UNIVERSITY OF CALIFORNIA SAN DIEGO

Machine learning based partial differential equation (PDE) recovery

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Ruixian Liu

Committee in charge:

Peter Gerstoft, Chair
Michael Bianco
Michael J. Buckingham
Wenyuan Fan
William S. Hodgkiss
Truong Quang Nguyen
Bhaskar D. Rao

2023

The Dissertation of Ruixian Liu is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

TO MY FAMILY, MY FRIENDS AND COLLABORATORS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Thank you to my advisor Prof. Peter Gerstoft for guiding me to complete this PhD degree with your insightful advice on the choice of research path. Sincere thanks to Prof. Bhaskar D. Rao for sharing his expertise on sparse regression and for his indispensable assistance in editing the papers as a coauthor. Deep gratitude to Dr. Michael Bianco who greatly helped me to understand the physics informed neural networks and collaborated with me in multiple papers. My sincere appreciation is also for all professors in my PhD committee including Prof. Truong Q. Nguyen, Prof. Michael J. Buckingham, Prof. William S. Hodgkiss and Prof. Wenyuan Fan.

Thanks to all my Noiselab labmates for making UCSD fun and memorable. Sincerely hope we could cross paths again.

Thanks to my family for supporting my education and career, and for the unconditional affection you have bestowed upon me.

Profound appreciation is extended to my cherished friends for their unwavering encouragement in my advancement through thorns.

This dissertation is a collection of papers that were published or have been prepared for publication. The text of Chapter 2, in full, is a reprint of the material as it appears in R. Liu, M. Bianco, P. Gerstoft, "Automated Partial Differential Equation Identification", *The Journal of the Acoustical Society of America*, 150(4), 2364-2374, 2021. The dissertation author was the primary researcher and author of Chapter 2. The coauthors listed in this publication directed and supervised the research.

The text of Chapter 3, in full, is a reprint of the material as it appears in R. Liu, P. Gerstoft, M. Bianco, B. D. Rao, "Recovery of Spatially Varying Acoustical Properties via Automated PDE Identification", *The Journal of the Acoustical Society of America*, 153(6), 3169-3180, 2023. The dissertation author was the primary researcher and author of Chapter 3. The coauthors listed in this publication directed and supervised the research.

The text of Chapter 4, in full, is a reprint of the material as it appears in R. Liu, P. Gerstoft, "SD-PINN: Deep Learning based Spatially Dependent PDEs Recovery", which is submitted to

IEEE Access. A preliminary version of this work is published as R. Liu, P. Gerstoft, "SD-PINN: Physics informed neural networks for spatially dependent PDEs", *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023. The dissertation author was the primary researcher and author of Chapter 4. The coauthors listed in this publication directed and supervised the research.

VITA

2017	B. Eng. in Communication Engineering, Shanghai University, China
2019	M. S. in Electrical Engineering (Signal and Image Processing), University of California San Diego
2023	Ph. D. in Electrical Engineering (Signal and Image Processing), University of California San Diego
2022–2023	Research Assistant, Oracle Corporation

PUBLICATIONS

Liu, R., Bianco, M. and Gerstoft P. "Wave Equation Extraction from a Video using Sparse Modeling", *Proceedings of 53rd Annual Asilomar Conference on Signals, Systems, and Computers (ACSSC)*, pp. 2160-2165, 2019

Liu, R., Bianco, M. and Gerstoft P. "Automated Partial Differential Equation Identification", *The Journal of the Acoustical Society of America*, vol. 150, pp. 2364–2374, 2021

Liu, R., Bianco, M., Gerstoft P. and Rao B. D. "Data-driven Spatially Dependent PDE Identification", *Proceedings of 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* pp. 3383–3387, 2022

Liu, R., Gerstoft P., Bianco, M. and Rao B. D. "Recovery of Spatially Varying Acoustical Properties via Automated PDE Identification", *The Journal of the Acoustical Society of America*, vol. 153, pp. 3169–3180, 2023

Liu, R. and Gerstoft P. "SD-PINN: Physics informed neural networks for spatially dependent PDEs", *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023

Gerdes, M. T., Wang, Y., Wei, X., Wang, G. C., Liu, R. and Gross K. C. "Vibration Resonance Spectrometry (VRS) for the Advanced Streaming Detection of Rotor Unbalance", *Machines*, vol. 11, no. 6, pp. 639, 2023

Liu, R. and Gerstoft P. "SD-PINN: Deep Learning based Spatially Dependent PDEs Recovery", *submitted to IEEE Access*, 2023

ABSTRACT OF THE DISSERTATION

Machine learning based partial differential equation (PDE) recovery

by

Ruixian Liu

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California San Diego, 2023

Peter Gerstoft, Chair

The dynamics of many natural phenomena are described by partial differential equations (PDEs). With the rise in computing resources, machine learning methods for PDE recovery directly from observations are emerging. Unlike traditional PDE derivation, a machine learning method requires less mathematical prowess and is widely applicable to various dynamical systems. This dissertation uses two machine learning methods, sparse modeling and physics informed neural networks, to address PDE recovery issues: (1) identifying unknown PDE terms; (2) recovering PDE coefficients from noisy, partial observations. We tackle both spatially-independent and spatially-dependent PDEs, with the latter's recovery elucidating the spatial variations of medium material properties, essential for modern industrial applications.

First, sparse modeling methods are used to identify unknown PDEs. A dictionary with redundant hypothetical PDE terms is computed numerically from observations, and then sparse modeling approaches extract some of the terms from this dictionary to form the identified PDE. For spatially-dependent PDE identification, a dictionary is computed from the observations at each location, and then sparse regression is used to extract active terms from the dictionary to be the PDE terms at this location. The methods are validated on both synthetic datasets and real laser measurements of structural vibrations.

Next, a deep learning method spatially dependent physics informed neural network (SD-PINN) is used to recover spatially-dependent PDEs from noisy and partial observations. The spatially-dependent PDE coefficients for each term at all locations in the region of interest (ROI) are modeled as a low-rank coefficient matrix. The low-rank assumption is from the fact that the physical property of the material at one location is impacted by its surroundings, which results in decreased degrees of freedom for the entries in the coefficient matrix. The coefficient matrix is rewritten as a product of two smaller matrices, where "smaller" is compared to the minimal value of the row number and column number of the coefficient matrix. These two matrices contain remarkably fewer unknowns than the coefficient matrix itself. Given only noisy observations from a subset of locations within ROI, SD-PINN can efficiently recover these two matrices and thus reconstruct the coefficient matrix, which encapsulates the PDE coefficients for all locations.

Chapter 1

Introduction

Many natural dynamical phenomena are controlled by partial differential equations (PDEs). Conventional techniques for modeling these dynamical systems using PDEs often hinge on underlying physical laws, but pinpointing the correct PDE terms through analytical means can prove challenging [1]. As a result, the development of more practical, data-driven methods for identifying PDEs has become a focal point of rigorous research.

The goal of this dissertation is to develop and apply machine learning methods to better analyze the measurements of dynamical fields and recover the governing PDEs for them. The methods in this dissertation involves sparse modeling approaches and physics informed neural network (PINN), and the PDEs to be recovered involves spatially-independent PDEs and spatially-dependent PDEs with the latter one indicating how the physical properties of the medium vary spatially. Overall, two kinds of problems are solved:

- *Recovering unknown PDE terms.* The sparse modeling techniques are used to numerically compute all the potential PDE terms from the measurements data and identify the PDE terms that are most possible to be the active terms in the PDE governing this dynamical field. This task applies for both spatially-independent PDEs and spatially-dependent PDEs, and relative methods are discussed in Chapter 2 and 3.

- *Recovering unknown PDE coefficients for given PDE terms for noisy situations and missing measurements.* When the noise is high, the PINN exhibit its unique advantage of noise-

robustness. The traditional PINN only applies to spatially-independent PDEs recovery. This limitation is overcome in our proposed Spatially Dependent Physics Informed Neural Network (SD-PINN) in which we keep its strength on noise-robustness and extend it to enable its capability of recovering spatially-dependent PDEs even at locations without available measurements.

1.1 Basics of PDEs recovery

Many natural phenomena are described by parial differential equations (PDEs) which consist of multiple PDE terms. A PDE governing the dynamical behaviors of field U can be described by

$$\mathsf{N}[U] = a_1 U_x + a_2 U_y + a_3 U_t + a_4 U_{tt} + \dots \tag{1.1}$$

where the partial derivatives $U_x, U_y, U_t, \dots$ are the PDE terms and the $a_1, a_2, \dots$ are PDE coefficients. The PDE coefficients can be fixed (spatially-independent) across the whole region of interest (ROI) or be spatially-dependent, which indicates an inhomogeneous medium for the dynamics.

There are many types of PDEs, like the wave equation for the fluid dynamics, the heat equation for the heat diffusion and Maxwell's equation in electromagnetism. In this dissertation, we mainly use the wave equation as an example to demonstrate machine learning based PDE recovery approaches:

$$\mathsf{N}[U] = U_{tt} + \alpha U_t - c^2 \nabla^2 U \ , \tag{1.2}$$

where U_t, U_{tt} are the 1st and 2nd-order temporal derivatives of U and $\nabla^2 U = U_{xx} + U_{yy}$ the sum of 2nd-order spatial derivatives of U (also called the Laplacian of U). The PDE coefficient $\alpha \geq 0$ is the wave attenuation factor and $c^2 > 0$ is the square of the phase speed c.

In this dissertation, we assume that the sensors measuring the dynamical field is either placed in 1D spatially or 2D forming a rectangular array. With the additional dimension for time, the measurements data denoted by $\mathbf{U}$, which is a measured field potentially satisfying a PDE, is a

2D or 3D matrix.

The machine learning based PDE recovery methods contain two steps: (1) computing the partial derivatives from $\mathbf{U}$ as the potential active PDE terms; (2) computing the coefficients with respect to these terms. Different methods have various computing schemes, but overall these two steps always exist.

1.2 PDE identification using sparse modeling

In real scenarios, the number of PDE terms is limited. This inspires the PDE identification methods based on sparse regression, in which a dictionary $\mathbf{\Phi}$ containing redundant hypothetical active PDE terms is first computed based on the measurements $\mathbf{U}$ and then a sparse regression is applied to select only a few of the terms to be the true active ones. Since the true active terms are sparsely distributed within the dictionary, the sparse regression is viable. For example, such a dictionary $\mathbf{\Phi}$ containing D terms may be

$$\mathbf{\Phi} = [\mathbf{u}_t,\ \mathbf{u}_{tt},\ \mathbf{u}_x,\ \mathbf{u}_{xx},\ \mathbf{u}_{xxx},\ \mathbf{u}\mathbf{u}_x,\ \mathbf{u}_y,\ \mathbf{u}_{yy},\ \mathbf{u}_{yyy},\ \mathbf{u}\mathbf{u}_y] \in \mathbb{R}^{N\times D} \tag{1.3}$$

where each column is the stack of values for one PDE term computed at all spatio-temporal coordinates in the ROI directly from $\mathbf{U}$ by numerical differentiation, e.g., the finite difference. If the PDE governing the field is the wave equation (1.2), then the terms $\{\mathbf{u}_t,\ \mathbf{u}_{tt},\ \mathbf{u}_{xx},\ \mathbf{u}_{yy}\}$ should be extracted via sparse regression.

For the spatially-dependent PDE case, the dictionary is separately computed for every location. For example, a dictionary $\mathbf{\Phi}_n$ is built for the location of index n in the ROI, and the active PDE terms at this location are extracted from $\mathbf{\Phi}_n$ during which the PDE coefficients specific to this location are also recovered.

There are already sound research on this topic. In the PDE identification work "PDE-FIND" [1], the sequential threshold ridge regression (STRidge) is applied to select the active terms from the dictionary. In the work [2], a sparse Bayesian learning (SBL) technique is

leveraged to locate the active terms. In this dissertation, we introduce a cross-validation based PDE identification method [3] for spatially-independent PDEs as detailed in Chapter 2 and an ℓ_1-norm minimization based approach [4] for spatially-dependent PDEs as in Chapter 3. A common advantage of works [3] and [4] is that they do not require any assumed active PDE terms.

1.3 Physics informed neural network

The physics informed neural network (PINN) [5, 6, 7] is a method to recover the PDE coefficients for given PDE terms from measurements using a feedforward neural network (FNN) parametrized by θ, which includes weights and bias of all layers. The input of the FNN is the spatio-temporal coordinate $(\boldsymbol{x},t)$ in the ROI (where $\boldsymbol{x}$ is a vector in 2D cases), and the output is the estimated dynamical field measurement $\widehat{U}(\boldsymbol{x},t)$ at there. Suppose the true measurements of the field $U(\boldsymbol{x},t)$ are available with the region Ω_{u} inside the ROI, the loss function

$$loss_{\mathrm{u}}(\theta) = \sum_{(\boldsymbol{x},t)\in\Omega_{\mathrm{u}}} (\widehat{U}(\boldsymbol{x},t) - U(\boldsymbol{x},t))^2 \tag{1.4}$$

is minimized during training to encourage the FNN to output better dynamical field estimation. Meanwhile, the PDE terms, i.e., the partial derivatives of U, are estimated by FNN using automatic differentiation [8]. Then these estimations of PDE terms are used to recover their coefficients by minimizing a $loss_{\mathrm{f}}$. Take the wave equation (1.2) for example, the $loss_{\mathrm{f}}$ is

$$loss_{\mathrm{f}}(\widehat{\alpha},\widehat{c},\theta) = \sum_{(\boldsymbol{x},t)\in\Omega_{\mathrm{f}}} (\widehat{U}_{tt}(\boldsymbol{x},t) + \widehat{\alpha}\widehat{U}_t(\boldsymbol{x},t) - \widehat{c}^2\nabla^2\widehat{U}(\boldsymbol{x},t))^2 \tag{1.5}$$

where $\widehat{U}_{tt},\ \widehat{U}_t,\ \nabla^2\widehat{U} = \widehat{U}_{xx} + \widehat{U}_{yy}$ are partial derivatives estimated by automatic differentiation and Ω_{f} are the collection of coordinates used for computing $loss_{\mathrm{f}}$. The weighted sum of the two

losses forms the overall *loss* by minimizing which the PINN is trained:

$$loss(\theta, \widehat{\alpha}, \widehat{c}) = w_{\mathrm{u}} \times loss_{\mathrm{u}} + w_{\mathrm{f}} \times loss_{\mathrm{f}} \, . \tag{1.6}$$

The PINN is robust to noise, because the similarity between the $\widehat{U}$ estimated by FNN and the true U is constrained by the minimization of $loss_{\mathrm{f}}$. To be specific, when U is noisy, the ability of the FNN to model the dynamical field is constrained by $loss_{\mathrm{f}}$, inhibiting its capacity to simulate every detail. Consequently, the FNN's simulation of the additive noise affecting the field is also suppressed and the output of FNN $\widehat{U}$ is thus denoised. The automatic differentiation on this denoised $\widehat{U}$ will benefit the calculation of PDE terms and thus benefit PDE recovery. In addition, the automatic differentiation is more accurate than other numerical differentiation approaches, such as the finite difference [9, 10].

1.4 Dissertation overview

This dissertation applies machine learning based methods to a diverse set of problems in PDE recovery. Our results demonstrate that the sparse modeling methods can identify the unknown PDEs without any active PDE terms assumed active *a priori*, and the PINN can be extended to recover spatially-dependent PDEs from noisy and incomplete measurements.

Chapters 2 and 3 address PDE identification using sparse modeling approaches. In Chapter 2, a cross validation with a sparsity penalty algorithm is proposed to select the active PDE terms from a dictionary of redundant terms computed from the measurements. Unlike previous developments which assume at least one active PDE term, here the assumption of any active PDE term is not required and the method can automatically locate all active terms within the dictionary. The method is applied to an Aluminum plate to recover the dispersion properties of the waves propagating on it. In Chapter 3, an ℓ_1-norm minimization based method is proposed to recover the spatially-dependent PDEs from measurements. In this work, no assumption for active PDE terms is required as well, and the method is computationally efficient. This method

can also be used with a denoising approach based on integration transformation to recover the PDEs from noisy measurements.

Chapter 4 describes a spatially-dependent PDE coefficients recovery approach using an extension of physics informed neural networks, i.e., the SD-PINN. From the (potentially noisy) dynamical field measurements at some locations in the ROI, the method is able to recovery the spatially-dependent PDEs at all locations including those without available measurements. Some matrix analysis knowledge including low-rank factorization is involved.

The dissertation is concluded in Chapter 5, in which the sparse modeling approaches and neural network based methods for PDE recovery are summarized.

