, ..., n]$ and $y \in [1, ..., m]$ are spatial coordinates and $t \in [1, ..., h]$ is a temporal coordinate. An example of physical field data is shown in the observation data in Fig. 3. We consider the task of discovering two kinds of PDEs: (1) PDEs with constant coefficients, PDEs-CC; and (2) PDEs with variable coefficients, PDEs-VC. For simplicity, partial derivative terms are denoted by forms like u_x and u_{xx}, which are equivalent to $\frac{\partial u}{\partial x}$ and $\frac{\partial^2 u}{\partial x^2}$. The time derivatives such as u_t (i.e., $\frac{\partial u}{\partial t}$) of a PDE nearly always exist [13], therefore we follow prior works and set u_t as the regression label. Let p denote the number of partial derivative candidate terms considered in the task.

Definition 1 (PDEs with constant coefficients, PDEs-CC)**.** *PDEs-CC are the simplest PDEs, whose coefficients ξ_i are fixed along all coordinates:*

$$u_t = \sum_{i=1}^{p} \Theta(u)_i \xi_i, \quad \Theta(u)_i \in [1, u, u_x, u_y, u_{xx}, ..., uu_x, ...]. \tag{1}$$

Definition 2 (PDEs with variable coefficients, also known as parametric PDEs or PDEs-VC)**.** *The coefficients of PDEs-VC are changing in some dimensions, e.g., the spatial dimensions:*

$$u_t = \sum_{i=1}^{p} \Theta(u)_i \xi_i(x, y), \quad \Theta(u)_i \in [1, u, u_x, u_y, u_{xx}, ..., uu_x, ...]. \tag{2}$$

A simple example of explicit function is $\xi_i(x, y) = \sin x + \cos y$ and other $\xi_i(x, y)$ may be anistropic random fields [14] that are hard to express by explicit functions.

We can see that a PDE has two parts: the set of $\Theta(u)_i$ for $\forall i$ is the PDE structure, while the set of $\xi_i(x, y)$ for $\forall i$ is the PDE coefficients. Here, each $\Theta(u)_i$ represents a monomial basis function of u or the combination of two monomial basis functions of u. We consider monomial basis functions only up to the third derivative since higher-order derivatives can be inaccurate due to differential precision [4]. In Eqs.(1-2), the coefficient $\xi(x, y)$ changes w.r.t. spatial coordinates x and y. In this paper, we discuss the case of spatial variations. If the task is to capture variations in the temporal dimension, we can simply replace $\xi(x, y)$ with $\xi(t)$.

Accordingly, the goal of PDE discovery is to determine:

- **Terms**: which coefficient ξ_i is nonzero so that the term $\Theta(u)_i$ exists in the PDE structure;
- **Coefficients**: the exact values of all nonzero coefficients at each spatial coordinate.

Naturally, the accuracy of coefficient estimation would affect the correctness of determining which coefficient is nonzero. This coupling motivates us to choose methods that can perform structure learning and coefficient estimation simultaneously (e.g., sparse regression). Moreover, since the simplicity of PDE is important, we are looking for the PDE with the fewest terms. For example, $u_t = u_x$ is simpler than $u_t = u_x + u_y$ under similar data fitting.

2.2 Sparse Regression for PDE Discovery

Sparse regression is widely adopted in previous works to estimate both the terms and coefficients of PDEs. For parametric PDEs with variable coefficients across spatial dimensions, many linear regressions are separately performed for coefficients at different spatial coordinates (x, y):

$$Y = XW + \epsilon, \quad \epsilon \sim \eta\mathcal{N}(0, \sigma^2) \in \mathbb{R}^h, \tag{3}$$

$$\widehat{W} = \underset{W}{\operatorname{argmin}} \|Y - XW\|_2^2 + \lambda \|W\|_2^2, \tag{4}$$

where $Y = [Y_1, Y_2, ..., Y_h]^\top \in \mathbb{R}^h$ denotes u_t of all the h samples along the temporal dimension, X_{ji} denotes $\Theta(u)_i$ of the j-th sample in $X \in \mathbb{R}^{h \times p}$, $W = [W_1, W_2, ..., W_p]^\top \in \mathbb{R}^p$ denotes all the coefficients ξ_i of the p candidate terms, and ϵ denotes the inevitable noise in data. The above expression describes the scheme where we aim at discovering one PDE from one physical field u, which can also extend to the discovering of multiple PDEs from multiple physical fields. Here, Eq.3 and Eq.4 repeat $n \times m$ times along the spatial dimensions x and y to get every $\widehat{W}^{[x,y]}$.

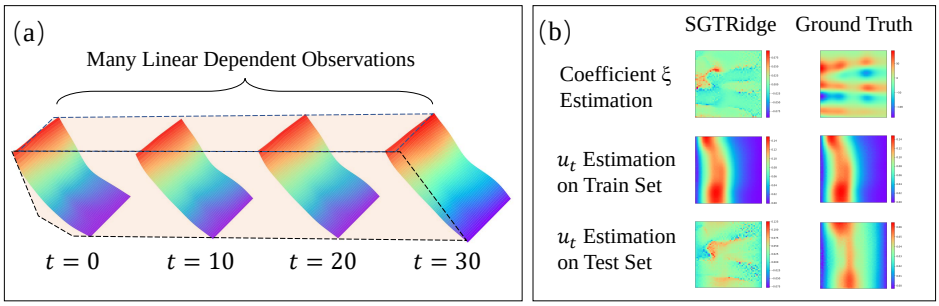

Figure 2: The linear dependent observations and data quality issues cause the overfitting of baselines such as SGTRidge. The estimated PDE coefficients are fairly irregular and cannot match the ground truth. Although it can fit the training data well, it fails to generalize to the test data.

2.3 The Challenge of Coefficient Estimation

Though the above methods have been effective for some PDEs with simple variable coefficients [5, 15, 16, 13, 6], they still have difficulty in discovering PDEs with highly nonlinear coefficients due to overfitting. To illustrate this, we use the mean absolute error (MAE) to measure the error of target (u_t) fitting across training, development, and test sets. With the correctness of PDE structure and accurate coefficient estimation, we shall obtain low target fitting MAE on test sets. As shown in Fig. 2 (a) and extensively mentioned in the literature [4, 14, 10], many physics observations are linearly dependent along the temporal dimension since the coefficient fields that determine the observation are not changing along time. Linear-dependent observations make the linear equation $Y = XW$ with $\mathrm{rank}(X) \leq p$ an underdetermined system that causes overfitting. Furthermore, data sparsity and noise also impair the data quality and exacerbate the problem. Fig. 2 (b) shows that the estimated coefficients by baseline sparse regression models such as SGTRidge [5] are irregular and cannot match the ground truth, and the estimation of the target u_t cannot generalize to test sets. The overfitting deviates the model from searching for the correct coefficients and terms, despite its good performance on the training set. Data details of Fig. 2 are shown in Section 4.3 and Appendix D.

3 Physics-Guided Spatial Kernel Estimation

While various PDE terms and coefficients could overfit the training data, scientists are only interested in the PDE that is interpretable in terms of physics and can stably describe the natural phenomena. In this paper, we incorporate physical principles into the PDE learning model. First, we consider smoothness [11, 14], which is a first principle as PDEs must involve computing derivatives. A "first principle" refers to a basic assumption that cannot be deduced from any other assumption, which is the foundation of theoretical derivation. Here, we state the local smooth principle in Definition 3 that ensures the basic accuracy of differentiation. This aligns well with our observation of many physical data, such as the locally smooth coefficient fields in Fig. 1 and the ground-truth coefficients and data in Fig. 2. On the contrary, the coefficient estimation and data fitting of SGTRidge are irregular as shown in Fig. 2, because the estimation of coefficients is separate at each spatial point, which does not consider the smoothness of coefficients across spatial dimensions. Naturally, we expect that a smooth nonlinear function on spatial dimensions can help model the nonlinear coefficients.

Definition 3 (Local smoothness)**.** *Given coefficient $\xi(x, y)$, the coefficients within a local area (with radius r) can be considered as a k-Lipschitz continuous function. Given the spatial distance of any two adjacent coordinates Dist $= \|S(x, y) - S(x', y')\| \leq r$ where $S(x, y)$ is the spatial coordinate vector, the slope of the coefficient function is bounded by $\alpha \geq 0$ as $\frac{|\xi(x,y)-\xi(x',y')|}{\|S(x,y)-S(x',y')\|} \leq \alpha$.*

Considering the principles of smoothness, we propose a local kernel estimation in the sparse regression that correlates the coefficient estimation at each spatial coordinate to the adjacent coefficient estimation. A spatially symmetrical kernel (i.e., spatial rotation invariance) for all coordinates (i.e., spatiotemporal translation invariance) would estimate coefficients with respect to conservation laws.