Chapter 2

Automated Partial Differential Equation Identification

2.1 Introduction

Partial differential equations (PDEs) govern many natural dynamical phenomena. Traditional methods for modeling dynamical systems with PDEs are typically based on physical principles, and analytically determining the correct PDE terms can be difficult [1]. Thus the more applicable data-driven PDE identification methods are the subject of intensive research.

There has been significant development in data-driven PDE extraction theory thanks to the advancements in physics-informed machine learning [1, 2, 7, 11, 12, 13, 14, 15, 16, 18]. Our exploration is inspired by recent work in sparse modeling [1, 15]. Sparse modeling [19, 20] assumes a parsimonious data representation [21, 22] that scales well to big data problems and has obtained compelling results in many related fields [23, 24]. Early applications of sparse, data-driven PDE estimation to real data has appeared [25, 26, 27].

Often we have a priori assumption for the PDE and then retrieve relevant terms. In previous PDE-discovery developments, one or more active PDE terms (e.g., the 1st order time derivative term [1, 15] or multiple PDE terms [7]) are assumed a priori for the PDE. The other contributing terms together with their coefficients are then derived from this prior information. Thus only parts of the PDE is found by data-driven approaches. This can be problematic when the assumed existing term is not obvious, e.g., to identify the governing PDE for a surface wave

which may either be an inviscid Burgers equation or a non-attenuating wave equation who share no PDE terms in common, one must specify the correct existing term according to sufficient prior knowledge.

To alleviate the data-driven PDE identification method's reliance on the prior information, the proposed approach can *automatically* identify all contributing terms consisting the PDE for the dynamics shown by the data. The method computes a dictionary of hypothetical PDE terms from data using finite difference and pseudo spectral methods, and selects the contributing terms using sparsity and resampling. We show that the wave-, Burgers- and Helmholtz-equation are well-identified from data.

2.2 Theory

From a given observed field, the inverse problem solves the background parameters generating the field. Often the inverse problem is solved under strong assumptions as only source locations are unknown or it is a wave guide problem. The PDE generating the field has been assumed known. We relax this assumption and solve for the PDE that could have generated the observations.

2.2.1 Background

Consider a field $U(x,y,t)$ across spatial x,y and temporal t coordinates. Let it be governed by a PDE $\mathsf{N}[U(x,y,t)]$ with $f(x,y,t)$ the source term

$$\mathsf{N}[U(x,y,t)] = f(x,y,t) \tag{2.1}$$

with corresponding spatial and temporal boundary conditions. We are here concerned with discovering the homogeneous PDE $\mathsf{N}[U(x,y,t)] = 0$, thus $f(x,y,t) = 0$.

Examples of $\mathsf{N}[U]$ with the typical 2–3 terms include

$$\mathsf{N}[U] = \frac{\partial^2 U}{\partial t^2} + \alpha \frac{\partial U}{\partial t} - c^2 \nabla^2 U \ , \qquad \text{Wave eq.} \tag{2.2}$$

$$\mathsf{N}[U] = \omega^2 U + c^2 \nabla^2 U \ , \qquad \text{Helmholtz eq.} \tag{2.3}$$

$$\mathsf{N}[U] = \frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} - \nu \frac{\partial^2 U}{\partial x^2}. \qquad \text{Burgers eq. in 1D} \tag{2.4}$$

In many physical environments the exact form of $\mathsf{N}(U)$ is unknown. Consider the general form with D terms

$$\mathsf{N}[U] = a_1 \frac{\partial U}{\partial x} + a_2 \frac{\partial U}{\partial y} + a_3 \frac{\partial U}{\partial t} + a_4 \frac{\partial^2 U}{\partial t^2} + \dots \tag{2.5}$$

often up to 2nd order is assumed, but 4th order is not uncommon. Non-linear terms like $U \frac{\partial U}{\partial x}$ can appear (2.4), and the time derivative might be absent (2.3).

Consider the data of the form $\mathbf{U} \in \mathbb{C}^{N_x \times N_y \times N_t}$ for N_x horizontal N_y vertical and N_t temporal points, with stepsize Δx, Δy and Δt. The field is generated by a physical source or perturbed initial condition, and propagates through the media. We identify PDEs governing the field from the region of interest (ROI), which is $\mathbf{U}$ excluding the near-field for potentially existing sources and the spatial-temporal boundary regions where derivatives are not defined.

2.2.2 Building a dictionary

From $\mathbf{U}(i_x, i_y, i_t)$ we obtain hypothetical PDE terms by evaluating derivatives at all points in ROI. The derivatives are estimated using numerical methods including finite difference [9] and the pseudo-spectral approach [28]. At every point, the homogeneous PDE like (2.5) is satisfied as

$$a_1 U_x(i_x, i_y, i_t) + a_2 U_y(i_x, i_y, i_t) + \dots = 0 \ . \tag{2.6}$$

In vector notation (2.6) becomes

$$\begin{aligned} \boldsymbol{\varphi}^{\mathrm{T}}(i_x, i_y, i_t)\mathbf{a} = 0\ , \quad \mathbf{a} = [a_1 \dots a_D]^{\mathrm{T}} \\ \boldsymbol{\varphi}^{\mathrm{T}}(i_x, i_y, i_t) = [U_x(i_x, i_y, i_t)\ U_y(i_x, i_y, i_t)\ \dots]. \end{aligned} \tag{2.7}$$

For all points in ROI, we obtain

$$\boldsymbol{\Phi}\mathbf{a} = \mathbf{0}, \quad \boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\varphi}^{\mathrm{T}}(1_x, 1_y, 1_t) \\ \vdots \\ \boldsymbol{\varphi}^{\mathrm{T}}((N)_x, (N)_y, (N)_t) \end{bmatrix} \in \mathbb{C}^{N \times D}, \tag{2.8}$$

with $N < N_x N_y N_t$, $\boldsymbol{\varphi}(i_x, i_y, i_t)$ all hypothetical PDE terms evaluated at (i_x, i_y, i_t), and $\mathbf{a} \in \mathbb{C}^D$ the PDE term coefficients. $(N)_x$ is i_x when row index $i = N$.

Each column of $\boldsymbol{\Phi}$ contains values for one PDE term evaluated at every point in ROI. Denote the dth column as $\boldsymbol{\phi}_d$, from (2.8), we rewrite $\boldsymbol{\Phi}$ having $D = 14$ terms used for our experiments with indices shown in superscripts as

$$\boldsymbol{\Phi} = [\ \boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_D] = [\ \overset{1}{\mathbf{1}}\ \overset{2}{\mathbf{u}_t}\ \overset{3}{\mathbf{u}_{tt}}\ \overset{4}{\mathbf{u}}\ \overset{5}{\mathbf{u}_x}\ \overset{6}{\mathbf{u}_y}\ \overset{7}{\mathbf{u}\circ\mathbf{u}_x}\ \overset{8}{\mathbf{u}\circ\mathbf{u}_y}\ \overset{9}{\mathbf{u}_{xx}}\ \overset{10}{\mathbf{u}_{xy}}\ \overset{11}{\mathbf{u}_{yy}}\ \overset{12}{\mathbf{u}\circ\mathbf{u}_{xx}}\ \overset{13}{\mathbf{u}\circ\mathbf{u}_{xy}}\ \overset{14}{\mathbf{u}\circ\mathbf{u}_{yy}}] \tag{2.9}$$

with $\mathbf{u} = \mathrm{vec}(\mathbf{U}) \in \mathbb{C}^N$, subscripts indicating numerical differentiation, $\mathbf{1}$ is all-ones vector, and $\circ$ the Hadamard (element-wise) product. $\boldsymbol{\Phi}$ contains common terms for multiple PDEs possibly governing $\mathbf{U}$ in ROI including spatial, temporal derivatives of various orders and non-linear terms. Only a few of these are in the true PDE, i.e., $\|\mathbf{a}\|_0 \ll D$ with $\|\cdot\|_0$ the number of non-zero entries.

To calculate $\boldsymbol{\Phi}$, the 2nd order finite difference (FD) we use for 1st and 2nd order derivatives is computed by approximating analytical derivatives using truncated Taylor series. With step Δx, its truncation error is $O(\Delta x^2)$ [9]. For FD, the first and last pixels in each dimension are not considered as ROI.

The pseudo-spectral (PS) method [28, 29] is typically more accurate than finite difference, as it is the limit of finite difference approximations when the order tends to infinity [30]. The PS is based on Fourier transform. Denote some discrete signal along x-axis for fixed y, t as $\mathbf{u}(x) = \mathbf{U}(:,y,t)$ in $\mathbb{C}^{N_x}$, with its spectral coefficients $\tilde{u}_r$ obtained by $\tilde{u}_r = \frac{1}{N_x}\sum_{j=0}^{N_x-1}\mathbf{u}(x_j)e^{-2\pi i jr/N_x}, i = \sqrt{-1}$, the pth order derivative is

$$\partial_x^{(p)}\mathbf{u}(x_j) = \sum_{r=-N_x/2+1}^{N_x/2}(ik_r)^p\tilde{u}_r e^{ik_r x_j},\ j = 0,\ldots,N_x-1 \tag{2.10}$$

where the wavenumber $k_r = \frac{2\pi}{\Delta x}\frac{r}{N_x}$. To avoid issues at the spatial boundaries, Tukey windowing is used to preserve most parts of signal. In all experiments, for each dimension, 40% of the signal is tapered and excluded from the ROI, with 20% at either end.

2.2.3 Identifying PDE terms

Beyond the assumption that the representation is parsimonious, we assume no prior intuition of which PDE terms in the library should be relevant to a given problem. The approach is data-driven as we rely on cross-validation to obtain coefficients, which is a commonly used technique in machine learning to avoid fitting noise due to redundant terms. The proposed method which is non-recursive and free of the assumption for i.i.d. Gaussian noise forms an intuitive alternative to the threshold sparse Bayesian learning approach[14], and is summarized in Algo.1 with details in the following.

Because of noise introduced by derivatives computing and measurements, the equality in (2.8) may not hold. To enforce parsimony and avoid the trivial $\mathbf{a} = \mathbf{0}$, we assume there being one term $\boldsymbol{\phi}_j$ in $\boldsymbol{\Phi}$ included in the PDE and search for T other terms, thus estimate $\mathbf{a}$ by

Algorithm 1: PDE identification.

Input: $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_D] \in \mathbb{C}^{N\times D}$, λ
Output: $\mathbf{a} = [a_1 \dots a_D]^{\mathrm{T}} \in \mathbb{C}^D$
$\bar{\boldsymbol{\Phi}} = [\bar{\boldsymbol{\phi}}_1 \ \dots \ \bar{\boldsymbol{\phi}}_D]$, where $\bar{\boldsymbol{\phi}}_d = \boldsymbol{\phi}_d / \|\boldsymbol{\phi}_d\|_2$, $\forall d$
for $j = 1 : D$ **do**
  for $\mathrm{T} = 0 : D-1$ **do**
    $\mathrm{L}_j(\mathrm{T}) = \mathrm{CrossValid}(j, \mathrm{T})$ `// Eq.(2.13)`
  $\widehat{\mathrm{T}}_j = \arg\min_{\mathrm{T}} \mathrm{L}_j(\mathrm{T}) + \lambda \mathrm{L}_j(D-1)\mathrm{T}$
  $\widehat{\bar{\mathbf{a}}}_j = \arg\min_{\bar{\mathbf{a}}_j} \|\bar{\boldsymbol{\Phi}}\bar{\mathbf{a}}_j\|_2$, s.t. $\bar{a}_j = 1, \|\bar{\mathbf{a}}_j\|_0 = \widehat{\mathrm{T}}_j + 1$
  $\mathrm{Err}(j) = \|\bar{\boldsymbol{\Phi}}\widehat{\bar{\mathbf{a}}}_j\|_2 / \|\widehat{\bar{\mathbf{a}}}_j\|_2$
$\widehat{j} = \arg\min_j \mathrm{Err}(j)$ `// Choose assumed term`
$\widehat{\mathbf{a}} = \arg\min_{\mathbf{a}} \|\boldsymbol{\Phi}\mathbf{a}\|_2$, s.t. $a_{\widehat{j}} = 1, \|\mathbf{a}\|_0 = \widehat{\mathrm{T}}_{\widehat{j}} + 1$

$$\{\widehat{\mathbf{a}}, \widehat{j}, \widehat{\mathrm{T}}\} = \arg\min_{\mathbf{a}, j, \mathrm{T}} \|\boldsymbol{\Phi}\mathbf{a}\|_2 \tag{2.11}$$
$$\text{s.t. } a_j = 1, \ \|\mathbf{a}\|_0 = \mathrm{T} + 1, \ \mathrm{T} = \arg\min_{\mathrm{T}'} \psi_j(\mathrm{T}').$$

For a given j, $\|\mathbf{a}\|_0$ is chosen from sparsity-penalized cross-validation error, defined by ψ_j (2.14), as described next. Since D is small ($\sim 10^1$), we cycle through all D columns for j in (2.11) and optimize $\mathbf{a}$, T in every case. Then $\widehat{\mathbf{a}}$ is selected by minimizing a normalized fitting error (defined in (2.15), to be discussed) over all cases.

Specifically, let $\bar{\boldsymbol{\Phi}} = [\bar{\boldsymbol{\phi}}_1 \dots \bar{\boldsymbol{\phi}}_D]$ with $\bar{\boldsymbol{\phi}}_d = \boldsymbol{\phi}_d / \|\boldsymbol{\phi}_d\|_2, \forall d$ be the normalized $\boldsymbol{\Phi}$. Under the assumption $a_j = 1$ for an arbitrary $j \in \{1 \dots D\}$, we solve $\bar{\mathbf{a}}_j = [\bar{a}_1 \dots \bar{a}_D]$ as

$$\widehat{\bar{\mathbf{a}}}_j = \arg\min_{\bar{\mathbf{a}}_j} \|\bar{\boldsymbol{\Phi}}\bar{\mathbf{a}}_j\|_2^2, \quad \text{s.t. } \|\bar{\mathbf{a}}_j\|_0 = \mathrm{T}_j + 1, \ \bar{a}_j = 1. \tag{2.12}$$

The T_j, i.e., the number of non-zero entries other than $\bar{a}_j$ in $\bar{\mathbf{a}}_j$, is chosen using K-fold cross-validation [31] with an additional sparsity penalty. For cross-validation, we evenly divide the rows of $\bar{\boldsymbol{\Phi}}$ into K folds. The kth fold $\bar{\boldsymbol{\Phi}}^k$ is the validation data including the jth

column $\bar{\boldsymbol{\phi}}^k_{j\_\text{val}} \in \mathbb{C}^{\frac{N}{K}}$ and the other columns denoted by $\bar{\boldsymbol{\Phi}}^k_{-j\_\text{val}} \in \mathbb{C}^{\frac{N}{K}\times(D-1)}$. All other folds are concatenated as training data including the jth column $\bar{\boldsymbol{\phi}}^k_{j\_\text{tr}} \in \mathbb{C}^{\frac{K-1}{K}N}$ and the other columns in $\bar{\boldsymbol{\Phi}}^k_{-j\_\text{tr}} \in \mathbb{C}^{\frac{K-1}{K}N\times(D-1)}$. For each fold k, we calculate the coefficient $\widehat{\bar{\mathbf{a}}}^k_{-j\_\text{tr}}(\mathrm{T})$ with T non-zero entries minimizing $\|\bar{\boldsymbol{\phi}}^k_{j\_\text{tr}} + \bar{\boldsymbol{\Phi}}^k_{-j\_\text{tr}}\bar{\mathbf{a}}^k_{-j\_\text{tr}}\|_2^2$. To solve this least squares objective using limited columns of $\bar{\boldsymbol{\Phi}}^k_{-j\_\text{tr}}$, we choose the columns contributing most in the least squares solution, resulting in a threshold least squares (TLS) scheme (detailed in Appendix.2.6.2). The TLS selects locations for non-zero values in $\bar{\mathbf{a}}^k_{-j\_\text{tr}}$ using the entries with T largest magnitudes in the least squares solution for fitting $\bar{\boldsymbol{\phi}}^k_{j\_\text{tr}}$ using all columns in $\bar{\boldsymbol{\Phi}}^k_{-j\_\text{tr}}$. Compared to a classic basis selection method OMP[32], it is non-iterative and can work better when the column in $\bar{\boldsymbol{\Phi}}^k_{-j\_\text{tr}}$ most correlated to $\bar{\boldsymbol{\phi}}^k_{j\_\text{tr}}$ is not active, with an example given in Sec.2.3.3. The loss for cross-validation is (here $K = 5$)

$$\mathrm{L}_j(\mathrm{T}) = \frac{1}{K}\sum_{k=1}^{K} \|\bar{\boldsymbol{\phi}}^k_{j\_\text{val}} + \bar{\boldsymbol{\Phi}}^k_{-j\_\text{val}}\widehat{\bar{\mathbf{a}}}^k_{-j\_\text{tr}}(\mathrm{T})\|_2^2. \tag{2.13}$$

Minimizing (2.13) might not give the correct sparsity due to two reasons: (i) Columns in $\bar{\boldsymbol{\Phi}}$ are often coherent since they are computed from the same $\mathbf{U}$. A newly incorporated column might be well-fitted by linearly combined existing columns, and thus cause $\mathrm{L}_j(\mathrm{T})$ to plateau. For example, consider non-dispersive attenuating waves $U = \mathrm{Re}(e^{-i(k(x+2y)-\omega t)} + e^{-2i(k(2x+y)-\omega t)})$ governed by (2.2) whose complex wavenumber $k \approx \frac{\omega}{c}(1-\frac{i\alpha}{2\omega})$, when $\frac{\alpha}{2c} \approx 0$ we have $U_t \approx -\frac{c}{3}(U_x+U_y)$ (see (2.22) and (2.28) in Appendix.2.6.1), causing $U_{tt} = -\alpha U_t + c^2(U_{xx}+U_{yy}) \approx -\alpha(mU_t + n(U_x+U_y)) + c^2(U_{xx}+U_{yy})$ for some non-zero m and n, i.e., $\mathrm{L}_3(3) \approx \mathrm{L}_3(5)$. (ii) After all the relevant terms being recognized, the incorporated irrelevant columns with small coefficients can fit the noise in $\bar{\boldsymbol{\phi}}_j$ introduced by numerical differentiation, and thus decrease $\mathrm{L}_j(\mathrm{T})$ when T already exceeds the correct sparsity.