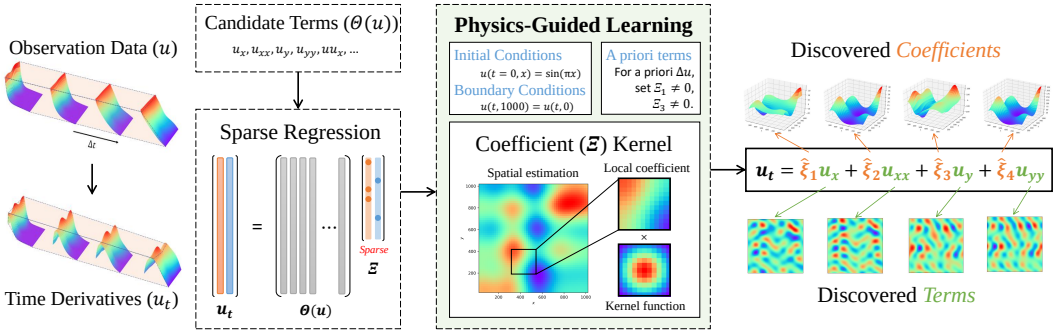

Figure 3: Diagram of the physics-guided discovery of PDEs with highly nonlinear coefficients.

A schematic diagram of the physics-guided learning framework is shown in Fig. 3. We expect it to enhance the model robustness for learning PDEs with highly nonlinear coefficients.

We prove that the proposed sparse regression with local kernel estimation can reduce the coefficient estimation error and reduce the error caused by noise when the coefficient fields comply with the local smoothness principle, with theorems and proofs in the Appendix B. Furthermore, as long as the spatial coordinates of the coefficients are provided, this local kernel estimation is mesh-free for spatiotemporal data, so nonlinear coefficients can be modeled even with irregularly sparse data.

For each (x, y), the proposed model considers all (x', y') that $\|S(x, y) - S(x', y')\| < r$ to compute

$$\widehat{\Xi} = \underset{\Xi}{\operatorname{argmin}} \|Y - X\Xi\|_2^2, \quad \widehat{\Xi}_i^{[x,y]} = \frac{\sum K_i^{[x',y']} \widehat{W}_i^{[x',y']}}{\sum K_i^{[x',y']}}. \tag{5}$$

$$K_i^{[x',y']} = \exp(-\frac{D^{[x',y']}}{2\gamma}), \quad D^{[x',y']} = \|S(x, y) - S(x', y')\|_2^2. \tag{6}$$

where $[x, y]$ denotes the spatial coordinate of the estimated coefficient while $[x', y']$ denotes each spatial coordinate of the adjacent coefficients. r denotes the radius of the local area. Here, $W \in \mathbb{R}^{n \times m}$ denotes the model parameters introduced in Eq.3, while $\Xi \in \mathbb{R}^{n \times m}$ is an intermediate tensor replacing W to represent the estimated coefficients. γ and q are both hyperparameters. We denote the spatial coordinate vector as $S(x, y)$ and denote the distance between two spatial coordinates (x, y) and (x', y') as $\|S(x, j) - S(x', y')\|$. The proposed coefficient estimation only takes the spatial distance as input information and is thus mesh-free to apply to continuous spaces for use in real practices.

We use the local kernel estimated Ξ of spatially adjacent coefficients instead of W as the regression weight to optimize the model $\widehat{Y} = X\Xi$. The learning of W at each spatial coordinate is dependent on adjacent X and Y, as W at each spatial coordinate participates in the calculation of all Ξ within the local area. Therefore, the proposed method calculates adjacent coefficients with nonlinearity when performing sparse regression, which leverages the physical principles to enrich data information and address overfitting. Here we can choose Radius Basis Function (RBF) kernel as K.

4 Experiments

4.1 Experimental Setting

Setup. Our experiments aim to discover PDEs terms and coefficients. The proposed method is compared with PDE-net [15], Sparse Regression (we compare with SGTRidge [5] here; SINDy [3, 17] is also an example of sparse regression) and A-DLGA [6]. We split the first 30% data in the time axis as the training set, the next 30% data as the development set, and the last 40% as the test set. We perform an additional experiment on the model robustness by adding Gaussian noise in Appendix E. For each model on each dataset, we tune the hyperparameters, i.e. γ, q and λ, via grid search, so that it has the lowest target u_t fitting error on the development (Dev, i.e. validation) set. The algorithm outline and the implementation details are presented in Appendix C.

Datasets. To test how well the proposed model performs on the discovery of PDEs-CC, we consider the Burgers' equation, the Korteweg-de Vries (KdV) equation, and the Chaffe-Infante equation. For PDEs-VC, we consider the convection diffusion equation and the governing equation of underground seepage where two cases are spatiotemporal 2D PDEs with simple variable coefficients (see Fig. 1(c)) and five cases are spatiotemporal 3D PDEs with different highly nonlinear coefficient fields (see Fig. 1(d)) that are hard to express explicitly. The data details are provided in Appendix D.

Evaluation Metrics. We use three metrics for evaluation:

1. Recall of the discovered PDE terms compared to ground truth;
2. The mean absolute error of coefficient ξ estimation;
3. The mean absolute error of target u_t fitting.

The recall rate of PDE terms and the coefficient estimation error indicate whether the discovered PDE is close to the ground-truth PDE that can generalize to future responses with the correct physical mechanism. Target fitting error tests how well the discovered PDE generalizes to the target u_t.

4.2 Results on PDEs with constant coefficients

We present the discovered PDEs with estimated PDE coefficients shown in the brackets and PDE terms discovered correctly. We use irregular samples to simulate sparsity in the continuous space.

Burgers' equation. We consider the discovery of the spatiotemporal 3D Burgers' equation with two physical fields u and v. For the sparse regression, we prepare a group of candidate functions that consist of polynomial terms $\{1, u, v, u^2, uv, v^2\}$, derivatives $\{1, u_x, u_y, v_x, v_y, \Delta u, \Delta v\}$ and their combinations. Following the physics-guided learning, we set diffusion terms as known a priori. The dimensionality of the dataset is $100 \times 100 \times 200$. We irregularly sample 10000 data and add 10% Gaussian noise. The discovered PDEs are shown below and the coefficients are averaged for each term. The recall, coefficient error, and target fitting error are shown in Table 1, which shows that our model performs well even with noisy, 3D, irregularly sampled data and multiple physical fields.

$$\begin{aligned} u_t &= 0.005(0.005015)u_{xx} + 0.005(0.004990)u_{yy} - 1(1.0152)uu_x - 1(1.0085)vu_y, \\ v_t &= 0.005(0.005018)v_{xx} + 0.005(0.004984)v_{yy} - 1(1.0097)uv_x - 1(1.0125)vv_y. \end{aligned} \tag{7}$$

Table 1: Model performance under different noisy levels for PDEs with constant coefficients.

Metrics	Recall (%)			Coefficient Error ($\times 10^{-3}$)			Fitting Error ($\times 10^{-3}$)		
Noise Level	0%	10%	20%	0%	10%	20%	0%	10%	20%
Burgers' Equation	100	100	100	2.603	6.124	6.946	0.205	0.356	1.004
KdV Equation	100	100	100	1.417	7.385	14.36	3.729	187.8	375.5
C-I Equation	100	100	100	3.623	12.69	25.38	1.691	11.85	23.71

Korteweg-de Vries (KdV) equation. We consider the discovery of spatiotemporal 2D Korteweg-de Vries (KdV) equation. We prepare a group of candidate functions that consist of polynomial terms $\{1, u, u^2\}$, derivatives $\{1, u_x, u_{xx}, u_{xxx}\}$ and their combinations. The dimensionality of the dataset is 512×201. We irregularly sample 5000 data and add 10% Gaussian noise. The discovered PDEs are shown in below and the coefficients are averaged for each term for the constant coefficient. The recall, coefficient error and target fitting error are shown in Table 1.

$$u_t = -1(1.0011)uu_x - 0.0025(0.002506)u_{xxx}. \tag{8}$$

Chaffe-Infante equation We consider the discovery of spatiotemporal 2D Chaffe-Infante equation. We prepare a group of candidate functions that consist of polynomial terms $\{1, u, u^2, u^3\}$, derivatives $\{1, u_x, u_{xx}\}$ and their combinations. The dimensionality of the dataset is 301×201. We irregularly sample 5000 data and add 10% Gaussian noise. The discovered PDEs are shown below and the coefficients are averaged for each term for the constant coefficient. The recall, coefficient error, and target fitting error are shown in Table 1. The coefficient of u_{xx} is less accurate as u_{xx} is very small.

$$u_t = 1(0.9212)u_{xx} - 1(0.9996)u + 1(1.0337)u^3. \tag{9}$$

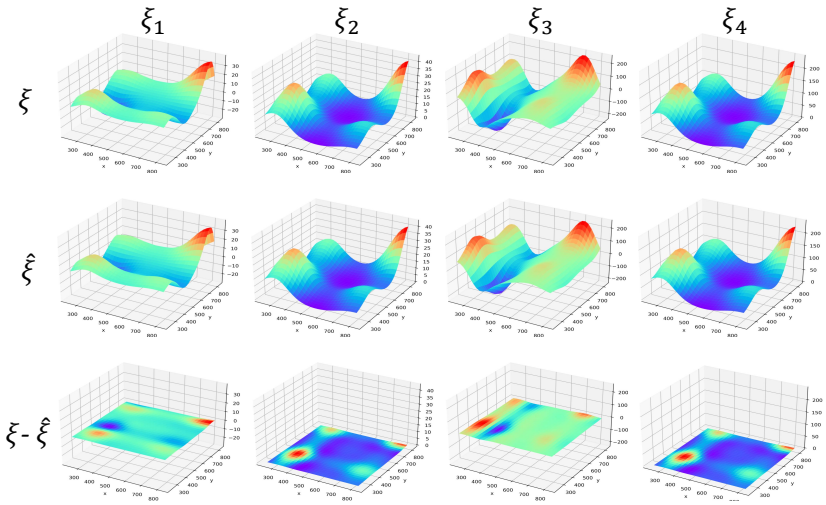

Figure 4: Comparison of estimated and correct nonzero coefficients of 1-HNC. Three rows represent the results of Ground Truth, our model and their residual error, respectively. Four columns represent four ξ_i that are nonzero in reality. Approximately, $|\xi^* - \hat{\xi}|/|\xi^*| = 1\%$.