To exclude redundant atoms, we incorporate a sparsity penalty term [33], and select the

optimal sparsity as:

$$\widehat{\mathrm{T}}_j = \arg\min_{\mathrm{T}} \psi_j(\mathrm{T}),\ \psi_j(\mathrm{T}) = \mathrm{L}_j(\mathrm{T}) + \lambda \mathrm{L}_j(D-1)\mathrm{T} \tag{2.14}$$

with $\lambda = 1$ chosen empirically working well for our data. The $\mathrm{L}_j(D-1)$ is the average fitting error (2.13) for all folds with all terms used.

With $\widehat{\bar{\mathbf{a}}}_j$ in (2.12) solved by TLS using $\mathrm{T}_j = \widehat{\mathrm{T}}_j$, the normalized fitting error with the range $[0,1]$ is

$$\mathrm{Err}(j) = \|\bar{\mathbf{\Phi}}\widehat{\bar{\mathbf{a}}}_j\|_2 / \|\widehat{\bar{\mathbf{a}}}_j\|_2. \tag{2.15}$$

where $\mathrm{Err}(j) = 1$ indicates $\widehat{\mathrm{T}}_j = 0$ and thus $\bar{\boldsymbol{\phi}}_j$ can not be fitted properly by other columns of $\bar{\mathbf{\Phi}}$. Calculating $\mathrm{Err}(j)$ for all j, and $\widehat{j} = \arg\min_j \mathrm{Err}(j)$ is chosen. Setting $j = \widehat{j}$, $\mathrm{T} = \widehat{\mathrm{T}}_{\widehat{j}}$, and letting $a_j = 1$ in (2.11) provides $\widehat{\mathbf{a}}$.

We verify this PDE identification approach by both simulation and real data experiments as to be described in the following sections.

2.3 Synthetic experiments

Three sets of experiments are carried out, i.e., identifying (i) wave equations from 3D spatial-temporal areas; (ii) Helmholtz equations from 2D spatial areas; (iii) Burgers equations from 2D spatial-temporal areas. Datasets used for (ii) are from the frequency components of wavefields used for (i).

2.3.1 Wave equation

The PDE identification is tested with videos $\mathbf{U}$ sampled from continuous wavefields generated by the wave equation (2.2). Cylindrical propagation is assumed, since we are modeling a plate. For a source f at (f_x, f_y) and field at (x_0, y_0) with an Euclidean distance d to the source, $\mathbf{U}(x_0, y_0, t) = A\frac{e^{\mathrm{Im}(k)d}}{\sqrt{d}} f(t - \frac{d}{c})$ where k is wavenumber, c phase speed, A amplitude. The $\mathrm{Im}(k)$ is determined by (2.2), see Appendix A.

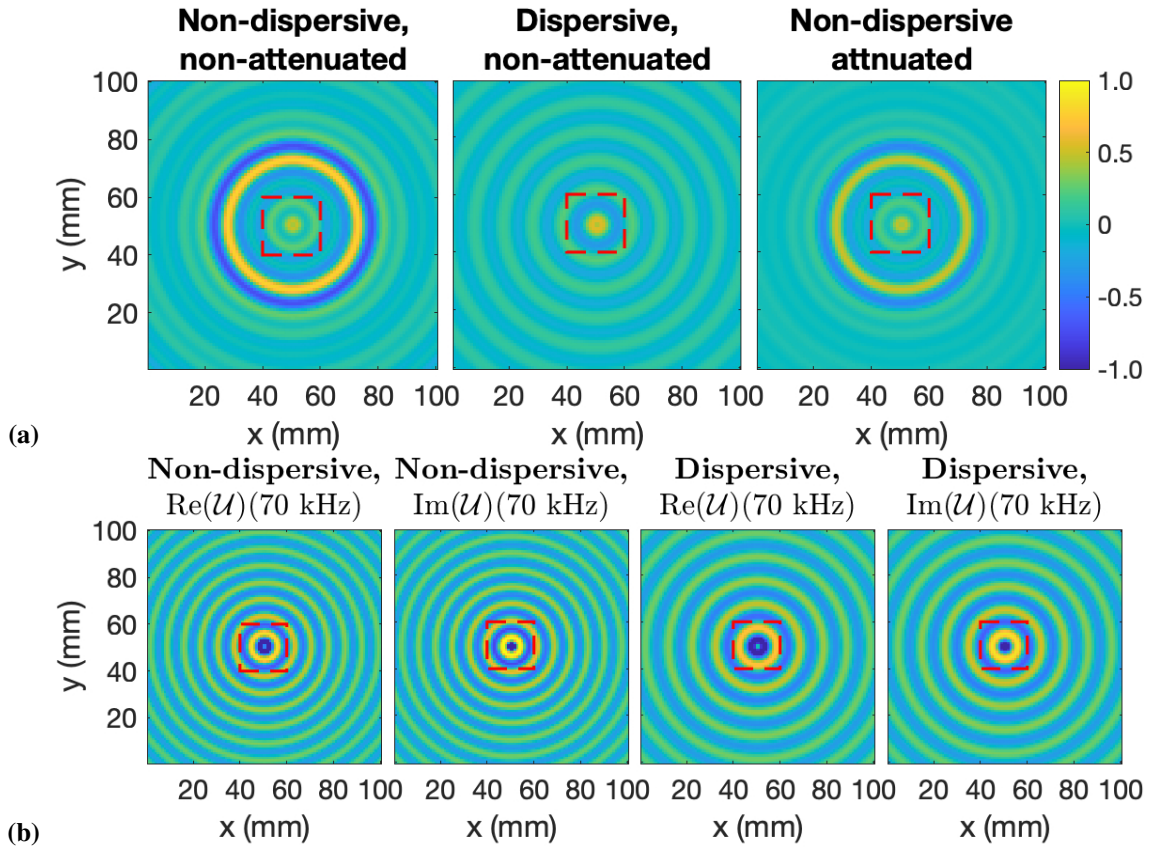

Figure 2.1. (a) Frame 50 for the synthetic wave equation and (b) frequency components of 70 kHz for the synthetic Helmholtz equation. Pixels inside the red square are not considered.

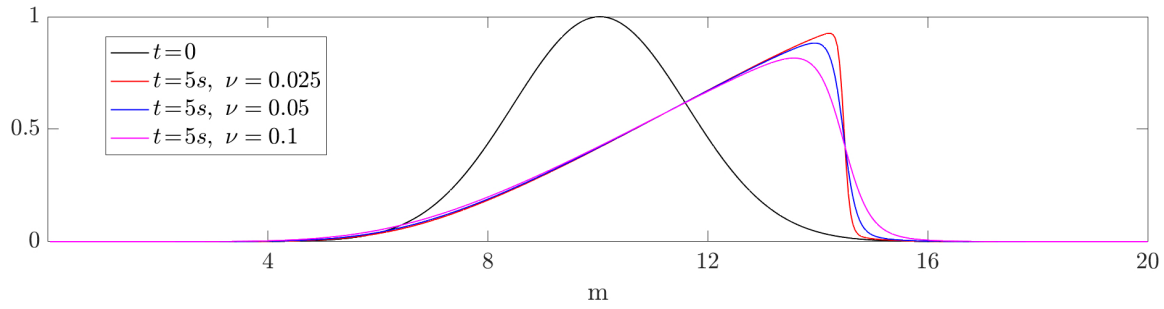

Figure 2.2. The initial state and the waveforms at t=5 s corresponding to Burgers equations with various ν.

We simulate propagation similar to the metal plate in the real data. Three videos with N_x,N_y,N_t set to 100, and $\Delta x = \Delta y = 0.001$ m and $\Delta t = 10^{-6}$ s are generated with free boundaries. The source $f(t)$ is at the center $(50.5\Delta x, 50.5\Delta y)$ and formed by summing 5 sinusoidals 30, 40, 50, 60, and 70 kHz with unit magnitude and zero phase at $t = 0$. Each video shows a specific wavefield, (1) non-dispersive non-attenuated wave, (2) dispersive non-attenuated wave; (3) non-dispersive attenuated wave. The 50th frame for each of them is in Fig.1(a).

The field $\{(i_x, i_y, i_t) | 46 \leq i_x \leq 55; 46 \leq i_y \leq 55; \forall i_t\}$ near the source is dropped when extracting the PDE. We extract the PDEs for the waves at each frequency provided by a bank of ideal bandpass filters. Since $u_{tt} \approx -k^2 u$ with a constant k always holds for narrow band signals ($|\bar{\boldsymbol{\phi}}_3^{\mathrm{T}} \bar{\boldsymbol{\phi}}_4| > 0.99$ in our experiments), we do not consider $\mathbf{u}$ in (2.9).

For the non-dispersive non-attenuated waves, all the waves propagate at $c = 500$ m/s, with $\alpha = 0$. For the dispersive non-attenuated waves, the waves at 30, 40, 50, 60 and 70 kHz are with phase speeds at $c = 300$, 400, 500, 600 and 700 m/s respectively and $\alpha = 0$ also holds. For the non-dispersive attenuated waves, $c = 500$ m/s and $\alpha = 2 \times 10^4$ are for all frequencies.

For all the datasets with derivatives based on both FD and PS, minimizing Err(j) in (2.15) gives $a_3 = 1$, which is for $\mathbf{u}_{tt}$. For non-attenuated waves, T $= 2$ is selected by (2.14) with a_9 and a_{11} non-zero; for attenuated wave, T $= 3$ is chosen with the non-zero entries at a_2, a_9 and a_{11}. The results are detailed in Table.2.1, with all entries in $\mathbf{a}$ being 0 except a_2, a_3, a_9 and a_{11}. The wave equations are discovered since a_2, a_9 and a_{11} are the coefficients for $\mathbf{u}_t$, $\mathbf{u}_{xx}$ and $\mathbf{u}_{yy}$. The

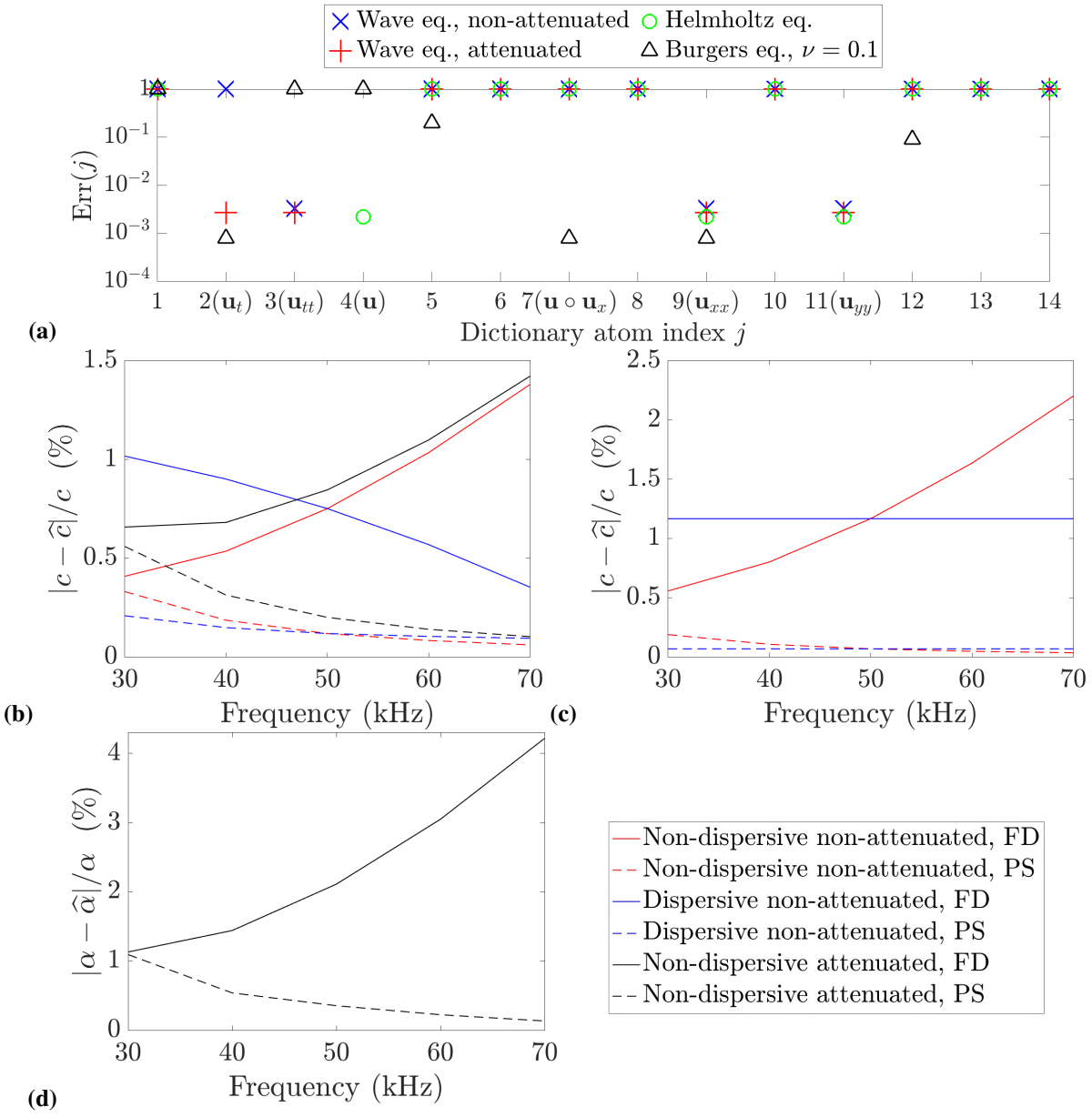

Figure 2.3. (a) Err(j) vs. atom index j in $\mathbf{\Phi}$, including: wave equations for attenuated and dispersive non-attenuated waves at 70kHz, Helmholtz equation for the dispersive wave at 70kHz, Burgers equation. (b–c) $\frac{|c-\widehat{c}|}{c}$ for wave equations and Helmholtz equations. (d) $\frac{|\alpha-\widehat{\alpha}|}{\alpha}$ of attenuated waves.

estimated speed $\widehat{c} = \sqrt{\frac{|a_9|+|a_{11}|}{2}}$. Fig.3(a) shows Err$(j)$ based on PS, and Fig.3(b), 3(d) where $\widehat{\alpha} = a_2$ suggest the method works well as the correctly chosen PDE terms are with errors less than 5% (majority $< 2.5\%$). For a given dataset, using the PS based dictionary always provides a smaller error than using the FD based dictionary. The relation between errors and frequencies in Fig.3(b),3(d) is explained as following.

(i) The spatial-temporal differentiation works as high-pass filtering in wavenumber-frequency domain. For PS which computes derivatives in wavenumber-frequency domain, an input signal of a higher frequency or wavenumber indicates a larger ratio between the derivative of the signal and the noise (from numerical differentiation), which benefits the identification. As the frequency increases, the wavenumber increases linearly for non-dispersive waves and is a constant in our dispersive waves (100 m^{-1}). The identified coefficients have smaller errors in both cases and the performance improvement is larger for the non-dispersive case.

(ii) The FD computes derivatives in the spatial-temporal domain. As the period or wavelength decreases, the identification suffers from insufficient sampling. For our non-dispersive cases, both the wavelength and the period decrease for higher frequencies, the insufficient sampling is significant and leads to increasing errors. For the dispersive case, only period decreases while the wavelength is constant for larger frequencies, the benefit described in (i) is significant and results in decreasing coefficients errors.

Comparing (i) and (ii), the PS is more robust to insufficient sampling. This is due to its computing the derivatives in wavenumber-frequency domain, implying an implicit trigonometric interpolation in spatial-temporal domain before numerical differentiation[34].

The proposed approach can also identify the PDE from a summation of its multiple solutions. We show this by experiments using the two unfiltered non-dispersive wavefields (attenuated and non-attenuated). Each of them is a summation of 5 solutions (one solution for one frequency) of one wave equation (2.2), with $c = 500$ m/s and $\alpha = 0$ or 2×10^4.

Using the dictionary constructed by PS, the identified PDE for the non-attenuated waves

Table 2.1. Results for synthetic wave equation recovery experiments, with '–' denoting the same values as in its upper entry. In last 4 columns for each dataset, the top value in each entry is the result based on FD and the bottom is based on PS.

Freq (kHz)	c (m/s)	α ($\times 10^4$)	a_2 ($\times 10^4$)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\widehat{c}$ (m/s)	c (m/s)	α ($\times 10^4$)	a_2 ($\times 10^4$)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\widehat{c}$ (m/s)	c (m/s)	α ($\times 10^4$)	a_2 ($\times 10^4$)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\widehat{c}$ (m/s)
Non-dispersive, non-attenuated waves							Dispersive, non-attenuated waves						Non-dispersive, attenuated waves					
30	500	0	FD: 0	2.52	2.52	502	300	0	FD: 0	0.92	0.92	303	500	2	FD: 2.02	2.53	2.53	503
			PS: 0	2.52	2.52	502			PS: 0	0.90	0.90	301			PS: 2.02	2.53	2.53	503
40	–	–	–	2.53	2.53	503	400	–	–	1.63	1.63	404	–	–	FD: 2.03	2.53	2.53	503
				2.51	2.51	501				1.60	1.60	401			PS: 2.01	2.52	2.52	502
50	–	–	–	2.54	2.54	504	500	–	–	2.54	2.54	504	–	–	FD: 2.04	2.54	2.54	504
				2.51	2.51	501				2.51	2.51	501			PS: 2.01	2.51	2.51	501
60	–	–	–	2.55	2.55	505	600	–	–	3.64	3.64	603	–	–	FD: 2.06	2.56	2.56	505
				2.50	2.50	500				3.61	3.61	601			PS: 2.00	2.51	2.51	501
70	–	–	–	2.57	2.57	507	700	–	–	4.93	4.93	702	–	–	FD: 2.08	2.57	2.57	507
				2.50	2.50	500				4.91	4.91	701			PS: 2.00	2.51	2.51	501

is

$$U_{tt} - 2.50 \times 10^5 (U_{xx} + U_{yy}) = 0 \tag{2.16}$$

and for the attenuated waves is

$$U_{tt} + 2.00 \times 10^4 U_t - 2.50 \times 10^5 (U_{xx} + U_{yy}) = 0 \, . \tag{2.17}$$

Thus the recovered $\widehat{c} \approx 500$ m/s for both waves, $\widehat{\alpha} \approx 2 \times 10^4$ for the attenuating wave. For the dictionary based on FD, they are

$$\begin{aligned} &U_{tt} - 2.56 \times 10^5 (U_{xx} + U_{yy}) = 0 \\ &U_{tt} + 2.06 \times 10^4 U_t - 2.56 \times 10^5 (U_{xx} + U_{yy}) = 0 \, , \end{aligned} \tag{2.18}$$

for the non-attenuated and attenuated waves respectively. So the recovered $\widehat{c} \approx 506$ m/s for both waves, $\widehat{\alpha} \approx 2 \times 10^4$ for the attenuating wave.

2.3.2 Helmholtz equation

Fourier transforming the $\mathbf{U}$ governed by wave equations (2.2) with $\alpha = 0$ at each spatial location over time, we obtain the frequency components $\mathscr{U} \in \mathbb{C}^{N_x \times N_y \times N_f}$, $N_f = N_t$. Data in each spatial frame of $\mathscr{U}$ satisfy Helmholtz equation (2.3). We thus use frequency components of the previous non-attenuated waves as datasets for Helmholtz equations identification.

The 1st spatial frame of $\mathscr{U}$ is for DC, and $\Delta f = \frac{1/\Delta t}{N_f} = 10$ kHz between neighboring frames. Thus we have 10 datasets used for Helmholtz equation identification, with each dataset being one frame among the 4–8th frames in two spectra $\mathscr{U}$, which are for non-attenuated (i) non-dispersive and (ii) dispersive waves. Fig.1(b) shows the 8th frame for both $\mathscr{U}$. The ROI on each frame excludes region $\{(i_x, i_y)|46 \leq i_x \leq 55; 46 \leq i_y \leq 55\}$ near the source. Using the same symbol $\mathbf{U}$ to denote $\mathscr{U}$ and constructing $\mathbf{\Phi}$ as (2.9), since each equation is identified from a 2D frame in frequency domain, $\mathbf{u}_t$ and $\mathbf{u}_{tt}$ are not included.

Minimizing $\mathrm{Err}(j)$, $a_4 = 1$ is selected and thus $\mathbf{u}$ is chosen. From (2.14), $\mathrm{T} = 2$ is chosen for all experiments with a_9 and a_{11} non-zero, the coefficients for $\mathbf{u}_{xx}$ and $\mathbf{u}_{yy}$. The results are detailed in Table.2.2, with all entries in $\mathbf{a}$ being 0 except a_4, a_9 and a_{11}. The PDE (2.3) is scaled by $1/\omega^2$ with $a_4 = 1$, thus the estimated speed $\widehat{c} = \omega\sqrt{\frac{|a_9|+|a_{11}|}{2}}$, see Fig.3(c). $\mathrm{Err}(j)$ for 70 kHz component from dispersive waves is in Fig.3(a). The relation between errors and frequencies in Fig.3(c) is explained as following.

(i) For Helmholtz equations, no temporal derivatives are involved, and thus the benefit for higher frequencies due to the differentiation's working as a high-pass filter disappears. But for larger wavenumbers this benefit from differentiation still exists. This explains the decreasing errors for non-dispersive waves and the constant error for dispersive waves (whose wavenumber is a constant 100 m^{-1}) for higher frequencies using PS based dictionaries.

(ii) For the FD cases, the wavenumber (thus wavelength) is constant for all frequencies in the dispersive waves. So the same sampling sufficiency leads to a constant error. The non-dispersive waves have shorter wavelengths for higher frequencies, and the insufficient sampling issue outweighs the benefit from larger wavenumbers, causing an increasing error. Comparing it with the errors for dispersive waves (FD) in Fig.3(b), decreasing samples spatially is more influential than decreasing samples temporally. This is because for our data the temporal sampling is more sufficient, e.g., for the 50 kHz wave propagating at 500 m/s, there are 10 spatial samples in one wavelength and 20 temporal samples in one period.

Table 2.2. Results for synthetic Helmholtz equation recovery experiments. For entries in the last 3 columns of each dataset, the top value is the result based on FD and the bottom is based on PS. $\omega = 2\pi \times$ Freq.

Freq (kHz)	True c (m/s)	$\omega^2 a_9$ ($\times 10^5$)	$\omega^2 a_{11}$ ($\times 10^5$)	$\widehat{c}$ (m/s)	True c (m/s)	$\omega^2 a_9$ ($\times 10^5$)	$\omega^2 a_{11}$ ($\times 10^5$)	$\widehat{c}$ (m/s)
Non-dispersive wave case					Dispersive wave case			
30	500	FD:2.53 PS: 2.51	2.53 2.51	503 501	300	FD:0.92 PS: 0.90	0.92 0.90	304 300
40	–	FD:2.54 PS: 2.51	2.54 2.51	504 501	400	FD:1.64 PS: 1.60	1.64 1.60	405 400
50	–	FD:2.56 PS: 2.50	2.56 2.50	506 500	500	FD:2.56 PS: 2.50	2.56 2.50	506 500
60	–	FD:2.58 PS: 2.50	2.58 2.50	508 500	600	FD:3.68 PS: 3.60	3.68 3.60	607 600
70	–	FD:2.61 PS: 2.50	2.61 2.50	511 500	700	FD:5.02 PS: 4.91	5.02 4.91	708 700

2.3.3 Burgers equation

The Burgers equation (2.4) with viscosity ν can describe shock wave formation. We consider 1D Burgers equation on the field $\mathbf{U} \in \mathbb{R}^{500\times 500}$ with $\Delta x = 0.04$ m and $\Delta t = 0.01$ s. Three fields with a same initial condition (a perturbation shaped as a Gaussian distribution PDF) governed by (2.4) with $\nu = 0.025, 0.05$ and 0.1 are generated by 4th order Runge-Kutta method [35]. Fig.2 shows the common initial state and the waveforms at $t = 5$ s for various ν. For a larger ν, the shock at $t = 5$ s becomes smoother due to the increased diffusion. No source is included.

In the 1D case, terms in (2.9) involving derivatives along y are meaningless and thus excluded. With derivatives based on both FD and PS, $a_7 = 1$ is selected by minimizing (2.15) for all experiments, and thus $\mathbf{u} \circ \mathbf{u}_x$ is identified, see Fig.2.3a. T = 2 is found by minimizing (2.14), with non-zero entries $a_2,\ a_9$ being $1.01, 1.00, 1.00$ and $-0.024, -0.05, -0.10$ for the 3 cases of ν based on PS. For FD, recovered a_2, a_9 are the same except that $a_2 = 1.00, a_9 = -0.025$ when $\nu = 0.025$. It works better because the spatial derivatives used for implementing Runge-Kutta method are FD based.

If we use OMP instead of the TLS for the cross-validation and the final coefficients

recovery, it will provide incorrect PDEs. Because under the assumption $\mathbf{u}_{xx}$ is active ($a_9 = 1$), the $\mathbf{u}_{tt}$ is its most correlated term and will be selected in the 1st iteration by OMP. The $\mathbf{u}_{tt}$ is correlated with all other terms in the dictionary, and thus incorporating $\mathbf{u}_{tt}$ into the set of the terms to fit $\mathbf{u}_{xx}$ will introduce the components of some irrelevant terms. This makes $\mathrm{L}_9(\mathrm{T})$ become plateaued and thus $\psi_9(\mathrm{T})$ be minimized at a sparsity larger than the correct value. With a larger $\widehat{\mathrm{T}}_9$ selected, the $\mathrm{Err}(j)$ in (2.15) for $j = 9$ is smaller than for the other correct assumptions ($j = 2$ or 7) which has the correct $\widehat{\mathrm{T}}_j$, because of more involved terms. For the Burgers equations, the $\widehat{\mathrm{T}}_9 = 7, 7, 6$ using PS and $3, 6, 6$ using FD for $\nu = 0.025, 0.05, 0.1$ (Fig.2.4 shows the $\mathrm{L}_9(\mathrm{T})$, $\psi_9(\mathrm{T})$ for $\nu = 0.025$ with PS), and $j = 9$ always minimizes (2.15). After $a_9 = 1$ is assumed, since the $\widehat{\mathrm{T}}_9$ is incorrect (which is supposed to be 2, see (2.4)), the Burgers equations can not be identified.

Fundamentally, the OMP's only considering the most correlated atom without utilizing the relationship among all atoms in the dictionary in its 1st iteration leads to its failure. Unlike OMP, for $j = 9$, the TLS selects T atoms contributing most to the orthogonal projection of $\mathbf{u}_{xx}$ in the subspace spanned by all the $D-1$ terms in $\bar{\mathbf{\Phi}}^k_{-9\_\mathrm{tr}}$. This orthogonal projection is a vector sum and is influenced by the relationship (correlation) among all vectors in the dictionary. A linear combination of $\mathbf{u}_t$ and $\mathbf{u} \circ \mathbf{u}_x$ forms the majority of the projection, and thus they are identified when $\mathrm{T} = 2$. With all the true active terms selected, the other non-zero entries in $\bar{\mathbf{a}}^k_{-9\_\mathrm{tr}}$ are found by fitting small noise in the training data, so they are of small magnitudes and work poor in the validation data, causing $\mathrm{L}_9(\mathrm{T})$ to plateau and $\widehat{\mathrm{T}}_9 = 2$ being selected (see an example as a comparison to OMP using the same dataset in Fig.2.4). So (2.15) is not minimized at $j = 9$ by involved irrelevant terms. In fact, $\mathrm{Err}(2) \approx \mathrm{Err}(7) \approx \mathrm{Err}(9)$ and either of the assumption for $j = 2, 7$ or 9 leads to the correct PDE.

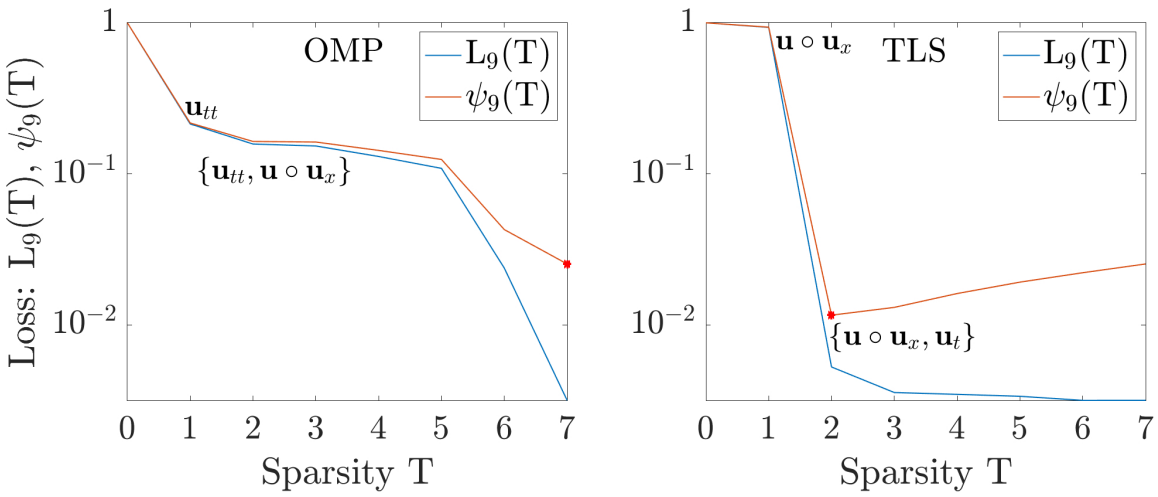

Figure 2.4. Atoms selection via OMP and TLS for Burgers equation identification, assuming $\mathbf{u}_{xx}$ active (i.e., $j = 9$). For a given method and a sparsity T, the selected atoms are the same for every fold in cross-validation as indicated around $\psi_9(\mathrm{T})$ for $\mathrm{T} = 1$ or 2. The red asterisk shows the minimizer of $\psi_9(\mathrm{T})$.

2.4 Application to real video

Our approach is demonstrated on a video of aluminum plate vibrations [36], see Fig.4. One period of this video considered is $\mathbf{U} \in \mathbb{R}^{100\times100\times100}$ with Δx and Δy 1 mm and sampled at 300 kHz. Vibrations contained in $\mathbf{U}$ are impulse responses for a delta function in the past, thus no source is within the selected time.

Since aluminum plate waves are dispersive [37], the signal is band-pass filtered to isolate wave equations for each frequency. We explore frequency bins centered from 20 to 70 kHz, with 5 kHz steps. Each bin has 1 kHz width. As in synthetic experiments, $\mathbf{u}$ is not considered for these filtered narrow band signals. $a_3 = 1$ minimizes (2.15) for all frequencies, with some shown in Fig.5, and $\mathrm{T} = 2$ with non-zero entries at a_9, a_{11} is always chosen by (2.14), as detailed in Table.2.3. Wave equations on the plate are discovered, with $\widehat{c} = \sqrt{\frac{|a_9|+|a_{11}|}{2}}$ shown in Fig.6.