4.3 Results on PDEs with variable coefficients

Convection diffusion equation. We consider the discovery of the spatiotemporal 2D convection diffusion equation with two different variable coefficient fields. We prepare a group of candidate functions that consist of polynomial terms $\{1, u, u^2\}$, derivatives $\{1, u_x, u_{xx}, u_{xxx}\}$ and their combinations. The dimensionalities of the two cases are all 100×251. We randomly sample 5000 data to simulate mesh-free sparsity and add 10% Gaussian noise. The discovered PDEs are shown below and the relative coefficient errors of the two cases are less than 0.3%. The recall of terms is 100% and the average u_t fitting error for the two cases are 1.8274×10^{-4}, and the coefficient estimation is visualized in Fig. A3 in Appendix F. Our model can discover the terms correctly and estimate coefficients accurately for the parametric convection diffusion equation from noisy and sparse data.

$$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx}; \quad |\xi - \hat{\xi}| < 3 \times 10^{-3}, |\xi - \hat{\xi}|/|\xi| < 0.3\%. \tag{10}$$

The governing equation of underground seepage. We consider the discovery of the spatiotemporal 3D governing equation of underground seepage with five different highly nonlinear variable coefficients, namely 1-HNC, 2-HNC, ..., 5-HNC. We prepare a group of candidate functions that consist of polynomial terms $\{1, u, u^2\}$, derivatives $\{1, u_x, u_{xx}, u_{xxx}\}$ and their combinations. The dimensionality of the fives cases are all $50 \times 50 \times 51$. We irregularly sample 10000 data and add 5% Gaussian noise. The discovered PDEs are shown below, and the relative coefficient errors are less than 1% for the five cases. The recall of terms and target u_t fitting errors are shown in Table 2. Our model can discover the terms correctly from noisy and irregularly sampled sparse data and can generalize to future data for all the five cases with highly nonlinear coefficients. The coefficient estimation of 1-HNC is visualized in Fig. 4 as an example, with the visualizations of more cases in Appendix F showing that the relative coefficient estimation error is less than 1%.

$$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_y + \hat{\xi}_3 u_{xx} + \hat{\xi}_4 u_{yy}; \quad |\xi - \hat{\xi}| < 1.8513, |\xi - \hat{\xi}|/|\xi| < 1\% \tag{11}$$

The discovered PDEs contain convection terms and diffusion terms along spatial dimensions, which align well with the ground-truth PDEs of underground seepage derived from the conservation of mass and Darcy's law [18]. On the contrary, all the other baselines render false PDE terms; the test target fitting errors of baselines are much larger than their training errors, reflecting overfitting. The test fitting errors of our model are much smaller than baselines, showing that our model effectively reduces the estimation error. To test the robustness of our model, we include results under noise from 5% to 20%. In most previous works for PDEs-CC [4, 17, 8] or PDEs-VC [5, 15, 10], the model

robustness to 5% or 10% noise is tested. Comparatively, the noise scale we test for our model is fairly large. We show that our model performs well for most cases under 10% noise. When the noise level goes up to 20%, in many cases one out of four PDE terms discovered would be wrong as

$$u_t = \hat{\xi}_1 \underline{uu_x} + \hat{\xi}_2 u_y + \hat{\xi}_3 u_{xx} + \hat{\xi}_4 u_{yy}, \quad \text{or} \quad u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_y + \hat{\xi}_3 u_{xx} + \hat{\xi}_4 \underline{uu_{yy}}. \tag{12}$$

We find that even under extremely large noise, our model can discover PDEs that can generalize well to future data on test sets, since uu_x and u_x are very similar when u is not rapidly changing along spatial dimensions, which is visualized in Fig. A2 in Appendix E. Moreover, we find that the model performs well in a wide range of hyperparameters, with details in Appendix E. Overall, our model shows excellent robustness against overfitting, especially with sparse and noisy data.

Table 2: Target fitting errors and recalls by different methods.

Datasets Noise	Metric	SGTRidge 5%	PDE-Net 5%	A-DLGA 5%	The Proposed Model 5%	 10%	 20%
1-HNC	Train Err ($\times 10^{-3}$)	1.148	2.190	16.70	3.314	6.283	11.88
	Dev Err ($\times 10^{-3}$)	3.907	10.85	47.39	3.919	6.373	12.10
	Test Err ($\times 10^{-3}$)	28.25	31.80	136.8	**3.686**	6.367	12.29
	Recall (%)	50	50	25	**100**	100	75
2-HNC	Train Err ($\times 10^{-3}$)	2.283	2.697	19.83	3.794	5.676	8.329
	Dev Err ($\times 10^{-3}$)	26.87	42.23	43.04	3.794	5.674	8.332
	Test Err ($\times 10^{-3}$)	106.1	169.2	123.8	**3.585**	5.499	8.318
	Recall (%)	25	25	25	**100**	100	75
3-HNC	Train Err ($\times 10^{-3}$)	0.134	0.129	1.033	0.331	0.583	0.969
	Dev Err ($\times 10^{-3}$)	1.588	1.563	2.661	0.342	0.588	0.997
	Test Err ($\times 10^{-3}$)	9.095	9.005	7.872	**0.343**	0.589	1.018
	Recall (%)	25	25	50	**100**	100	75
4-HNC	Train Err ($\times 10^{-3}$)	0.336	0.301	21.50	1.733	2.235	3.662
	Dev Err ($\times 10^{-3}$)	11.52	10.13	37.61	1.729	2.302	4.037
	Test Err ($\times 10^{-3}$)	94.47	85.60	150.0	**1.703**	2.521	7.505
	Recall (%)	0	0	0	**100**	75	25
5-HNC	Train Err ($\times 10^{-3}$)	1.218	1.506	20.66	1.940	3.139	4.438
	Dev Err ($\times 10^{-3}$)	6.624	7.508	42.72	1.984	2.984	4.254
	Test Err ($\times 10^{-3}$)	13.63	15.17	109.6	**1.733**	3.049	4.345
	Recall (%)	50	50	50	**100**	100	100

5 Conclusion and Future Work

How to discover Partial Differential Equations (PDEs) with highly nonlinear variable coefficients from sparse and noisy data is an important task. To address the overfitting of coefficients caused by data quality issues in previous baselines, we propose a physics-guided spatial kernel estimation in sparse regression that aligns well with the local smooth principle in PDEs and conservation laws. The proposed model incorporates physical principles into a nonlinear smooth kernel to model the highly nonlinear coefficients. We theoretically prove that it strictly reduces the coefficient estimation error of previous baselines and is also more robust against noise. With spatial coordinates of coefficients, the model can apply to mesh-free spatiotemporal data without grids. In experiments, it demonstrates the ability to find various PDEs from sparse and noisy data. More importantly, it for the first time reports the discovery of PDEs with highly nonlinear coefficients, while previous baselines yield false results. Our model performs well with a wide range of hyperparameters and noise level up to 20%. With the state-of-the-art performance, our method brings hope to discover complex PDEs that comply with the continuously differentiable and local smoothness principles to help scientists understand unknown complex phenomena. In the future, how to avoid the intervention of correlated similar terms and improve the accuracy of differentiation remain important. Our method works for PDEs that comply with the principles, but may remain intractable for more rarely complex coefficient fields. Also, how to discover equations without the prior knowledge of time-dependent target term is not discussed yet.

References

[1] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[2] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

[3] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the national academy of sciences*, 113(15):3932–3937, 2016.

[4] Samuel H Rudy, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, 2017.

[5] Samuel Rudy, Alessandro Alla, Steven L Brunton, and J Nathan Kutz. Data-driven identification of parametric partial differential equations. *SIAM Journal on Applied Dynamical Systems*, 18(2):643–660, 2019.

[6] Hao Xu, Dongxiao Zhang, and Junsheng Zeng. Deep-learning of parametric partial differential equations from sparse and noisy data. *Physics of Fluids*, 33(3):037132, 2021.

[7] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

[8] Chengping Rao, Pu Ren, Yang Liu, and Hao Sun. Discovering nonlinear pdes from scarce data with physics-encoded learning. In *International Conference on Learning Representations*, 2022.

[9] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature communications*, 12(1):1–13, 2021.

[10] Jun Li, Gan Sun, Guoshuai Zhao, and H Lehman Li-wei. Robust low-rank discovery of data-driven partial differential equations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 767–774, 2020.

[11] Dongxiao Zhang and Zhiming Lu. An efficient, high-order perturbation approach for flow in random porous media via karhunen–loeve and polynomial expansions. *Journal of Computational Physics*, 194(2):773–794, 2004.

[12] SP Huang, ST Quek, and KK Phoon. Convergence study of the truncated karhunen–loeve expansion for simulation of stochastic processes. *International journal for numerical methods in engineering*, 52(9):1029–1043, 2001.

[13] Hao Xu, Haibin Chang, and Dongxiao Zhang. Dlga-pde: Discovery of pdes with incomplete candidate library via combination of deep learning and genetic algorithm. *Journal of Computational Physics*, 418:109584, 2020.

[14] Dongxiao Zhang. *Stochastic methods for flow in porous media: coping with uncertainties*. Elsevier, 2001.

[15] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, pages 3208–3216. PMLR, 2018.

[16] Zichao Long, Yiping Lu, and Bin Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *Journal of Computational Physics*, 399:108925, 2019.

[17] Kathleen Champion, Bethany Lusch, J Nathan Kutz, and Steven L Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[18] Hilton H Cooper Jr. The equation of groundwater flow in fixed and deforming coordinates. *Journal of Geophysical Research*, 71(20):4785–4790, 1966.

[19] Jeremy Morton, Freddie D Witherden, Antony Jameson, and Mykel J Kochenderfer. Deep dynamical modeling and control of unsteady fluid flows. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 9278–9288, 2018.

[20] Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional koopman operators for model-based control. In *International Conference on Learning Representations*, 2019.

[21] Hyuk Lee and In Seok Kang. Neural algorithm for solving differential equations. *Journal of Computational Physics*, 91:110–131, 1990.

[22] Isaac E. Lagaris, Aristidis C. Likas, and Dimitrios Ioannis Fotiadis. Artificial neural networks for solving ordinary and partial differential equations. *IEEE transactions on neural networks*, 9 5:987–1000, 1998.

[23] Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 481–490, 2016.

[24] J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.

[25] Brandon Amos and J Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*, pages 136–145. PMLR, 2017.

[26] Justin Sirignano and Konstantinos Spiliopoulos. Dgm: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375:1339–1364, 2018.

[27] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115:8505 – 8510, 2018.

[28] Chengping Rao, Hao Sun, and Yang Liu. Physics-informed deep learning for computational elastodynamics without labeled data. *Journal of Engineering Mechanics*, 147(8):04021043, 2021.

[29] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[30] Yuntian Chen, Dou Huang, Dongxiao Zhang, Junsheng Zeng, Nanzhe Wang, Haoran Zhang, and Jinyue Yan. Theory-guided hard constraint projection (hcp): A knowledge-based data-driven scientific machine learning method. *Journal of Computational Physics*, 445:110624, 2021.

[31] Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, Anima Anandkumar, et al. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2020.

[32] Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

[33] Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018.

[34] Yinhao Zhu, Nicholas Zabaras, Phaedon-Stelios Koutsourelakis, and Paris Perdikaris. Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. *Journal of Computational Physics*, 394:56–81, 2019.

[35] Yohai Bar-Sinai, Stephan Hoyer, Jason Hickey, and Michael P Brenner. Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, 116(31):15344–15349, 2019.

[36] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of pde systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403:109056, 2020.

[37] Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21), 2021.

[38] Han Gao, Luning Sun, and Jian-Xun Wang. Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. *Journal of Computational Physics*, 428:110079, 2021.

[39] Filipe De Avila Belbute-Peres, Thomas Economon, and Zico Kolter. Combining differentiable pde solvers and graph neural networks for fluid flow prediction. In *International Conference on Machine Learning*, pages 2402–2411. PMLR, 2020.

[40] Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pages 8459–8468. PMLR, 2020.

[41] Han Gao, Matthew J Zahr, and Jian-Xun Wang. Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502, 2022.

[42] Yunzhu Li, Jiajun Wu, Russ Tedrake, Joshua B Tenenbaum, and Antonio Torralba. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. In *International Conference on Learning Representations*, 2018.

[43] Benjamin Ummenhofer, Lukas Prantl, Nils Thuerey, and Vladlen Koltun. Lagrangian fluid simulation with continuous convolutions. In *International Conference on Learning Representations*, 2019.

[44] Guillaume Lample and François Charton. Deep learning for symbolic mathematics. In *International Conference on Learning Representations*, 2019.

[45] Martin Magill, Faisal Qureshi, and Hendrick W de Haan. Neural networks trained to solve differential equations learn general representations. In *32nd Conference on Neural Information Processing Systems*, 2018.

[46] Kiwon Um, Robert Brand, Philipp Holl, Nils Thuerey, et al. Solver-in-the-loop: Learning from differentiable physics to interact with iterative pde-solvers. In *34nd Conference on Neural Information Processing Systems*, 2020.

[47] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. In *34nd Conference on Neural Information Processing Systems*, 2020.

[48] Saso Dzeroski and Ljupco Todorovski. Discovering dynamics: from inductive logic programming to machine discovery. *Journal of Intelligent Information Systems*, 4(1):89–108, 1995.

[49] Miles Cranmer, Alvaro Sanchez Gonzalez, Peter Battaglia, Rui Xu, Kyle Cranmer, David Spergel, and Shirley Ho. Discovering symbolic models from deep learning with inductive biases. *Advances in Neural Information Processing Systems*, 33:17429–17442, 2020.

[50] Subham Sahoo, Christoph Lampert, and Georg Martius. Learning equations for extrapolation and control. In *International Conference on Machine Learning*, pages 4442–4450. PMLR, 2018.

[51] Samuel Kim, Peter Y Lu, Srijon Mukherjee, Michael Gilbert, Li Jing, Vladimir Čeperić, and Marin Soljačić. Integration of neural network-based symbolic regression in deep learning for scientific discovery. *IEEE Transactions on Neural Networks and Learning Systems*, 32(9):4166–4177, 2020.

[52] Yuntian Chen and Dongxiao Zhang. Integration of knowledge and data in machine learning. *arXiv preprint arXiv:2202.10337*, 2022.

[53] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.

[54] Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357:125–141, 2018.

[55] Valerii Iakovlev, Markus Heinonen, and Harri Lähdesmäki. Learning continuous-time {pde}s from sparse data with graph neural networks. In *International Conference on Learning Representations*, 2021.

[56] Chi Chiu So, Tsz On Li, Chufang Wu, and Siu Pang Yung. Differential spectral normalization (dsn) for pde discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[57] Nanzhe Wang, Dongxiao Zhang, Haibin Chang, and Heng Li. Deep learning of subsurface flow via theory-guided neural network. *Journal of Hydrology*, 584:124700, 2020.

Appendix

A Related Work

Dynamical system modeling Machine learning is widely leveraged to predict the future response of desired physical fields from data [19, 20]. As an alternative, scientists can also obtain the future response by solving a partial differential equation (PDE) that describes the dynamical system. Early pioneering works [21, 22] using neural networks to simulate dynamical systems can date back to three decades ago. More recent machine learning algorithms [23–26] can be mainly divided into two branches: the mesh-based discrete learning and the meshfree continuous learning of simulation. Within the meshfree learning branches, pure data-driven approaches [27, 15] are mainly based on high-quality data and physics-informed approaches [7, 28–30] use physics knowledge to enhance models to adapt to noisier and sparser data. Recent studies on neural operators [31, 32] also use neural networks to learn the meshfree and infinite-dimensional mapping for dynamical systems. Within the mesh-based learning branches, convolutional networks are widely adopted [33, 34] to simulate PDEs for spatiotemporal systems [35–38]. The geometry-adaptive learning of nonlinear PDEs with arbitrary domains [39–41] and the particle-based dynamical system modeling [42, 43] by graph neural networks rises as a promising direction. Moreover, deep learning also renders giving symbolic representation of solutions to PDEs[44] possible and demonstrate higher accuracy [45–47].

Data-driven discovery Early trials for equation discovery in the last century [48] uses inductive logic programming to find the natural laws. Two research streams have been proposed to search the governing equations. The first stream aims at identifying a symbolic model [49] that describes the dynamical systems from data, which uses symbolic regression [1, 2] and symbolic neural networks [50, 51] to discover functions by comparing differentiation of the experimental data with analytic derivatives of candidate function. The second stream is mainly to incorporate prior knowledge [52, 9, 8] and perform sparse regressions [4, 53, 3, 54, 35] to discover PDEs by selecting term candidates. Evolutionary algorithm [13] is also proposed to start with an incomplete library and evolve through generations.