The proposed approach is compared to a classic phase speed estimation based on Fourier transform [38, 39]. The method estimates phase speeds $\widehat{c}_{cl}$ by finding the primary wavenumber $\widehat{k}$ for each frequency f and $\widehat{c}_{cl} = f/\widehat{k}$. Due to the isotropic property of the wave propagation on the plate, from the wavenumber-frequency spectrum $\mathbf{K}$ of $\mathbf{U}$, for frequency f_0,

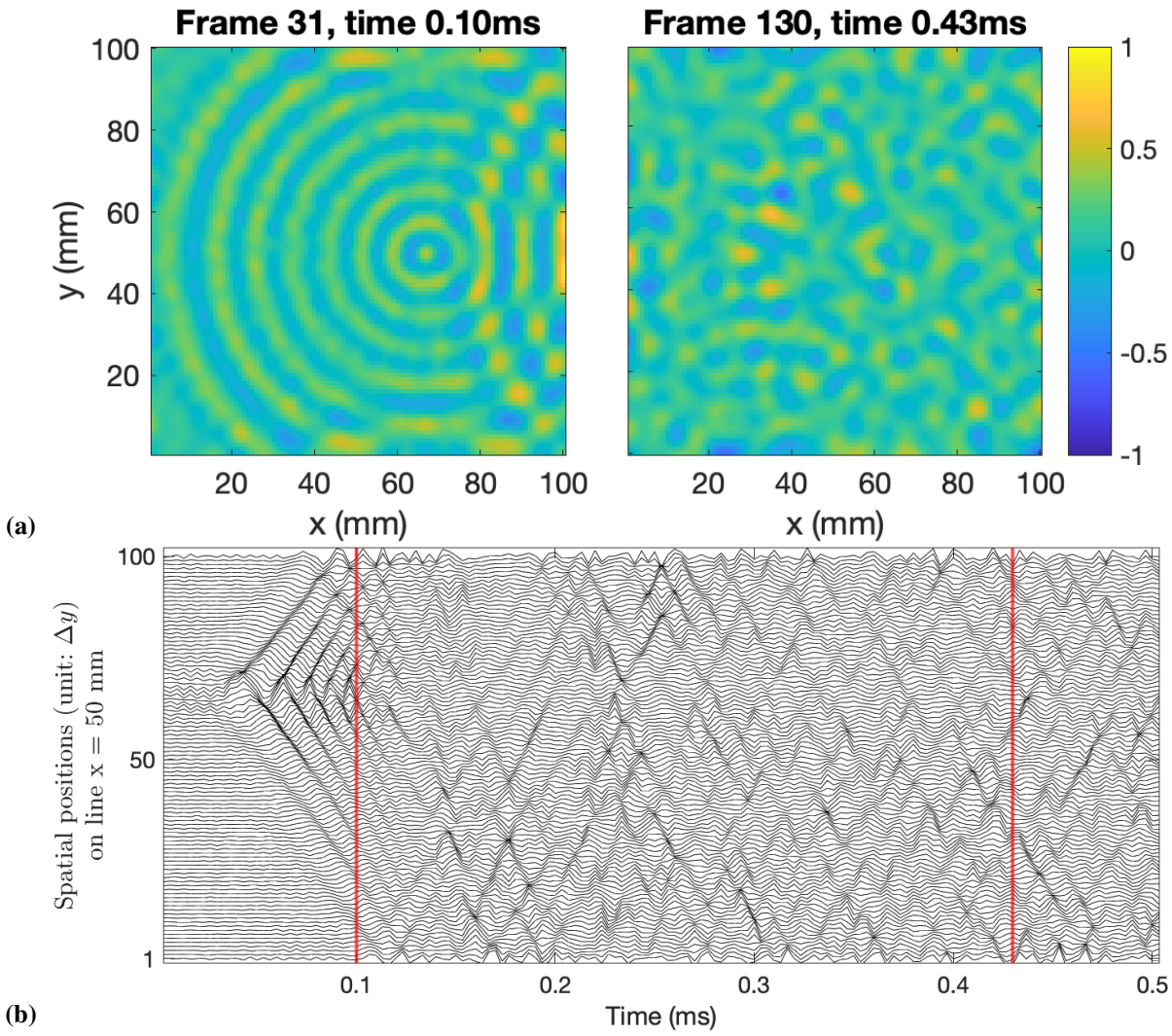

Figure 2.5. The vibrating plate: (a) the first and last selected frames, with magnitudes normalized; (b) the traces for locations at $x = 50$ mm, the selected time period is between the red lines.

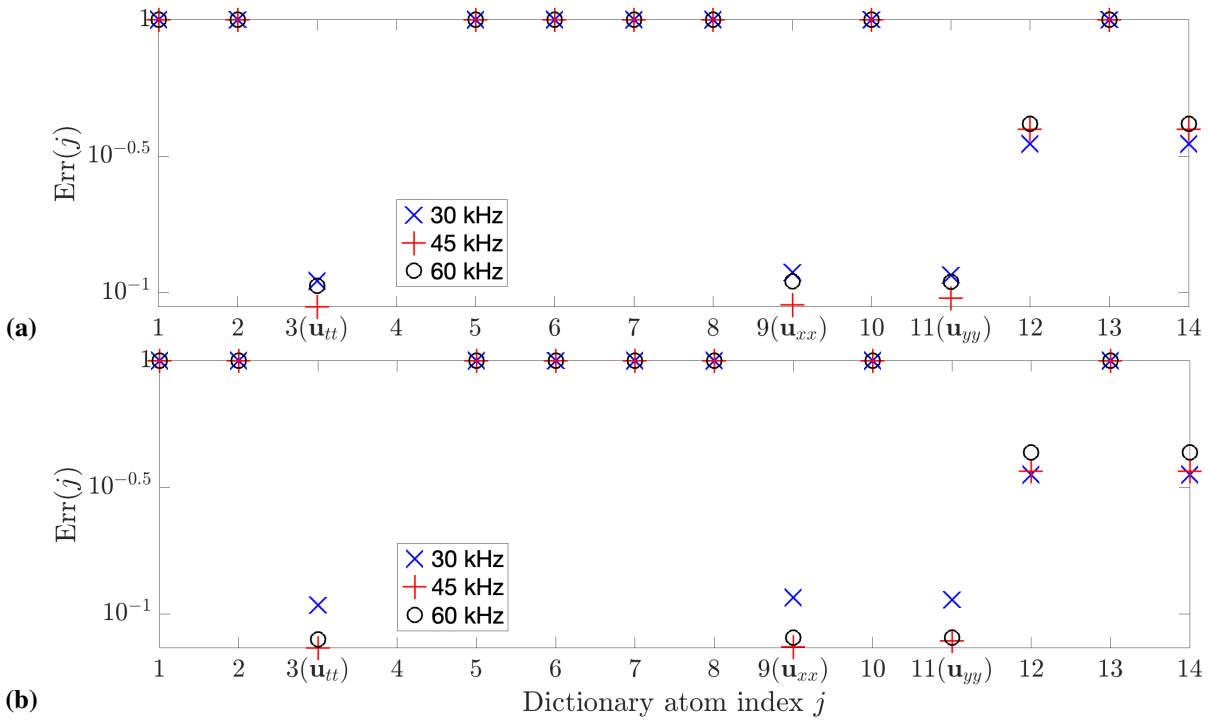

Figure 2.6. Err(j) vs. atom index j in $\mathbf{\Phi}$ for 3 frequency bins of vibrating plate signal, with derivatives based on (a) FD, and (b) PS.

$\widehat{k} = \arg\max_k \sum_{k_x=0}^{k} |\mathbf{K}(k_x, \sqrt{k^2 - k_x^2}, f_0)|$, which finds the radius of a quarter ring with the maximal power of $\mathbf{K}$.

The underestimation by FD for high frequencies (see Fig. 6) arises from insufficient sampling along time. The PS does trigonometric polynomial interpolation [34], and $\partial_t^{(p)}\mathbf{u}(t_j)$ (which is calculated in the same way as $\partial_x^{(p)}\mathbf{u}(x_j)$ in (2.10)) is evaluated at t_j (the point) sampled from the interpolated signal. Thus the high frequencies producing derivatives with large magnitudes are preserved. But FD evaluates $\partial_t^{(p)}\mathbf{u}(t_j)$ based on slopes of the line segments connecting $\mathbf{u}(t_j)$ with $\mathbf{u}(t_{j-1})$ and $\mathbf{u}(t_{j+1})$ respectively. When Δt is not sufficiently small, these slopes can be far from the slope of the tangent line passing $\mathbf{u}(t_j)$, causing significant bias.

2.5 Conclusion

We formulated a data-driven approach to extract PDEs without assumed terms and tested it on synthetic data and a real vibrating aluminum plate video. A dictionary containing

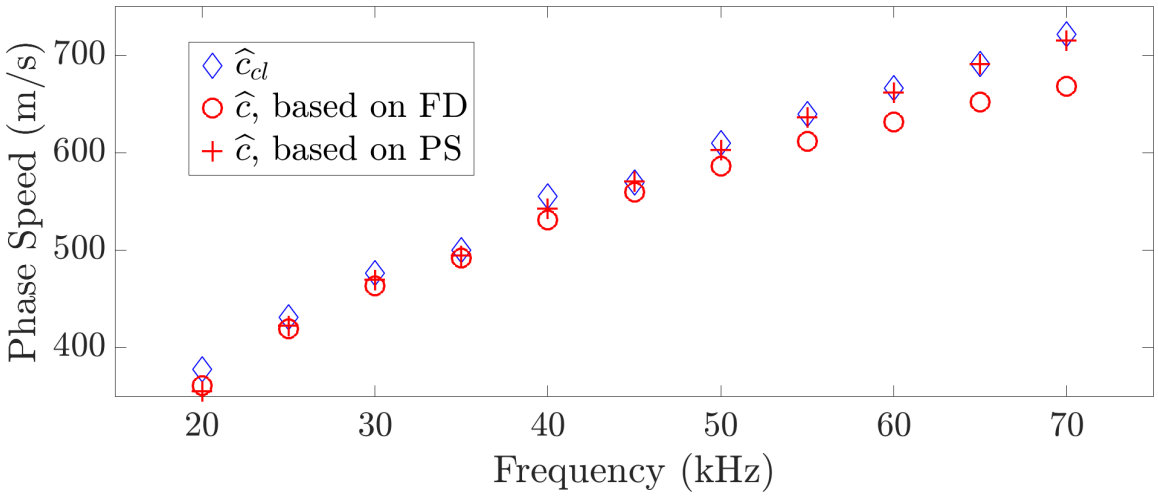

Figure 2.7. Phase speeds recovered from identified PDEs and wavenumber extraction.

hypothetical PDE terms is built and correct terms are extracted by sparse modeling using cross validation with a sparsity penalty.

2.6 Supplemental materials

2.6.1 Wavenumber determined by wave equations with attenuation

The term $c^2\nabla^2 u$ in wave equation indicates an isotropic propagation nature of the waves with a phase speed c. For the part of wave which propagates along direction $\mathbf{r}$ in a circular wave, the simplified equation

$$u_{tt} = -\alpha u_t + c^2 u_{rr} + f \,. \tag{2.19}$$

can depict its dynamics without the effect of decay due to the increasing area encompassed by the wave front. For $f = 0$ and a wave at frequency ω having propagated a distance r along $\mathbf{r}$, the complex solution

$$u_c = e^{-i(kr-\omega t)} \tag{2.20}$$

Table 2.3. Results for wave equation recovery on a real vibrating plate. In the column $-a_9$, $-a_{11}$ and $\widehat{c}$, the top value in each entry is the result based on FD and the bottom is based on PS.

Freq (kHz)	$-a_9$ ($\times 10^5$)	$-a_{11}$ ($\times 10^5$)	$\widehat{c}$ (m/s)	$\widehat{c}_{cl}$ (m/s)
20	FD: 1.25 PS: 1.21	1.36 1.32	361 355	377
25	FD: 1.84 PS: 1.92	1.68 1.65	419 422	431
30	FD: 2.16 PS: 2.22	2.13 2.18	463 469	476
35	FD: 2.40 PS: 2.40	2.44 2.49	492 494	500
40	FD: 2.81 PS: 2.88	2.83 3.01	531 543	556
45	FD: 3.13 PS: 3.32	3.13 3.19	559 571	570
50	FD: 3.46 PS: 3.69	3.43 3.59	587 603	610
55	FD: 3.73 PS: 4.04	3.75 4.06	612 637	640
60	FD: 4.04 PS: 4.40	3.94 4.37	632 662	667
65	FD: 4.23 PS: 4.74	4.27 4.81	652 691	691
70	FD: 4.45 PS: 5.06	4.48 5.17	668 715	722

can satisfy (2.19). Plugging (2.20) into (2.19) yields:

$$\begin{aligned} -\omega^2 u_c + i\alpha\omega u_c + c^2 k^2 u_c = 0 \\ -\omega^2 + i\alpha\omega + c^2 k^2 = 0 \,. \end{aligned} \tag{2.21}$$

So

$$\begin{aligned} k^2 = \frac{1}{c^2}(\omega^2 - i\alpha\omega) = \frac{\omega^2}{c^2}(1 - i\frac{\alpha}{\omega}) \\ k = \frac{\omega}{c}\sqrt{1 - i\frac{\alpha}{\omega}} \approx \frac{\omega}{c}(1 - \frac{i\alpha}{2\omega}) \end{aligned} \tag{2.22}$$

We can rewrite $u_c(r,t)$ as

$$u_c(r,t) = a(r,t) + ib(r,t) \tag{2.23}$$

where

$$\begin{aligned} a(r,t) = \mathrm{Re}(u_c(r,t)) \in \mathbb{R} \\ b(r,t) = \mathrm{Im}(u_c(r,t)) \in \mathbb{R} \,. \end{aligned} \tag{2.24}$$

In the following equations we abbreviate $a(r,t), b(r,t)$ as a, b respectively.

Plug (2.23) into (2.19) (assume $f = 0$), we have

$$\frac{\partial^2(a+ib)}{\partial t^2} + \alpha\frac{\partial(a+ib)}{\partial t} - c^2\frac{\partial^2(a+ib)}{\partial r^2} = 0 \tag{2.25}$$

thus

$$(\frac{\partial^2 a}{\partial t^2} + \alpha\frac{\partial a}{\partial t} - c^2\frac{\partial^2 a}{\partial r^2}) + i(\frac{\partial^2 b}{\partial t^2} + \alpha\frac{\partial b}{\partial t} - c^2\frac{\partial^2 b}{\partial r^2}) = 0 \tag{2.26}$$

and thus

$$\frac{\partial^2 a}{\partial t^2} + \alpha\frac{\partial a}{\partial t} - c^2\frac{\partial^2 a}{\partial r^2} = 0,\ \frac{\partial^2 b}{\partial t^2} + \alpha\frac{\partial b}{\partial t} - c^2\frac{\partial^2 b}{\partial r^2} = 0 \tag{2.27}$$

So $a(r,t)$ and $b(r,t)$ are both solutions for (2.19).

Since the displacement field is real, we use $a(r,t)$ which is $\mathrm{Re}(u_c(r,t))$, where k is determined in (2.22).

Plug (2.22) into (2.20), $a(r,t)$ is

$$a(r,t) \approx \mathrm{Re}(e^{-i[\frac{\omega}{c}(1-\frac{i\alpha}{2\omega})r-\omega t]}) = e^{-\frac{\alpha r}{2c}}\cos(\frac{\omega}{c}r - \omega t)\ . \tag{2.28}$$

If $\frac{\alpha r}{2c}$ is small across the domain, the attenuation does not contribute much to the derivatives of $a(r,t)$, causing $\frac{\partial a}{\partial t} \approx -c\frac{\partial a}{\partial r}$.

2.6.2 Threshold least squares (TLS)

The TLS finds the coefficients $\mathbf{a} = [a_1 \dots a_D]^{\mathrm{T}} \in \mathbb{C}^D$ which selects T other columns in $\mathbf{\Phi} \in \mathbb{C}^{N\times D}$ to fit its jth column. The notations here may not refer to the same variables as in the main text, for example, when in the training stage of the K-fold cross-validation, we use the $K-1$ concatenated $\bar{\mathbf{\Phi}}^k$ defined in the text as the "$\mathbf{\Phi}$" here and thus the "N" is assigned as $\frac{K-1}{K}N$.

First we normalize each column of $\mathbf{\Phi}$ by its $\ell 2$ norm and denote the normalized dictionary as $\bar{\mathbf{\Phi}} \in \mathbb{C}^{N\times D}$. For a given j, use $\bar{\mathbf{\Phi}}_{-j} \in \mathbb{C}^{N\times(D-1)}$ to denote $\bar{\mathbf{\Phi}}$ dropping its jth column $\bar{\boldsymbol{\phi}}_j$, and similarly use $\mathbf{\Phi}_{-j} \in \mathbb{C}^{N\times(D-1)}$ to denote $\mathbf{\Phi}$ dropping its jth column $\boldsymbol{\phi}_j$. Correspondingly, we set $a_j = 1$ and store the other entries of $\mathbf{a}$ in $\mathbf{a}_{-j} \in \mathbb{C}^{D-1}$. The TLS is employed to compute $\mathbf{a}_{-j}$

Algorithm 2: Threshold least squares (TLS).