While these algorithms only discover the equation structure for PDEs with constant coefficients, later works also start to work on the discovery of PDEs with variable coefficients. For PDEs with variable coefficients, we need to determine their PDE structures (the partial derivative terms that form the PDE) and coefficients (the variable coefficients that multiply partial derivative terms in the PDE) at the same time. Sequential Group Threshold Regression [5] combines coefficient regression and term selection to find PDEs with variable coefficients. PDE-Net [15, 16], Graph-PDE [55] and Differential Spectral Normalization [56] are proposed to use neural blocks such as convolution to discover the PDEs models. In addition, DLrSR [10] solves the noise problem by separating the clean low-rank data and outliers. A-DLGA [6] proposes to alleviate data linear dependency at the sacrifice of estimation error. Up until now, the current state-of-the-art approaches have proven to discover some PDEs with variable coefficients, but the discovery of PDEs with highly nonlinear coefficients remains a challenge [5, 6, 16] due to the overfitting of the sparse regressions and data quality issues.

B Theoretical Analysis

In this section, we introduce the theoretical analysis to demonstrate the advantages of our model. We provide several theorems in the following with proofs. The proposed spatial kernel estimation (See Eqs.5 and 6) uses the spatial distance to estimate the probability density function of coefficients with nonlinearity. To help understand its advantage, we first introduce a spatial averaging estimation with linearity here. We intend to show that the spatial averaging estimation can have an estimation error upper bounded by the upper limit of coefficient difference between adjacent coordinates. The coefficient estimation error without such spatial averaging estimation, on the contrary, has no upper bound (i.e., can be fairly large). Furthermore, we will show that our proposed spatial kernel estimation has a lower estimation error than the spatial averaging estimation. We also prove that the estimation error caused by noise can be greatly alleviated by the spatial kernel. The analysis not only proves that our proposed spatial estimation strategy can reduce estimation errors but also demonstrates that the kernel estimation with nonlinearity is very suitable for the tasks.

The local averaging estimation is defined as follows. For each (x, y), the spatial averaging estimation considers all (x', y') that $\|S(x, y) - S(x', y')\| < r$ to compute

$$\widehat{\Xi}_{(avg)} = \underset{\Xi}{\operatorname{argmin}} \|Y - X\Xi\|_2^2, \quad \widehat{\Xi}_{(avg)}^{[x,y]} = \frac{\sum_{(x',y')} \widehat{W}^{[x',y']}}{\sum_{(x',y')} 1}. \tag{13}$$

where W has the same definition as used in the spatial kernel estimation, i.e. model parameters, as described by Eqs. 3 and 4. We use $\widehat{\Xi}_{(avg)}$ to denote the estimated coefficients and ξ to denote the ground-truth coefficients. To express its upper bound error, we introduce the Lipschitz continuity to express the local smoothness with a Lipschitz constant $\alpha \geq 0$, as introduced in Definition 3. Here, we consider the upper limit of coefficient difference between adjacent coefficients within the local area for all x, y, x', and y' as

$$\alpha \geq \frac{|\widehat{W}^{[x',y']} - \widehat{W}^{[x,y]}|}{\|S(x, y) - S(x', y')\|} \geq \frac{|\widehat{W}^{[x',y']} - \widehat{W}^{[x,y]}|}{r}. \tag{14}$$

According to the local smoothness principle stated in Definition 3, the coefficient variation along the spatial dimensions is an increasing function. In the worse case, we have all the coefficients on only one side with differences approximating αr. Therefore, the upper bound of estimation error is

$$sup(|\widehat{\Xi}_{(avg)}^{[x,y]} - \xi^{[x,y]}|) = \widehat{W}^{[x,y]} - \frac{\sum_{(x',y')} \widehat{W}^{[x,y]} - \alpha r}{\sum_{(x',y')} 1} = \alpha r. \tag{15}$$

While the spatial averaging coefficient estimation has a upper bound of coefficient estimation error, the spatially independent estimation in Eqs. 3 and 4 practiced by many baselines cannot guarantee to match the ground-truth coefficients even if Eq. 4 is optimized due to the existence of many linearly dependent observations. We assume that the spatial averaging estimation can avoid this issue by using extra data from adjacent coordinates within the local area in the sacrifice of introducing the estimation error as described in Eq. 15. We can also easily demonstrate that the local averaging estimation has a lower estimation error than the strategy practiced in A-DLGA [6] that makes coefficients grids coarser by merging grids within each spatial area into one grid, which also uses extra adjacent data to alleviate the issue caused by linearly independent observations. We formalize this in Theorem 1.

Theorem 1 (Reduction on coefficient error by local averaging estimation)**.** *With respect to the local smoothness principle, the coefficients estimated by the spatial averaging estimation has strictly lower upper-bound coefficient estimation error than A-DLGA.*

Proof. Assume that the *Local smooth Principle* in Definition 3 applies, we consider the estimation of coefficient $\xi(x, y)$. We denote the estimated coefficients of A-DLGA as $\widehat{W}^{\dagger}$. For the $\widehat{W}^{\dagger}$, the upper bound of estimation error should be the case where (x, y) locates at the edge of a local area, instead of at the center as in the spatial averaging estimation, so that the upper limit of coefficient difference between coordinates within the area would be $2\alpha r$. For each (x, y), A-DLGA considers all (x', y') that $\|S(x'', y'') - S(x', y')\| < r$, where $\|S(x'', y'') - S(x, y)\| < r$. Therefore,

$$sup(|\widehat{W}^{\dagger} - \xi|) = \widehat{W}^{[x,y]} - \frac{\sum_{(x',y')} \widehat{W}^{[x,y]} - 2\alpha r}{\sum_{(x',y')} 1} = 2\alpha r > \alpha r = sup(|\widehat{\Xi}_{(avg)} - \xi|). \tag{16}$$

■

We further demonstrate in Theorem 2 that our proposed spatial kernel estimation with nonlinearity reduces the coefficient estimation error of the local averaging estimation with linearity. Our model is more accurate than local averaging estimation, so it is more accurate than A-DLGA.

Theorem 2 (Reduction on coefficient error by local kernel)**.** *With respect to the local smooth principle, the coefficients estimated by the spatial kernel estimation has strictly lower coefficient estimation error than the spatial averaging estimation.*

Proof. Assume that the *Local smooth Principle* in Definition 3 applies, we consider the estimation of coefficient $\xi(x, y)$. Because the coefficient function is a k-Lipschitz continuous function, the

coefficient difference increases with spatial distance: the closer the coordinates, the smaller the coefficient difference. The spatial kernel estimation is stated in Eqs. 5 and 6. We denote the estimated coefficients as Ξ. Note that the kernel value defined as $K^{[x',y']} = \exp(-\frac{\|S(x,y)-S(x',y')\|_2^2}{2\gamma})$ in Eq. 6 decreases in the spatial distance, so closer coefficients give more contributions. Therefore, we have

$$|\widehat{\Xi}^{[x,y]} - \xi^{[x,y]}| = \frac{\sum K^{[x',y']}(\widehat{W}^{[x',y']} - \widehat{W}^{[x,y]})}{\sum K^{[x',y']}} \tag{17}$$

$$\propto \frac{\sum K^{[x',y']}\delta(\|S(x,y) - S(x',y')\|)}{\sum K^{[x',y']}} \tag{18}$$

$$\leq \frac{\sum \delta(\|S(x,y) - S(x',y')\|)}{\sum 1} \tag{19}$$

$$= |\widehat{\Xi}^{[x,y]}_{(avg)} - \xi^{[x,y]}|. \tag{20}$$

The above equations and inequalities prove that the coefficient estimation error of spatial kernel estimation is strictly lower than the coefficient estimation error of spatial averaging estimation. ■

Moreover, the spatial kernel estimation reduces the coefficient estimation error caused by noise in the sparse regression, which is proved in Theorem 3. This makes the PDE discovery more robust. For coefficient at a spatial coordinate (x, y), its estimation is affected by both the noise in its Y and the noises in Y at adjacent coordinates within the local area. We only discuss the estimation error caused by noise here, so we assume $\widehat{\Xi} = \widehat{W} = \xi$ if $\eta = 0, \epsilon = 0$, which means that the estimation of coefficients must be with a balance distribution of adjacent coefficients. This allows us to discuss the noise effect alone without the error caused by kernel discussed in Theorem 2. We prove that the weighted addition of independent Gaussian noises by kernel estimation has a lower error than the original sparse regression and the error is lower when more adjacent coefficients are considered.

Theorem 3 (Reduction on coefficient error caused by noise)**.** *Assume that the coefficient estimation error is only caused by noise so that $\widehat{\Xi} = \widehat{W} = \xi$ if $\eta = 0, \epsilon = 0$, then we must have $|\widehat{\Xi} - \xi| < |\widehat{W} - \xi|$ if $\eta \neq 0, \epsilon \sim \eta\mathcal{N}(0, \sigma^2) \in \mathbb{R}^h$.*

Proof. Consider $\epsilon \sim \eta\mathcal{N}_h(0, \sigma^2)$ in estimation that $\widehat{W} = (X^TX + \lambda I)X^T(Y + \epsilon)$. $|\widehat{W} - \xi| = (X^TX + \lambda I)X^T\epsilon$. We assume $\widehat{\Xi} = \widehat{W} = \xi$ if $\eta = 0, \epsilon = 0$, which means for each $\widehat{\Xi}^{[x',y']}$ there must always be another $\widehat{\Xi}^{[x'',y'']}$ such that 1) they have the same distance to $\widehat{\Xi}^{[x,y]}$ so they have the same kernel value, i.e. $\|S(x', y') - S(x, y)\| = \|S(x'', y'') - S(x, y)\|$, and 2) they are symmetrical to the value of $\xi^{[x,y]}$ so that their biases can be offset, i.e. $\xi^{[x,y]} - \xi_i^{[x',y']} = \xi^{[x'',y'']} - \xi^{[x,y]}$. Based on this, for all (x', y') that $\|S(x, y) - S(x', y')\| < r$, the estimated coefficients should be

$$|\widehat{\Xi}^{[x,y]} - \xi^{[x,y]}| = \frac{\sum_{(x',y')} K^{[x',y']}\epsilon_{(x',y')}}{\sum_{(x',y')} K^{[x',y']}}.$$

For each $\epsilon_{(x',y')} \sim \eta N(0, \sigma^2)$ that is i.i.d., as $\sum \epsilon \sim \eta N(0, \sum \sigma_i^2)$, we have $|\widehat{\Xi}^{[x,y]} - \xi^{[x,y]}| \sim \eta N(0, \sum_{(x',y')} \frac{K^{[x',y']}\sigma^2}{K^{[x',y']}})$. As each σ is the same for all adjacent coefficients within the area, we have $|\widehat{\Xi}^{[x,y]} - \xi^{[x,y]}| \sim \frac{\eta}{\sum_{(x',y')} 1} N(0, \sigma^2)$. However, $|\widehat{W}^{[x,y]} - \xi^{[x,y]}| \sim \eta N(0, \sigma^2)$. Therefore, $|\widehat{\Xi} - \xi| < |\widehat{W} - \xi|$ and $\frac{|\widehat{W} - \xi|}{|\widehat{\Xi} - \xi|} = \sum_{(x',y')} 1$. ■

C Algorithm Details

C.1 Iterative One-Out Sparse Regression

We use an iterative one-out regression that filters out one $X_{:,i}$ which gives the least Akaike information criterion (AIC) at each iteration of coefficient estimation. If we use M to denote the set of indexes of reserved coefficients, the formula of AIC is approximately used as follows

$$\text{AIC}(M) = 2\sum_{i \in M} 1 - 2ln\left(\left\|Y - \sum_{i \in M} X_{:,i}\Xi_i\right\|_2^2\right). \tag{21}$$

The iteration ends when there are only L coefficients left in the regression. This aims to filter out the most irrelevant ξ_i that maximize the least square errors to avoid its intervention in estimating coefficients. The iterative one-out regression repeats

$$M = M - [i] \text{ if AIC}(M - [i]) = \min(\text{AIC}(M - [j])) \text{ for } \forall\, j \in M, \tag{22}$$

$$\widehat{\Xi} = \underset{\Xi}{\operatorname{argmin}} \left\| Y - \sum_{i \in M} X_{:,i} \Xi_i \right\|_2^2. \tag{23}$$

Iterative one-out regression is an approximation of sparse group regression that improves the accuracy in determining the nonzero ξ_i without the interference of irrelevant terms, as a similar method has pointed out [5]. The overall algorithm is expressed in Algorithm 1.

Algorithm 1 The physics-guided spatial kernel sparse regression approach.

Require: Target time derivative term $u_t(x, y, t)$ and candidate equation terms $\Theta(u)_i(x, y, t)$ w.r.t. $x \in [1, ..., n]$, $y \in [1, ..., m]$ and $t \in [1, ..., h]$. p that $M = [1, 2, ..., p]$, $i \in M$, λ, γ, q, L.

For convenience, denote u_t as Y and denote $\Theta(u)_i$ as $X_{:,i}$.
while $size(M) > L$ **do**
  Compute $\widehat{\Xi}$ by Eqs.(3-6) using all $X_{:,i}$ and Y with $i \in M$;
  $M = M - [i]$ if AIC$(M - [i]) = \min(\text{AIC}(M - [j]))$ for $\forall\, j \in M$;
end while
Compute $\widehat{\Xi}_{best}$ by Eqs.(3-6) using all $X_{:,i}$ and Y with $i \in M$.
return M, $\widehat{\Xi}_{best}$.

D Data statistics

We introduce the governing equation of underground seepage in the following. The subsurface flows in the field of fluid mechanics with different coefficients are taken as (1-5)-HNCs to perform the experiments. The governing equation for the data is:

$$S_s \frac{\partial u}{\partial t} = \frac{\partial}{\partial x}(K(x, y)\frac{\partial u}{\partial x}) + \frac{\partial}{\partial y}(K(x, y)\frac{\partial u}{\partial y}).$$

where S_s denotes the specific storage; $K(x, y)$ denotes the hydraulic conductivity field; and u denotes the hydraulic head. In our work, u is the physical field the desired PDE describes and K is the coefficient field. This equation is also used by PDE-net [15] but its coefficient field is much simpler. The two variable coefficient fields used in PDE-net are $0.5(\cos(y) + x(2\pi - x)\sin(x)) + 0.6$ and $b(x, y) = 2(\cos(y) + \sin(x)) + 0.8$. The hydraulic conductivity field $K(x, y)$ in the governing equation is set to be heterogeneous to better simulate real situations in practice. The heterogeneous fields are often regarded as random fields with higher complexity following a specific distribution with corresponding covariance [11, 12, 14, 57].

In detail, in our paper a two-dimensional transient saturated flow in porous medium is considered. The domain is a square, which is evenly divided into 51×51 grid blocks and the length in both directions is 1020 [L], where [L] denotes any consistent length unit. The left and right boundaries are set as constant pressure boundaries and the hydraulic head takes values of $H_{x=0} = 202$ [L] and Hx=1020=200 [L], respectively. Furthermore, the two lateral boundaries are assigned as no-flow boundaries. The specific storage is assumed as a constant, taking a value of SS=0.0001 [L-1]. The total simulation time is 10 [T], where [T] denotes any consistent time unit, with each time step being 0.2 [T], resulting in 50 time steps. The initial conditions are $H_{t=0,x=0} = 202$ [L] and $H_{t=0,x\neq 0} = 200$ [L]. The mean and variance of the log hydraulic conductivity are given as 0 and 1, respectively. In addition, the correlation length of the field is $408[L]$. The hydraulic conductivity field is parameterized through KLE and 20 terms are retained in the expansion. Therefore, this field is represented by 20 random variables $\xi = \xi_1'(\tau), \xi_2'(\tau), ..., \xi_{20}'(\tau)$ in the considered cases. An example of conductivity field obtained through KLE is shown in Fig. A1(a), which exhibits strong anisotropy. MODFLOW software is adopted to perform the simulations to obtain the dataset, and the data distributions at two time steps are presented in Fig. A1(b) and (c) as an example.

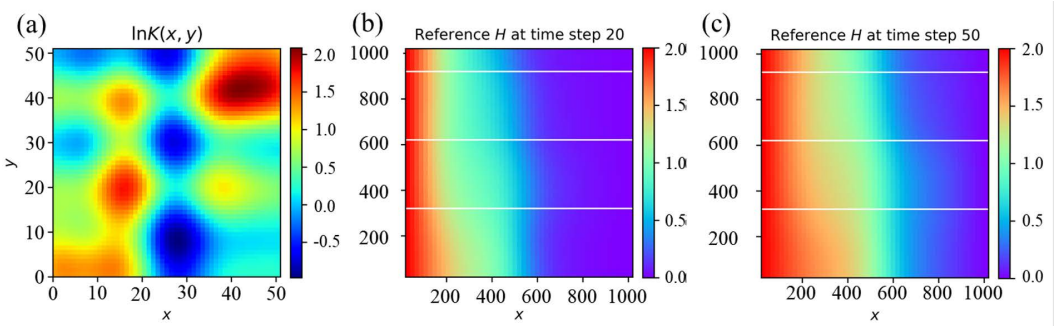

Figure A1: Color maps of conductivity field hydraulic pressure field.

For the two convection diffusion equation cases, we simulate the two-dimensional PDEs $u_t = -\frac{1}{x}u_x + 0.1u_{xx}$ and $u_t = -\frac{1}{(x+\sin \pi x)}u_x + 0.1u_{xx}$ with variable coefficients $-\frac{1}{x}$ and $-\frac{1}{(x+\sin \pi x)}$ that can be expressed explicitly and do not have physical meanings or references in the nature, respectively. These two equations are also generated by MODFLOW. All the data used are available in the *Supplementary Materials*. We use a single GPU machine of GTX1080 and the error variation to random seeds is statistically insignificant.