Input: $\boldsymbol{\Phi}_{-j} = [\boldsymbol{\phi}_1 \dots \boldsymbol{\phi}_{j-1}\ \boldsymbol{\phi}_{j+1} \dots \boldsymbol{\phi}_D] \in \mathbb{C}^{N\times(D-1)}$, $\boldsymbol{\phi}_j$, T

Output: $\mathbf{a}_{-j}$

$\mathbf{w}_{-j} = \{\|\boldsymbol{\phi}_d\|_2 | \forall d, d \neq j\} \in \mathbb{C}^{D-1}$ // Column norms of $\boldsymbol{\Phi}$ except its jth column

$\bar{\boldsymbol{\Phi}}_{-j} = \boldsymbol{\Phi}_{-j}\text{diag}^{-1}(\mathbf{w}_{-j})$ // The normalized $\boldsymbol{\Phi}_{-j}$ with its dth column denoted by $\bar{\boldsymbol{\phi}}_d$

$\bar{\mathbf{a}}^{\text{ls}}_{-j} = \bar{\boldsymbol{\Phi}}^{\dagger}_{-j}\boldsymbol{\phi}_j$ // Least squares

$\Omega = \{\Omega(1) \dots \Omega(\text{T})\} = \{\text{Indices of entries in } \bar{\mathbf{a}}^{\text{ls}}_{-j} \text{ with T maximal magnitudes}\}$ // Thresholding

$\bar{\boldsymbol{\Phi}}^{\text{th}}_{-j} = \{\bar{\boldsymbol{\phi}}_d | \forall d, d \in \Omega\} \in \mathbb{C}^{N\times\text{T}}$ // Columns kept

$\bar{\mathbf{a}}^{\text{th}}_{-j} = [\bar{a}^{\text{th}}_1 \dots \bar{a}^{\text{th}}_{\text{T}}]^{\text{T}} = \bar{\boldsymbol{\Phi}}^{\text{th}\dagger}_{-j}\boldsymbol{\phi}_j$ // Least squares

$\bar{\mathbf{a}}_{-j} = [\bar{a}_1 \dots \bar{a}_{D-1}]^{\text{T}} = \mathbf{0}$ // Initialize for $\mathbf{a}_{-j}$

$\bar{a}_{\Omega(i)} = -\bar{a}^{\text{th}}_i,\ \forall i = 1 \dots \text{T}$ // Assign non-zero values to selected entries

$\widehat{\mathbf{a}}_{-j} = \bar{\mathbf{a}}_{-j} \oslash \mathbf{w}_{-j}$ // Scaling by Hadamard division

such that $\boldsymbol{\phi}_j \approx -\boldsymbol{\Phi}_{-j}\mathbf{a}_{-j}$ (since $\|\boldsymbol{\Phi}\mathbf{a}\|_2 \approx 0$) subject to $\|\mathbf{a}_{-j}\|_0 = \text{T}$.

Algo.2 outlines the TLS, with "†" for pseudo-inverse, "diag" for constructing a diagonal matrix from a vector and "$\oslash$" for element-wise division (Hadamard division). Within Algo.2, the $\Omega = \{\Omega(1) \dots \Omega(\text{T})\}$ denotes the set of T selected indices, e.g., if $\text{T} = 3$ and the 3 entries with maximal magnitudes in $\bar{\mathbf{a}}^{\text{ls}}_{-j}$ have indices $2, 5, 7$, then $\Omega = \{\Omega(1), \Omega(2), \Omega(3)\} = \{2, 5, 7\}$.

2.6.3 Comparison with SINDy

In the previous data-driven PDE identification method SINDy[1], the authors used sequential threshold ridge regression (STRidge) to select active PDE terms in a normalized dictionary $\bar{\boldsymbol{\Phi}}_{-j} \in \mathbb{C}^{N\times(D-1)}$ (all columns have unit $\ell 2$ norm) to fit a given PDE term $\boldsymbol{\phi}_j$.

The STRidge is a recursive method, where a ridge regression is implemented and the columns corresponding to small coefficients are dropped in each recursion, as illustrated in Algo.3. After the active terms are selected, the final coefficients are acquired by least squares regressing the assumed term $\boldsymbol{\phi}_j$ onto the identified terms in the original dictionary (without normalization) $\boldsymbol{\Phi}_{-j}$.

Algorithm 3: Sequential threshold ridge regression (STRidge)[1]

Input: $\bar{\boldsymbol{\Phi}}_{-j}$, $\boldsymbol{\phi}_j$, λ, τ, iters
Output: $\bar{\mathbf{a}}$ whose ith entry denoted by $\bar{a}_i$
$\widehat{\bar{\mathbf{a}}} = \arg\min_{\bar{\mathbf{a}}} \|\bar{\boldsymbol{\Phi}}_{-j}\bar{\mathbf{a}} - \boldsymbol{\phi}_j\|_2^2 + \lambda\|\bar{\mathbf{a}}\|_2^2$
bigcoeffs $= \{i : |\widehat{\bar{a}}_i| \geq \tau\}$
$\widehat{\bar{\mathbf{a}}}[\sim \text{bigcoeffs}] = 0$ `// Threshold`
$\widehat{\bar{\mathbf{a}}}[\text{bigcoeffs}] = \text{STRidge}(\bar{\boldsymbol{\Phi}}_{-j}[:, \text{bigcoeffs}],$
$\boldsymbol{\phi}_j, \lambda, \tau, \text{iters} - 1)$ `// recursive call`

If the correct PDE term $\boldsymbol{\phi}_j$ is assumed, given proper λ and τ, the STRidge can work on our dataset. For example, the STRidge can recover the correct terms U_{xx} and U_{yy} for all frequencies in the Helmholtz equation dataset for dispersive waves given the correct assumption that U is an active term and $\lambda = 1$, $\tau = 0.1$.

If the incorrect assumed term is chosen, the SINDy can not recognize it and will return incorrect PDE. For the same Helmholtz equation dataset, if the assumed term is U_x, then only U_{xx} is identified to be active in the dictionary.

2.7 Acknowledgments

The text of Chapter 2, in full, is a reprint of the material as it appears in R. Liu, M. Bianco, P. Gerstoft, "Automated Partial Differential Equation Identification", *The Journal of the Acoustical Society of America*, 150(4), 2364-2374, 2021. The dissertation author was the primary researcher and author of Chapter 2. The coauthors listed in this publication directed and supervised the research.

Chapter 3

Recovery of Spatially Varying Acoustical Properties via Automated PDE Identification

3.1 Introduction

Natural phenomena are in general caused by partial differential equations (PDEs) with the PDE-coefficients derived from medium properties. For example, the observed waves propagating on a plate are governed by wave equations with the PDE-coefficients determined by phase speeds and attenuation factors, which are further decided by the elastic properties and density of the plate's material.

The focus of this paper is solving the inverse problem, i.e., we invert for the active PDE terms from the observations, determine the coefficients of each active PDE term, and then use the coefficients to recover physical properties. We highlight the use case of the proposed method in recovering various properties for the medium of propagating waves in acoustical scenarios, and thus the relative PDEs like the wave equation and Burgers' equation are used as examples.

This area has been an active focus of applied mathematics research [1, 15, 16, 18, 12, 11, 2, 7, 14, 3] with a few applications [26, 25, 27]. Considering the abundance of sensor-collected measurements and the wide range of use cases of material properties recovery (e.g., materials diagnostics, fatigue detection), this inversion technique would be broadly applicable.

Superior to previous data driven PDE identification approaches [1, 15, 16, 18, 12, 11, 2, 7, 14, 3, 26, 25, 27], our method can recognize spatially dependent PDEs, and thus recover 1D or 2D maps for the spatial variations of material properties from measurements of the phenomena. Unlike classic spatially dependent parameter estimation or tomography methods which require the PDE form known *a priori* and employ the domain knowledge only pertinent to such a particular PDE [24, 40, 41, 42, 43, 44, 45], the proposed method can identify multiple kinds of unknown PDEs using a same formulation with no PDE-dependent knowledge, and thus is widely applicable and can recover various properties in more scenarios with fewer assumptions.

We start with the spatially 1D case in the theory and include 2D examples for experiments. Consider a physical system $U(x,t)$ that describes the spatio-temporal dynamics. Within the system $U(x,t)$, suppose we have N_x evenly spatially distributed sensors collecting measurements (e.g., displacement, pressure, etc.) at M_t evenly separated time steps and thus we obtain measurements $\mathbf{U} \in \mathbb{R}^{N_x \times M_t}$ at discrete spatial-temporal coordinates of $U(x,t)$. The system properties are recovered by identifying its governing PDE

$$\mathsf{N}[U] = 0 \tag{3.1}$$

from the measurements. Here $\mathsf{N}[U]$ is a linear combination of PDE terms involving partial derivatives of U, e.g., $\mathsf{N}[U]$ describing the 1D wave propagation at speed c with the attenuation factor α is

$$\mathsf{N}[U] = U_{tt} + \alpha U_t - c^2 U_{xx} \ , \tag{3.2}$$

where U_t, U_{tt} are the 1st and 2nd-order temporal derivative and U_{xx} the 2nd-order spatial derivative of U. Recently, there are many developments focusing on identifying PDEs directly from observed data [1, 15, 16, 18, 12, 11, 2, 7, 14, 3, 26, 25, 27]. However, they have two limitations: (i) the need for prior knowledge of the active PDE terms; (ii) the inability to recover spatially-dependent parameters.

Regarding (i): Most previous developments require one or more active PDE-terms to be known *a priori* (e.g., the 1st order time derivative U_t [1, 15, 16, 18, 12, 11], the 2nd order time derivative U_{tt} [27], one term with predefined order [2], or multiple PDE terms [7]). They then derive other contributing PDE terms and their coefficients, and thus only parts of the PDE are inferred from data. This is problematic when the knowledge of which term should be assumed active is uncertain or unknown, e.g., to identify the governing PDE for a wave which may either be an inviscid Burgers' equation ($U_t + UU_x = 0$) or a non-attenuating wave equation ($U_{tt} - c^2\nabla^2 U = 0$) with no PDE terms in common, one must specify the correct active term from more prior information. The few methods that do not require active term assumption use a sparse Bayesian learning (SBL) based approach [14] or a cross-validation (CV) based method [3]. They both iteratively assume one active term from a library of terms, identify the PDE for each assumption using SBL [14] or sparsity penalized CV [3], and finally select the best assumption by comparing the posterior confidence [14] or the minimal fitting error [3]. They are time-consuming as the identification process is repeated for every assumption.

Regarding (ii): In reality the PDEs governing the observed system can have spatially-dependent coefficients, e.g., the coefficient for U_{xx} in (3.2) can vary across space which indicates the phase speed c is spatially-dependent. The spatially-dependent coefficients is due to the spatial variation of the materials, and thus the recovery of spatially-dependent PDEs can unveil the spatial properties of the underlying materials. The above methods, however, can only identify PDEs that are constant across space. The current spatially dependent coefficients recovery schemes are limited to a few specific PDEs [40, 41] and can not be used for PDE identification since they require the kind of PDE to be known. No methods can identify unknown PDEs which are potentially spatially-dependent.

Suppose there is no information about the spatial variation of the PDEs available and thus we must identify the PDE for every location. The challenges in tackling this task include: (a) fewer measurements available for one location comparing to the whole field; (b) longer CPU time because the process is repeated for all locations. Thus, a viable method should be robust in

selecting the correct PDE from limited measurements and also be computationally efficient.

To address these limitations, we propose an ℓ_1-norm minimization based data-driven method that identifies unknown PDEs for every spatial location without any assumption of the assumed active PDE term. No information about the spatial variation of the PDEs is assumed. An auxiliary vector is introduced to robustify the identification from limited measurements and enable the recovery of all active PDE terms without iterative assumptions. The method is applied pixel-wise and recovers the spatial variation of the PDEs to the highest resolution. For noisy measurements, we extend the integration transformation approach [12] to our spatially-dependent PDE identification scheme to make it more robust against noise.

This paper is extended from the work [4] to integrate the noise resistance technique and perform extensive experimental validations, including the ones using data collected from real physical settings. It is organized as follows: Sec. 3.2.1 presents the spatially-dependent PDE identification method, and Sec. 3.2.2 briefly introduces the integration transformation approach to alleviate the impact of noise. Sec. 3.3 shows synthetic experimental results, including an experiment for noise-robustness in Sec. 3.3.3. The experiments for a real vibrating aluminum plate using both clean and noisy measurements are in Sec. 3.4. In Sec. 3.5 we emphasize the efficiency of the proposed approach by comparing the required CPU time to identify the PDEs for 3 datasets using the proposed method, the methods in Refs. [14, 3] and exhaustive search. Finally, we conclude our work in Sec. 3.6.

Notation: For the measurements $\mathbf{U} \in \mathbb{R}^{N_x \times M_t}$, $\mathbf{U}(i_x, i_t)$ is $\mathbf{U}$ sampled at the coordinate $(i_x \Delta x, i_t \Delta t)$, where $0 \leq i_x \leq N_x - 1$, $0 \leq i_t \leq M_t - 1$, and $\Delta x,\ \Delta t$ are sampling intervals. Sets I_x, I_t contain N spatial and M temporal coordinates within the region of interest (ROI). Use $n \in [1, N]$ as the index of the elements in I_x. The temporal coordinates in I_t are indexed by $m \in [1, M]$. For any matrix $\mathbf{A}$ other than $\mathbf{U}$, the entry at its ith row and jth column where $i \geq 1,\ j \geq 1$ is denoted by $\mathbf{A}(i, j)$. $\mathbf{A}(i, :)$ denotes the ith row, and $\mathbf{A}(:, j)$ the jth column. The subscript/superscript of a matrix denotes the properties of the whole matrix, **not** its entries. E.g., $\mathbf{A}_n$ can indicate that all entries of $\mathbf{A}$ are computed from the measurements at location n, and $\mathbf{A}_n(i, j)$ is for the entry at

ith row and jth column of $\mathbf{A}_n$. For vector $\mathbf{a}$, its ith entry ($i \geq 1$) is denoted by either $\mathbf{a}(i)$ or a_i. The variable superscripted by $\widehat{}$ denotes its estimation.

3.2 Theory

In this part, we first propose the method to efficiently identify spatially-dependent PDEs in Sec. 3.2.1, and then introduce an integration transformation technique that increases the approach's robustness against measurement noise in Sec. 3.2.2.

3.2.1 PDE identification

We first formulate the PDE identification problem mathematically in Sec. 3.2.1, and then solve it in Sec. 3.2.1.

Problem formulation

We select N spatial locations with M time steps for each location from the measurements $\mathbf{U} \in \mathbb{R}^{N_x \times M_t}$ as the ROI. In this work, suppose we know the range of PDEs governing the dynamics in the ROI, to be specific, they may be the (attenuating) wave equation, the (viscous) Burgers' equation with a non-linear term UU_x (product of U and U_x), and the sine-Gordon equation with a non-derivative term $\sin(U)$ [46, 47]. They can model various fluid dynamics.

For the nth spatial location within the ROI, we use $\mathbf{u} \in \mathbb{R}^M$ to denote its measurements at all M time steps, i.e., from $\mathbf{U}(I_x(n), I_t(1))$ to $\mathbf{U}(I_x(n), I_t(M))$. We build a dictionary $\mathbf{\Phi}_n$ containing all $D = 6$ PDE terms potentially appearing in the PDEs mentioned above as:

$$\mathbf{\Phi}_n = \Big[\mathbf{u}_t, \quad \mathbf{u}_{tt}, \quad \mathbf{u} \circ \mathbf{u}_x, \quad \mathbf{u}_{xx}, \quad \mathbf{u}_{tx}, \quad \sin(\mathbf{u})\Big] \in \mathbb{R}^{M \times D}, \tag{3.3}$$

where each term is an M-length vector evaluated at all M time steps and the derivatives are computed numerically by finite difference [9], e.g., the mth entry of $\mathbf{u}_t$ is $[\mathbf{U}(I_x(n), I_t(m)+1) - \mathbf{U}(I_x(n), I_t(m)-1)]/(2\Delta t)$. The $\circ$ denotes element-wise production, e.g., the mth entry of $\mathbf{u} \circ \mathbf{u}_x$

is $\mathbf{U}(I_x(n),I_t(m)) \times \{[\mathbf{U}(I_x(n)+1,I_t(m))-\mathbf{U}(I_x(n)-1,I_t(m))]/(2\Delta x)\}$. Measurements outside the ROI boundaries should exist so that the spatial derivatives for $n=1$ or N and the temporal derivatives for $m=1$ or M can be computed by finite difference, which requires $N<N_x-1$ and $M<M_t-1$.

Initially, we can treat the problem as recovering the coefficient $\mathbf{a}_n = [\mathbf{a}_n(1) \ \dots \ \mathbf{a}_n(D)]^{\mathrm{T}} \in \mathbb{R}^D$ such that

$$\mathbf{\Phi}_n \mathbf{a}_n \approx \mathbf{0}, \ \|\mathbf{a}_n\|_1 > 0 \ , \tag{3.4}$$

where $\|\mathbf{a}_n\|_1 > 0$ is to avoid the trivial solution $\mathbf{a}_n = \mathbf{0}$, and the approximation is due to the assumption $M > D$ and the noise in the measurements or generated by numerical differentiation.