E Hyperparameter and Robustness Analysis

In addition to the main experiment and the robustness experiment, we also conduct a hyperparameter analysis to discuss the suitable range of hyperparameters to ensure model performance. We set the radius r within [2, 5, 10] and set the γ value of the Gaussian kernel within [0.03, 0.1, 0.3, 1] to find out whether our model is stable over a wide hyperparameter range. The γ value determines the variance of the kernel. When $\gamma \to 0$, the kernel estimation is equivalent to the averaging estimation introduced in Appendix B. When $\gamma \to \infty$, the kernel estimation degrades to separate regression at each spatial coordinate. We find that as the value of γ decreases, the coefficient error increases. This reflects the gradual approximation to the local averaging estimation when the values of the kernel function at each coordinate are almost the same. The increasing error aligns well with the Theorems 2 and 3 in Appendix B that local averaging estimation has larger estimation error.

Although a wide range of hyperparameters can all give the correct PDE structure, we can tune the hyperparameters for each specific case to obtain the best performance. Table A5 shows that $r = 5, \gamma = 1$ is the best hyperparameter setting for 1-HNC in the experiment. The radius r here is the spatial distance that is normalized for each dataset. These kernel functions that have tiny coefficient estimation error mainly rely on the most adjacent coefficients for estimation. The optimal values of both r and γ are determined by the graininess of the local smooth principle in real practice. If the radius is too large, the kernel will no longer be "local" to match the principle. In all, our model performs well within a wide range of hyperparameters. The hyperparameters we use in the main experiments are $r = 10, \gamma = 1$. We show the full results of hyperparameter analysis of (1-5)-HNCs in the following.

Table A1: PDE structures and coefficients discovered w.r.t. hyperparameters for 1-HNC.

Hyperparameters	PDE structure	Recall	Coefficient error
$r = 10, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.7090
$r = 10, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.2296
$r = 10, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	5.2279
$r = 10, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	8.2493
$r = 5, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.5962
$r = 5, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	1.4790
$r = 5, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.0122
$r = 5, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.2271
$r = 2, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	1.7029
$r = 2, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.6926
$r = 2, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	3.0050
$r = 2, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	3.1153

Table A2: PDE structures and coefficients discovered w.r.t. hyperparameters for 2-HNC.

Hyperparameters	PDE structure	Recall	Coefficient error
$r = 10, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_4 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1183
$r = 10, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.3749
$r = 10, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.8757
$r = 10, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	1.3561
$r = 5, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.5962
$r = 5, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	1.4790
$r = 5, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.0122
$r = 5, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.2271
$r = 2, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	1.7029
$r = 2, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	2.6926
$r = 2, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	3.0050
$r = 2, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	3.1153

Table A3: PDE structures and coefficients discovered w.r.t. hyperparameters for 3-HNC.

Hyperparameters	PDE structure	Recall	Coefficient error
$r = 10, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_4 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0272
$r = 10, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0874
$r = 10, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.2093
$r = 10, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	1.3561
$r = 5, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0206
$r = 5, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0514
$r = 5, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0701
$r = 5, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0776
$r = 2, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0933
$r = 2, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1478
$r = 2, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1650
$r = 2, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1711

Table A4: PDE structures and coefficients discovered w.r.t. hyperparameters for 4-HNC.

Hyperparameters	PDE structure	Recall	Coefficient error
$r = 10, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_4 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0416
$r = 10, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1317
$r = 10, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.3103
$r = 10, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.4879
$r = 5, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0330
$r = 5, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0823
$r = 5, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1121
$r = 5, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1242
$r = 2, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1046
$r = 2, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1656
$r = 2, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1848
$r = 2, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1916

Table A5: PDE structures and coefficients discovered w.r.t. hyperparameters for 5-HNC.

Hyperparameters	PDE structure	Recall	Coefficient error
$r = 10, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_4 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0270
$r = 10, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0871
$r = 10, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.2169
$r = 10, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.3618
$r = 5, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0304
$r = 5, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.0756
$r = 5, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1030
$r = 5, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1140
$r = 2, \gamma = 1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1156
$r = 2, \gamma = 0.3$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.1828
$r = 2, \gamma = 0.1$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.2039
$r = 2, \gamma = 0.03$	$u_t = \hat{\xi}_1 u_x + \hat{\xi}_2 u_{xx} + \hat{\xi}_3 u_y + \hat{\xi}_4\, u_{yy}$	100%	0.2114

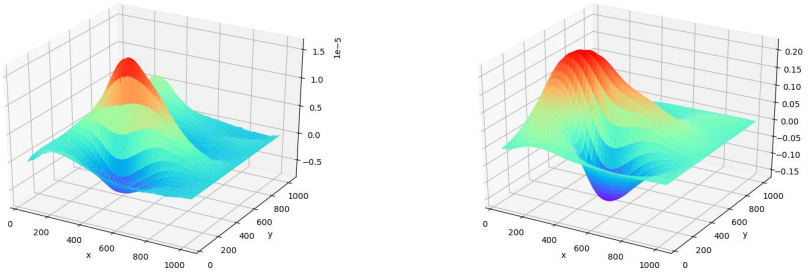

Figure A2: The similar distributions of partial derivative term u_x and uu_x, respectively. Correlated terms like these can be challenges for the filtering process of the iterative sparse regression.

The main experiments are mostly conducted with up to 20% noise. Here, we conduct a robustness analysis to discuss whether the proposed model can handle even larger noise of 25% and 30%. We find that for some PDE cases with constant coefficients, our model can perform well even under 30% noise. We report the discovered PDE structure and coefficient error of the 1-HNC dataset in Table A6. In the table, C-I equation denotes the Chaffe-Infante equation and C-D equation denotes the first

case of convection diffusion equation. From the table, we find that our model performs well under 30% noise for Burgers' equation and KdV equation, which outperforms the robustness reported in previous baselines. Compared to results in Table. 1, we find that the coefficient errors and fitting error of Burgers' equation and KdV equation are close to the results under 20%. However, for C-I equation and convection diffusion equation, the coefficient errors and fitting errors increases drastically since the discovered PDE terms are partially wrong. The wrongly discovered PDE terms for C-I equation is $\{u, u^2, u_{xx}\}$ instead of $\{u, u^3, u_{xx}\}$. The wrongly discovered PDE terms for the first case of C-D equation is $\{u, u_x\}$ instead of $\{u_x, u_{xx}\}$. We also show in Fig. A2 that some terms such as u_x and uu_x are highly correlated and have similar distributions. Correlated terms and coefficients like these increase are challenging for data-driven machine learning models to identify correct PDE terms.

Table A6: Model performance under larger noisy levels for PDEs.

Metrics	Recall (%)		Coefficient Error		Fitting Error	
Noise Level	25%	30%	25%	30%	25%	30%
Burgers' Equation	100	100	0.008	0.011	0.001	0.002
KdV Equation	100	100	0.018	0.021	0.466	0.559
C-I Equation	66	66	6.145	6.270	0.302	0.359
C-D Equation	50	50	1.254	1.276	0.043	0.050

F More Experimental Results

In this section, we visualize the estimated coefficients and compare them with the ground-truth coefficients. The visualized residual errors of the estimated coefficients show that the proposed our model is very accurate in coefficient estimation. The results of 1-HNC is shown in Fig. 4 in the main manuscript, so we only show the results of the other four cases of the governing equation of underground seepage here and the two cases of the convection diffusion equation. The estimated coefficients of all datasets are all obtained with the optimal hyperparameters tuned for each dataset.

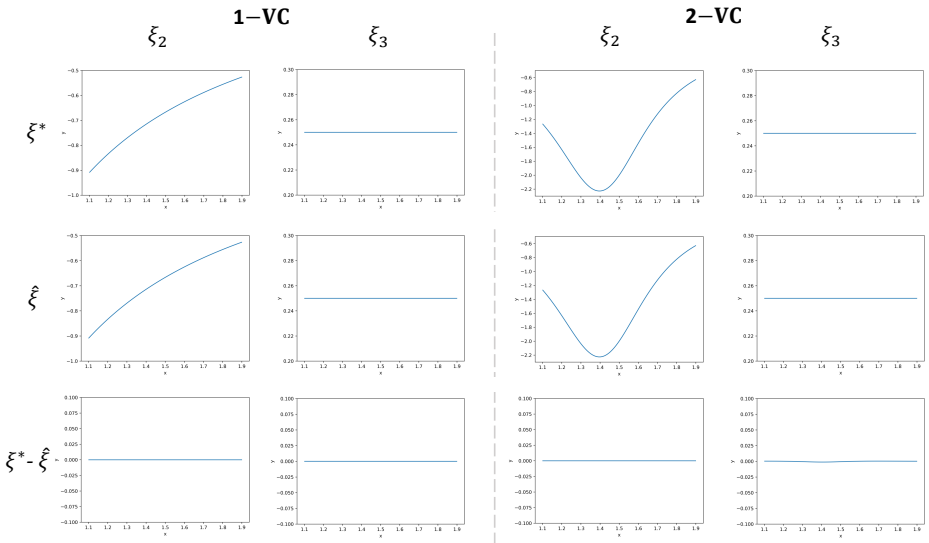

Figure A3: Comparison of estimated and correct nonzero coefficients of the two convection diffusion equation cases. Three rows represent the results of Ground Truth, our model and their residual errors, respectively. The two columns in each side represent two ξ_i that are nonzero in reality. Each sub-figure shows the value at each (x,) spatial coordinate. The scales in some sub-figures are different.

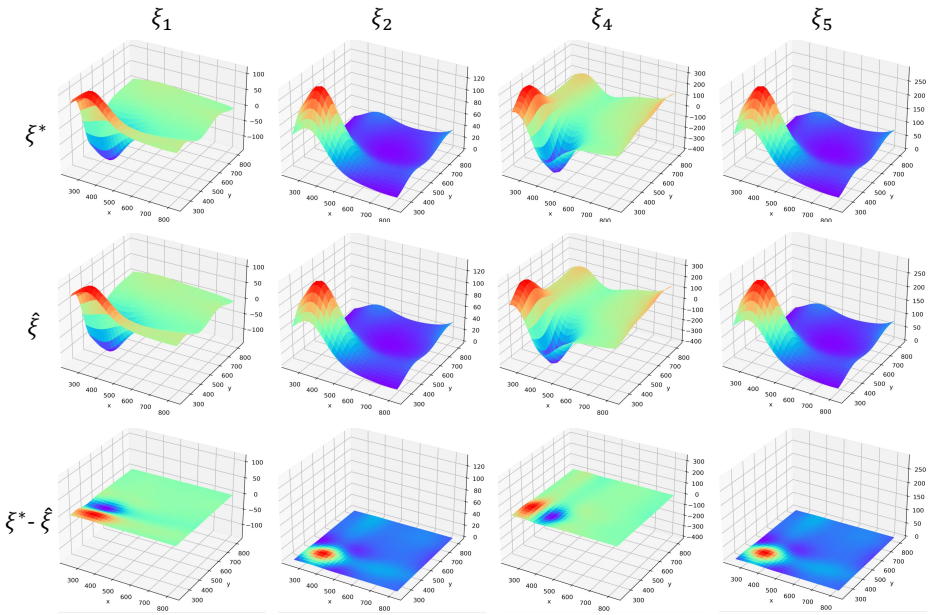

Figure A4: Comparison of estimated and correct nonzero coefficients of 2-HNC. Three rows represent the results of Ground Truth, our model and their residual error, respectively. Four columns represent four ξ_i that are nonzero in reality. Each sub-figure shows the value at each (x,y) spatial coordinate.

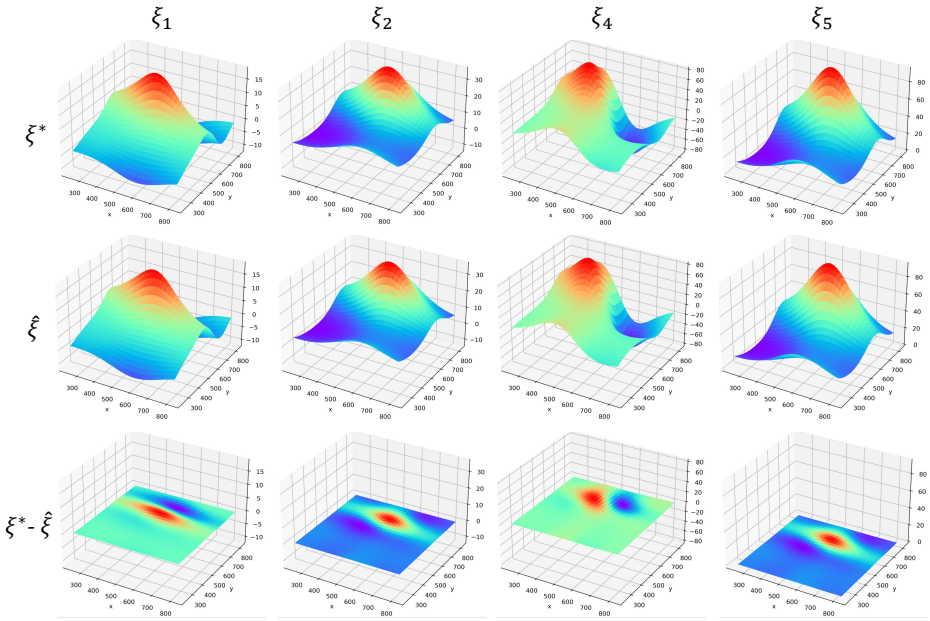

Figure A5: Comparison of estimated and correct nonzero coefficients of 3-HNC. Three rows represent the results of Ground Truth, our model and their residual error, respectively. Four columns represent four ξ_i that are nonzero in reality. Each sub-figure shows the value at each (x,y) spatial coordinate.

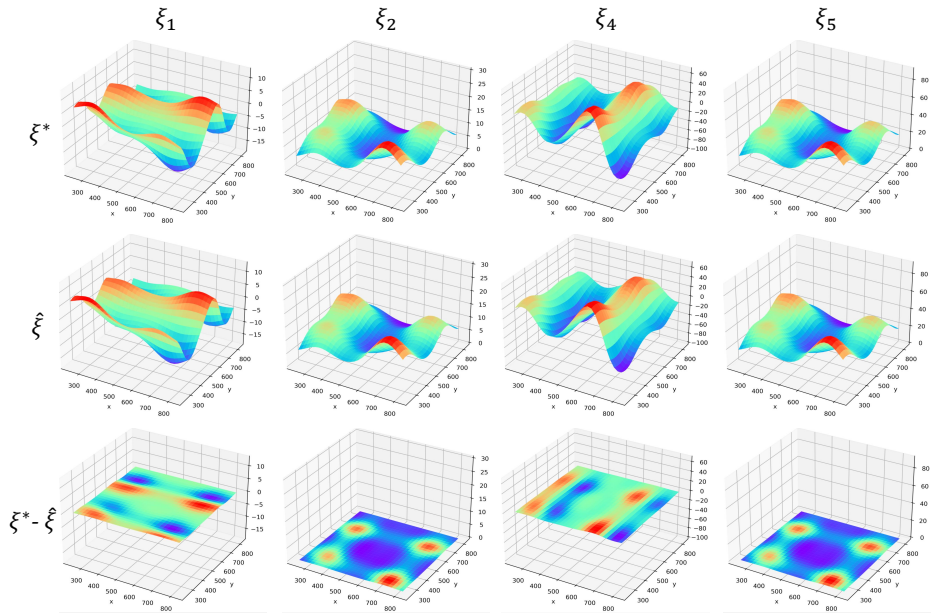

Figure A6: Comparison of estimated and correct nonzero coefficients of 4-HNC. Three rows represent the results of Ground Truth, our model and their residual error, respectively. Four columns represent four ξ_i that are nonzero in reality. Each sub-figure shows the value at each (x,y) spatial coordinate.

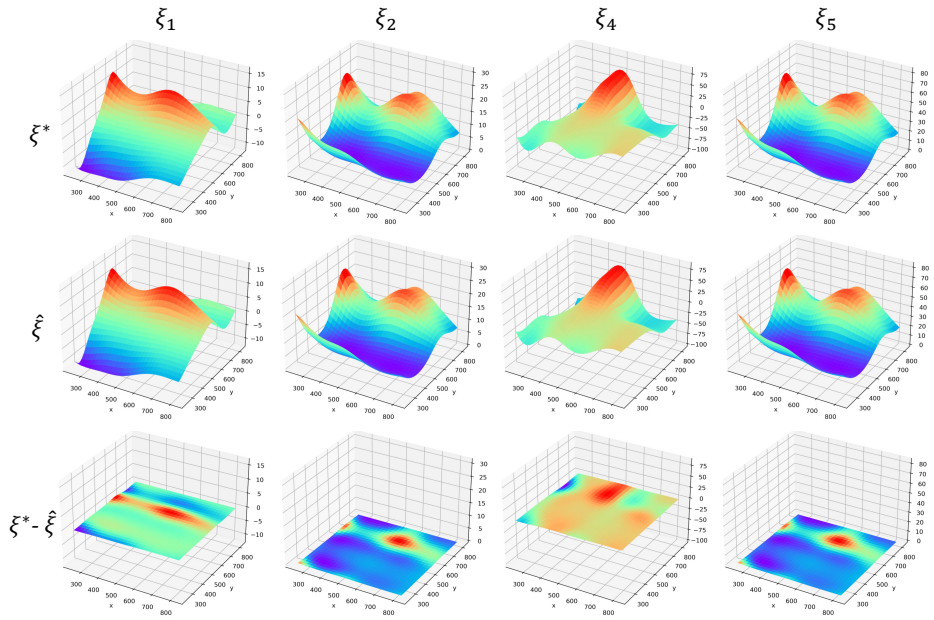

Figure A7: Comparison of estimated and correct nonzero coefficients of 5-HNC. Three rows represent the results of Ground Truth, our model and their residual error, respectively. Four columns represent four ξ_i that are nonzero in reality. Each sub-figure shows the value at each (x,y) spatial coordinate.