Exhaustive search can find the set of active PDE terms by iterating through all combinations of atoms in the dictionary and minimizing the fitting error with the constraint that not too many columns of $\mathbf{\Phi}_n$ are chosen (sparsity constraint). The cardinality of all possible sets ranges from 2 to D. For each hypothetical set, assume the coefficient for one term in $\mathbf{a}_n$ is 1, and fit the other terms by least squares regression.

Given an assumed active term with coefficient 1, the difference between $\mathbf{\Phi}_n\mathbf{a}_n$ and $\mathbf{0}$, i.e., $\|\mathbf{\Phi}_n\mathbf{a}_n\|_2$, is monotonically non-increasing as more columns of $\mathbf{\Phi}_n$ are chosen, and is decreasing for most cases. Using all columns of $\mathbf{\Phi}_n$ will minimize the difference, but it indicates that the PDE has all the D active terms, which is typically not true. Thus a sparsity constraint should be introduced to avoid selecting all terms in $\mathbf{\Phi}_n$ as active terms. One way to impose this constraint is to minimize an augmented loss function which is the sum of $\|\mathbf{\Phi}_n\mathbf{a}_n\|_2$ and a penalty proportional to $\|\mathbf{a}_n\|_0$[3].

This process requires combinatorial complexity because we can not directly select all the columns as discussed above, and need to explore all the possible sets of selected columns with cardinality ranging from 2 to D. For a dictionary with only a few columns like (3.3) this is feasible, but for larger dictionaries, the CPU time increases quickly. E.g., for (3.3), there are 57 kinds of sets in total, while for (3.24) with 9 terms, there are 502 cases. A demonstration of such

a transition is in Sec. 3.5.

Solving PDE coefficients by ℓ_1-norm minimization

Instead of exhaustive search, we introduce a normalization matrix $\mathbf{W}_n \in \mathbb{R}^{D\times D}$ which is diagonal with $\mathbf{W}_n(i,i) = \|\mathbf{\Phi}_n(:,i)\|_2$, and by finding the coefficients $\bar{\mathbf{a}}_n = [\bar{\mathbf{a}}_n(1)\dots\bar{\mathbf{a}}_n(D)]^{\mathrm{T}}$ that makes the columns in the normalized dictionary $\bar{\mathbf{\Phi}}_n = \mathbf{\Phi}_n\mathbf{W}_n^{-1}$ fit well under a fitting error tol_n

$$\|\bar{\mathbf{\Phi}}_n\bar{\mathbf{a}}_n\|_2^2 \le \mathrm{tol}_n \ \text{ s.t. } \ \|\bar{\mathbf{a}}_n\|_1 = 1\ , \tag{3.5}$$

we acquire $\mathbf{a}_n = \mathbf{W}_n^{-1}\bar{\mathbf{a}}_n$.

We use the normalized dictionary so that the variation of magnitudes for columns in $\mathbf{\Phi}_n$ does not affect the column selection, and thus the selection is only based on the dynamical characters (i.e., the variation of the entries within each column). In addition, we use $\|\bar{\mathbf{a}}_n\|_1 = 1$ in (3.5) instead of $\|\bar{\mathbf{a}}_n\|_1 > 0$ as in (3.4), otherwise the magnitudes of the non-zero elements in $\bar{\mathbf{a}}_n$ can be arbitrarily small to encourage $\mathbf{\Phi}_n\mathbf{W}_n^{-1}\bar{\mathbf{a}}_n \approx \mathbf{0}$.

The limitation of (3.5) is that $\|\bar{\mathbf{a}}_n\|_1 = 1$ is not a convex set and thus (3.5) is not solvable via efficient convex optimization tools. To make (3.5) amenable to convex optimization, we specify the positive/negative signs in $\|\bar{\mathbf{a}}_n\|_1$ and thus reduce $\|\bar{\mathbf{a}}_n\|_1 = 1$ to an affine constraint as detailed below.

We utilize physical information to reduce $\|\bar{\mathbf{a}}_n\|_1 = 1$ to $\sum_{i=1}^{D} s_i\bar{\mathbf{a}}_n(i) = 1$, where $s_i \in \{-1,1\}$ is decided by the information from potential PDE forms, e.g., s_1, s_2 are the same since coefficients for U_t and U_{tt} are of the same sign for all considered PDEs involving them. Although there are various PDEs being taken into consideration, the relations of the coefficient signs for the shared terms among different PDEs do not conflict, e.g., the signs for U_t and U_{xx} are always different for the viscous Burgers' equation and the attenuating wave equation (which indicates s_1 and s_4 are opposite). Setting $s_1 = 1$, we obtain the auxiliary vector $\mathbf{s} = [s_1\ \dots\ s_D]^{\mathrm{T}} \in \mathbb{R}^D$ for the dictionary (3.3) that is consistent with all the potential PDEs considered in this work as

$$\mathbf{s} = [1,1,1,-1,-1,1]^{\mathrm{T}} \in \mathbb{R}^{D} \tag{3.6}$$

and $\|\bar{\mathbf{a}}_n\|_1 = 1$ is reduced to $\mathbf{s}^{\mathrm{T}}\bar{\mathbf{a}}_n = 1$, which is affine.

The dictionary (3.3) includes the terms for all potential PDEs and the true governing PDE only involves a few of them by experience. Therefore, it is preferable for the identified PDE to have fewer active terms under the data fitting constraint. Thus we use ℓ_1-norm minimization due to its ability to promote sparsity [48, 49, 50, 51, 52, 53, 54]:

$$\bar{\mathbf{\Phi}}_n = \mathbf{\Phi}_n \mathbf{W}_n^{-1}, \quad \bar{\mathbf{\Phi}}_n^s = \begin{bmatrix} \bar{\mathbf{\Phi}}_n \\ \mathbf{s}^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{(M+1)\times D} \tag{3.7a}$$

$$\widehat{\bar{\mathbf{a}}}_n = \arg\min \|\bar{\mathbf{a}}_n\|_1 \ \text{ s.t. } \|\bar{\mathbf{\Phi}}_n^s \bar{\mathbf{a}}_n - \mathbf{e}\|_\infty \leq \tau_n \tag{3.7b}$$

$$\widehat{\mathbf{a}}_n = \mathbf{W}_n^{-1} \widehat{\bar{\mathbf{a}}}_n \ , \tag{3.7c}$$

where $\mathbf{e} \in \mathbb{R}^{M+1}$ has all zero entries except $\mathbf{e}(M+1) = 1$ and τ_n is a pre-defined toleration. The constraint $\|\bar{\mathbf{\Phi}}_n^s \bar{\mathbf{a}}_n - \mathbf{e}\|_\infty \leq \tau_n$ in (3.7b) enforces $\bar{\mathbf{\Phi}}_n \bar{\mathbf{a}}_n \approx \mathbf{0}$ and $\mathbf{s}^{\mathrm{T}}\bar{\mathbf{a}}_n \approx 1$. The optimization problem in (3.7) enables identifying all active PDE terms simultaneously and thus is more efficient than the previous methods in Refs. [14, 3] relying on iterations, which at first iteratively assume $\mathbf{a}_n(i) = 1$ for all $1 \leq i \leq D$ to solve (3.4), and then select the best assumption among these D cases.

The $\mathbf{s}$ in (3.6) is an example used for the considered PDEs in this paper for the spatially 1D case. Fixing the six entries in (3.6), it can be extended by adding more ± 1 entries to accommodate more PDE terms in the dictionary, e.g., (3.24) for the 2D extension. $\mathbf{\Phi}_n$ and $\mathbf{s}$ can also be extended to include more terms and thus encompass most PDEs for the dynamics of acoustical waves like the KdV equation, Stokes' wave equation[73], Van Wijngaarden's equation[73], etc. Since the sign relation in (3.6) is suitable for most acoustical PDEs, the $\mathbf{s}$ conveys less information in the potential kinds of PDEs compared to knowing at least one active

term and thus represents a weaker prior knowledge.

We only put the various dynamical patterns of U denoted by the PDE terms into the dictionary $\mathbf{\Phi}_n$, and leave the information of both the magnitudes and signs for these terms appearing in the PDE into the coefficient $\widehat{\mathbf{a}}_n$ for better physical interpretability. For example, for the wave equation we prefer the recovered PDE to have the form of (3.2) instead of $\mathsf{N}[U] = U_{tt} + \alpha U_t + c^2(-U_{xx})$. This is achieved by adding the negative signs onto entries of $\mathbf{s}$ instead of onto the corresponding columns in $\mathbf{\Phi}_n$ while keeping $\mathbf{s}$ an all-one vector.

The information in $\mathbf{s}$ enables identifying PDEs from limited data, because it encourages (3.7b) to select $\bar{\mathbf{a}}_n$ whose non-zero entries have the same signs as their corresponding entries in $\mathbf{s}$, and thus filter out the potential combinations of columns in $\bar{\mathbf{\Phi}}_n$ which are better fitted but have no physical meaning. To see this, suppose there are 2 vectors $\mathbf{p}, \mathbf{q} \in \mathbb{R}^D$ both satisfying the requirement for $\bar{\mathbf{a}}_n$ in (3.7b) for a small τ_n, i.e.,

$$\begin{gathered} \|\bar{\mathbf{\Phi}}_n \mathbf{p}\|_\infty \leq \tau_n,\ \|\bar{\mathbf{\Phi}}_n \mathbf{q}\|_\infty \leq \tau_n, \\ \left|\sum_{i=1}^{D} s_i p_i - 1\right| \leq \tau_n,\ \left|\sum_{i=1}^{D} s_i q_i - 1\right| \leq \tau_n \end{gathered} \tag{3.8}$$

and thus using the triangle inequality:

$$\left|\sum_{i=1}^{D} s_i p_i - \sum_{i=1}^{D} s_i q_i\right| \leq 2\tau_n \ . \tag{3.9}$$

If the signs of non-zero entries in $\mathbf{p}$ are the same as corresponding entries in $\mathbf{s}$, while for $\mathbf{q}$ one entry q_{i_0} has the opposite sign of s_{i_0} resulting in $s_{i_0} q_{i_0} < 0$, then when $-s_{i_0} q_{i_0} > \tau_n$ which is likely for a small τ_n, the method will choose $\mathbf{p}$ over $\mathbf{q}$ because of a smaller ℓ_1 norm:

$$\|\mathbf{p}\|_1 = \sum_{i=1}^{D} s_i p_i < \sum_{i=1}^{D} s_i q_i - 2 s_{i_0} q_{i_0} = \|\mathbf{q}\|_1 \ . \tag{3.10}$$

An example of the failed identification due to the replacement of the informative $\mathbf{s}$ by $\mathbf{1}$ is given

in Sec. 3.4.1.

In addition to encouraging selecting the $\bar{\mathbf{a}}_n$ whose entries have the correct signs, the incorporation of the non-zero $\mathbf{s}$ also aims to avoid the trivial solution $\bar{\mathbf{a}}_n = \mathbf{0}$ and thus the sign information $\mathbf{s}$ (even the incorrect $\mathbf{s}$ which leads to a wrong solution as discussed above) must be provided to make the method work. Meanwhile, to ensure that every column of $\bar{\mathbf{\Phi}}_n^s$ in (3.7a) has the same ℓ_2-norm so that the optimization is not influenced by magnitudes of dictionary atoms, the entries in $\mathbf{s}$ should have the same magnitude. Here the magnitude of one is used for simplicity.

We repeat (3.7) for all the N locations in the ROI and thus recover the physical properties described by spatially dependent PDEs. Note that for different locations the set of recovered active PDE terms can be different. For example, in a wavefield, the attenuation of the waves can be negligible in some regions and thus the wave equation (3.2) does not have U_t term, but for other regions where the attenuation is obvious the wave equation contains the term U_t.

To accelerate searching the PDE terms, we utilize the equivalence of the ℓ_2-norm and ℓ_∞-norm for a vector (for any $\mathbf{v} \in \mathbb{R}^k$, $\|\mathbf{v}\|_\infty \leq \|\mathbf{v}\|_2 \leq \sqrt{k}\|\mathbf{v}\|_\infty$), replace the ℓ_∞-norm constraint in (3.7b) with ℓ_2-norm which is $\|\bar{\mathbf{\Phi}}_n^s \bar{\mathbf{a}}_n - \mathbf{e}\|_2^2 \leq \tau_n$, and solve it using its Lagrangian (i.e., lasso [55, 56]):

$$\begin{aligned} \widehat{\bar{\mathbf{a}}}_n &= \arg\min_{\bar{\mathbf{a}}_n} \|\bar{\mathbf{a}}_n\|_1 + \lambda(\|\bar{\mathbf{\Phi}}_n^s \bar{\mathbf{a}}_n - \mathbf{e}\|_2^2 - \tau_n) \\ &= \arg\min_{\bar{\mathbf{a}}_n} \|\bar{\mathbf{\Phi}}_n^s \bar{\mathbf{a}}_n - \mathbf{e}\|_2^2 + \lambda_n \|\bar{\mathbf{a}}_n\|_1 \ , \end{aligned} \tag{3.11}$$

where $\lambda_n = \frac{1}{\lambda} = 0.2\lambda_0$ is chosen empirically with $\lambda_0 = 2\|\bar{\mathbf{\Phi}}_n^{s\mathrm{T}} \mathbf{e}\|_\infty = 2$, the boundary parameter above which the output of (3.11) is $\mathbf{0}$ according to the lasso path [57, 58, 59]. The (3.11) can be efficiently solved by coordinate descent, where a complete iteration of updating all D entries in $\bar{\mathbf{a}}_n$ costs $O((M+1)D)$ operations [60], and the number of iterations to reach convergence is often small.

Due to the noise from numerical computation, the $\bar{\mathbf{a}}_n$ minimizing (3.11) may not be sparse enough. To further promote sparsity, we threshold entries of $\bar{\mathbf{a}}_n$ using an adaptive threshold

proportional to $\|\bar{\mathbf{a}}_n\|_\infty$. Then use least squares regression to solve the coefficients only in the T kept entries (denoted by $\tilde{\mathbf{a}}_n(\Lambda)$, where Λ with cardinality T is the set of indices for the T kept non-zero entries) and assign 0 to other entries. Thus (3.7c) is replaced by:

$$\Lambda = \{\forall i, |\bar{\mathbf{a}}_n(i)| \geq \varepsilon \|\bar{\mathbf{a}}_n\|_\infty\} \tag{3.12a}$$

$$\tilde{\mathbf{\Phi}}_n^s = [\mathbf{\Phi}_n(:,\Lambda)^{\mathrm{T}} \ \ \mathbf{s}(\Lambda)]^{\mathrm{T}} \in \mathbb{R}^{(M+1)\times T} \tag{3.12b}$$

$$\widehat{\tilde{\mathbf{a}}}_n(\Lambda) = \tilde{\mathbf{\Phi}}_n^{s\dagger}\mathbf{e}\ , \tag{3.12c}$$

where ε is the threshold to be tuned. It can be tuned according to prior knowledge, or from grid search and cross validation if training data is available.

3.2.2 Denoising by integration

In this part, we extend the technique of integration transformation[12] to the spatially-dependent PDE identification to make it more robust to noise, see Sec. 3.2.2, and discuss implementation details in Sec. 3.2.2. Overall, the idea is to replace the $\mathbf{\Phi}_n$ in (3.3) with a new dictionary $\mathbf{\Phi}_n^{\mathrm{int}}$ built from integration transformation, as detailed below. All other steps for the PDE identification are the same as in Sec. 3.2.1.

Transformed dictionary by integration

The terms in $\mathbf{\Phi}_n$ defined in (3.3) are sensitive to noise, because the noise is typically broadband and its influence is emphasized by differentiation. We extend the denoising technique by integration [12] to spatially-dependent PDE identification to recover the spatial variation of properties from noisy measurements. To be specific, replacing $\mathbf{\Phi}_n$ in (3.3) with another dictionary $\mathbf{\Phi}_n^{\mathrm{int}}$ as described below and finishing all following steps for PDE identification in Sec. 3.2.1 using it.

The integration method uses integration by parts to transfer the derivatives of noisy measurements $\mathbf{U}$ to the derivatives of a pre-defined weighting function $\mathbf{W}$, which is noise-free.

For an arbitrary region Ω in the ROI and a finite smooth function $W(x,t)$ defined on Ω,

$$\mathsf{N}[U]=0 \quad \rightarrow \quad \int_{\Omega}\mathsf{N}[U]W\mathrm{d}\Omega=0\,, \tag{3.13}$$

where $\int_{\Omega}\mathsf{N}[U]W\mathrm{d}\Omega$ is for $\int_{(x,t)\in\Omega}\mathsf{N}[U(x,t)]W(x,t)\mathrm{d}x\mathrm{d}t$. From (3.13), in an arbitrary region Ω within the ROI, the integral of the product between W and the summation of all active PDE terms is zero. E.g., if $\mathsf{N}(U)$ is the wave equation (3.2), then (3.13) becomes

$$\int_{\Omega}(U_{tt}+\alpha U_t-c^2U_{xx})W\mathrm{d}\Omega=0\,. \tag{3.14}$$

Let Ω be a square region $\{x_l\le x\le x_u, t_l\le t\le t_u\}$ with $(x_u-x_l)/\Delta x=(t_u-t_l)/\Delta t=2a$ intervals so that it covers $(2a+1)\times(2a+1)$ spatial-temporal coordinates within the ROI, we can move the derivatives of U in (3.13) onto W, e.g., for component U_t

$$\begin{aligned}\int_{\Omega}WU_t\mathrm{d}\Omega&=\int_{\Omega}(WU)_t\mathrm{d}\Omega-\int_{\Omega}UW_t\mathrm{d}\Omega\\&=\int_{x_l}^{x_u}\left[(WU)\Big|_{t_l}^{t_u}\right]\mathrm{d}x-\int_{\Omega}UW_t\mathrm{d}\Omega\\&=-\int_{\Omega}UW_t\mathrm{d}\Omega\,,\end{aligned} \tag{3.15}$$

where W is defined as 0 at t_l and t_u. Similarly, using a W with its $(p-1)$th order derivatives $(p\ge 2)$ vanished at its spatial-temporal boundaries, we can transfer the derivatives on U for all terms in (3.3) onto W. An eligible W is

$$\begin{aligned}&W(x,t)=(\bar{x}^2-1)^p(\bar{t}^2-1)^p\,,\text{where}\\&\bar{x}=2\frac{x-x_l}{x_u-x_l}-1\in[-1,1],\ \bar{t}=2\frac{t-t_l}{t_u-t_l}-1\in[-1,1]\,.\end{aligned} \tag{3.16}$$

Since the PDEs are assumed spatially-dependent, for spatial location n_x in $\mathbf{U}$, we select M_{int} integration domains $\Omega_1,\ldots,\Omega_{M_{\text{int}}}$ centered at $(n_x,(m_0+\delta)_t),\ldots,(n_x,(m_0+M_{\text{int}}\delta)_t)$ where δ is the interval between temporal centers of two neighboring domains, and the spatial center of

the domains is always the location n_x. All the Ω_m with $m = 1, \ldots, M_{\text{int}}$ are of the same size as the Ω mentioned before. Ideally $\delta = 1$, but a larger δ is chosen to reduce computations. Thus we can construct a new library of atoms $\mathbf{\Phi}_n^{\text{int}} \in \mathbb{C}^{M_{\text{int}} \times D}$ consisting of integration for $D = 6$ integrands on all $\Omega_m,\ 1 \leq m \leq M_{\text{int}}$, whose mth row is

$$\begin{aligned} \mathbf{\Phi}_n^{\text{int}}(m,:) = \int_{\Omega_m} [-UW_t^m,\ UW_{tt}^m,\ -\frac{1}{2}U^2 W_x^m,\ UW_{xx}^m, \\ UW_{xt}^m,\ \sin(U)W^m] \mathrm{d}\Omega_m\ , \end{aligned} \tag{3.17}$$

where W^m is the shifted W so that its domain exactly overlaps Ω_m.

Computation of integrals

To compute entries in (3.17) numerically, we evaluate the values of W^m at discrete coordinates and save it into $\mathbf{W}^m \in \mathbb{R}^{N \times M}$, in which

$$\mathbf{W}^m(i_x, i_t) = \begin{cases} W(i_x \Delta x, i_t \Delta t), & \text{if } (i_x, i_t) \in \Omega_m \\ 0, & \text{otherwise} \end{cases} \tag{3.18}$$

where W is pre-defined with an eligible choice given in (3.16) whose x_u, x_l, t_u and t_l are the spatio-temporal boundary of Ω_m. Similarly, the derivatives are also computed and saved in matrices in $\mathbb{R}^{N \times M}$, e.g.,

$$\mathbf{W}_{tt}^m(i_x, i_t) = \begin{cases} W_{tt}(i_x \Delta x, i_t \Delta x), & \text{if } (i_x, i_t) \in \Omega_m \\ 0, & \text{otherwise} \end{cases} \tag{3.19}$$

where $W_{tt}(i_x \Delta x, i_t \Delta t) = \frac{\partial^2 W(x,t)}{\partial t^2}\Big|_{x = i_x \Delta x, t = i_t \Delta t}$ is computed analytically and thus noise-free.

A demonstration of the integration is shown in Fig. 3.1(a), where $M_{\text{int}} = 4$. The integration of dth integrand indicated in (3.17) within Ω_m is $\mathbf{\Phi}_n^{\text{int}}(m,d)$. The Fig. 3.1(b) shows a $\mathbf{W}^m$ with its non-zero region in $\Omega_m = \{(x,t) | 20\Delta x \leq x \leq 30\Delta x, 17\Delta t \leq t \leq 27\Delta t\}$ enclosed by a dashed box.

By choosing W as (3.16) with $p \geq 2$ and analytically computing its derivatives, all derivatives of W necessary for (3.17) are obtained, enabling the transfer of derivatives on U to derivatives on W. The integral for each term in (3.17) within each Ω_m is integrated along t and then along x numerically by summation approximation. Note that W and its derivatives are calculated analytically with $\bar{t}$ in $[-1,1]$, but the true domain of W covers $[-a\Delta t, a\Delta t]$. Thus, for the p_0th order temporal derivative of W, its real values are the analytical result divided by $(a\Delta t)^{p_0}$, and for spatial derivatives the scaling is similar. For example, the mth row of the 2nd column in $\mathbf{\Phi}_n^{\text{int}}$ is computed as

$$\mathbf{\Phi}_n^{\text{int}}(m,2) = (\frac{1}{a\Delta t})^2 \sum\nolimits_{\{i_x,i_t\}\in\Omega_m} \mathbf{U}(i_x,i_t)\mathbf{W}_{tt}^m(i_x,i_t)\Delta x \Delta t, \tag{3.20}$$

which is a scaled summation of $(2a+1)^2$ values of the element-wise product between $\mathbf{U}$ and $\mathbf{W}_{tt}^m$ within Ω_m.

To reduce the error caused by approximating the integration with a finite sum, which can dominate when the derivative of W has a high order, we use interpolation. Before integration, each time slice of the raw signal $\mathbf{U}$ within Ω_m (e.g., red line in Ω_2 in Fig. 3.1(a)) is interpolated using polynomial fitting. Let the raw signal be the blue line in Fig. 3.1(c) with $2a+1$ points (here $a=5$), we fit it using a 5th order polynomial and interpolate q values evenly between each neighboring points. Therefore the interpolated signal has $2a(q+1)+1$ points. To match the interpolated signal, W^m and its derivatives are also evaluated at these evenly spaced $2a(q+1)+1$ points along the temporal direction. We then sum up values of the integrand at all these points to be the temporal integral for this location, and the $\mathbf{\Phi}_n^{\text{int}}(m,d)$ is acquired by summing up such temporal integrals for all $2a+1$ spatial locations within Ω_m. Thus, in the example for (3.20), $\mathbf{\Phi}_n^{\text{int}}(m,2)$ becomes a scaled summation of $(2a+1)\times(2a(q+1)+1)$ values.

Replacing $\mathbf{\Phi}_n$ in (3.3) with $\mathbf{\Phi}_n^{\text{int}}$ efficiently increases its robustness to noise, as shown in Sec. 3.3.3 and 3.4.2. In all experiments, $a=5$, $p=3$, $q=9$.

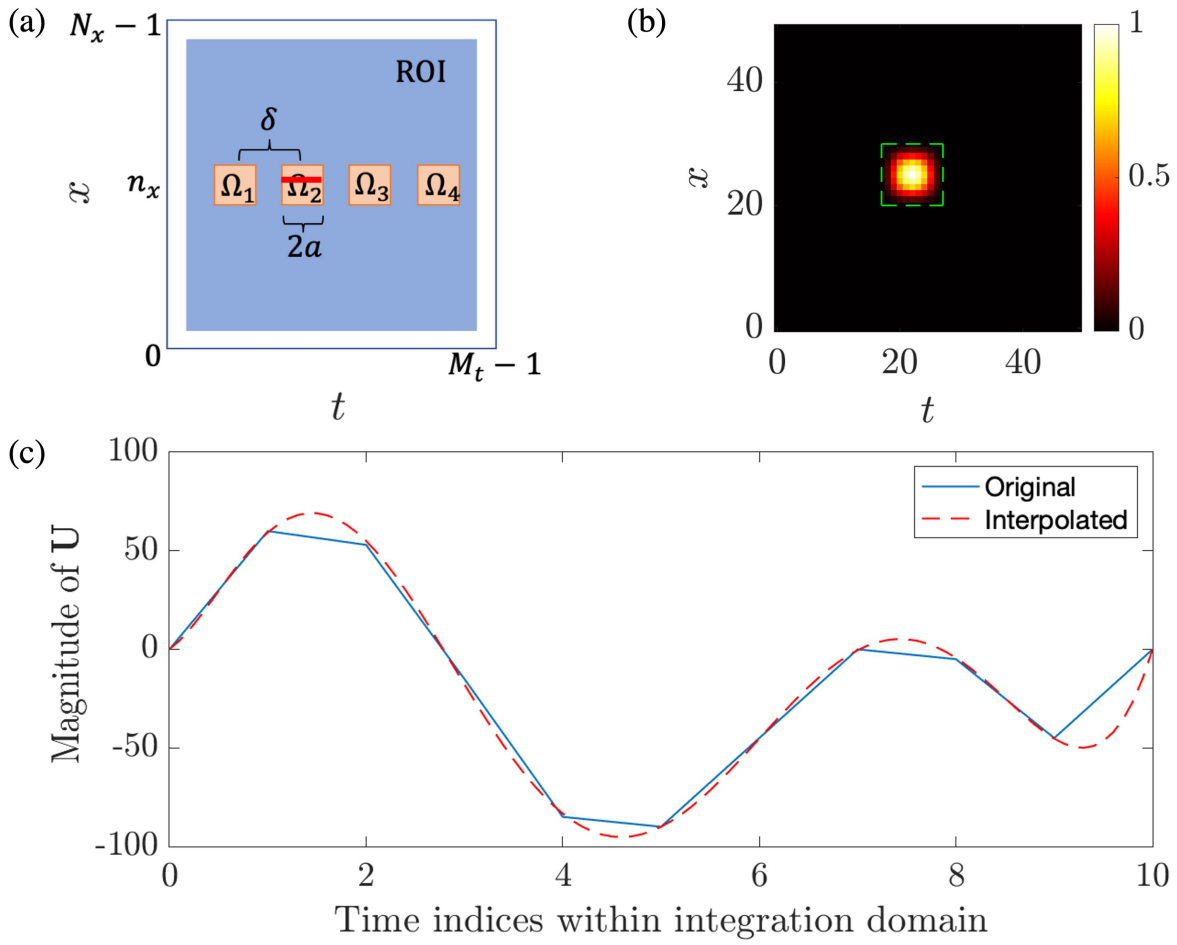

Figure 3.1. (a) 4 integration domains in ROI with each having $2a+1$ points in both x,t axes. (b) A $\mathbf{W}^m$ whose non-zero part centered at $(x = 25\Delta x, t = 22\Delta t)$. (c) When integrating the signal along time (e.g., red line in (a)), use polynomial interpolation and integrate on the interpolated slice (red dashed line).

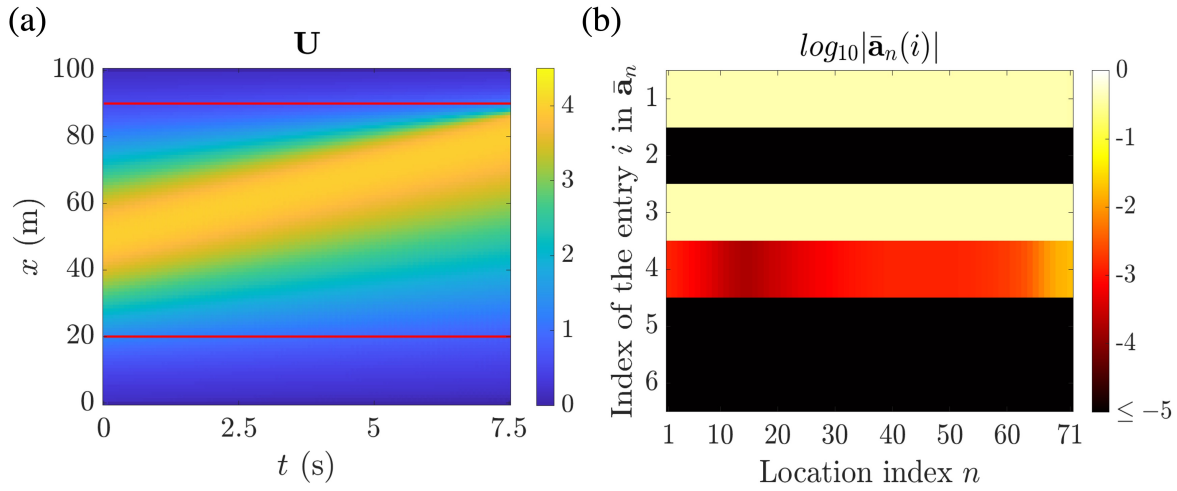

Figure 3.2. For Burgers' equation (3.22), (a) $\mathbf{U}$ with $\nu = 0.1$, $\Delta x = 1$ m, $\Delta t = 0.05$ s. Therefore $0 \leq i_x \leq 100$, $0 \leq i_t \leq 150$. The ROI is $\{i_x | 20 \leq i_x \leq 90\}$ between the red lines where obvious dynamics is observed. (b) $log_{10}|\bar{\mathbf{a}}_n(i)|$ where i corresponds to the indices in (3.21) for all $n = 1 \ldots 71$.

3.3 Numerical experiments

This part includes a spatially independent (not known *a priori*) and two spatially dependent PDE identification experiments. The Sec. 3.3.1 to 3.3.2 are based on clean signal, and Sec. 3.3.3 experiments with noisy measurements. The datasets are generated by finite difference modeling [9].

For the 1D case in Sec. 3.3.1, the dictionary used is (3.3). To simplify the demonstration, we index its columns as

$$\mathbf{\Phi}_n = \left[\overset{1}{\mathbf{u}_t}, \quad \overset{2}{\mathbf{u}_{tt}}, \quad \overset{3}{\mathbf{u} \circ \mathbf{u}_x}, \quad \overset{4}{\mathbf{u}_{xx}}, \quad \overset{5}{\mathbf{u}_{tx}}, \quad \overset{6}{\sin(\mathbf{u})} \right] \in \mathbb{R}^{M \times 6} . \tag{3.21}$$

and use the indices for the columns ($1 \sim 6$) to show the results.

3.3.1 Spatially-independent Burgers' equation

Here we recover the fluid viscosity from fluid speeds. The Burgers' equation [61, 62]

$$U_t + UU_x - \nu U_{xx} = 0 \tag{3.22}$$

is non-linear and can model the formation of shock waves in free turbulence, where $\nu \geq 0$ is the viscosity of the fluid which is spatially-independent in this example. The aim of this experiment is to show that the method can extract the medium property from a non-linear dynamic system, and it works when the PDE to be identified is in fact spatially-independent (indicating a homogeneous medium) but not known *a priori*.

A dataset $\mathbf{U} \in \mathbb{R}^{101\times 151}$ with $\Delta t = 0.05$ s and $\Delta x = 1$ m modeling the speed of the fluid at each location along a thin pipe as time progresses governed by (3.22) with $\nu = 0.1$ is generated as Fig. 3.2(a). The initial state is a scaled probability density function (PDF) of the normal distribution, as time goes by, the wave is moving in the positive x direction.

We choose the spatial region where the dynamics can be easily observed as the ROI, to be specific, choose $I_x = \{i_x | 20 \leq i_x \leq 90\}$ which is bounded by the red lines in Fig. 3.2(a), so $N = 71$ and $n = 1$ corresponds to $i_x = 20$. We do not consider the temporal boundaries where the derivatives are not well defined and thus use $I_t = \{i_t | 1 \leq i_x \leq 149\}$ for the ROI, i.e., $M = 149$.

To find the governing PDE, we build $\bar{\mathbf{\Phi}}_n^s$ according to (3.3), (3.6) and (3.7a) for every $1 \leq n \leq N$. From (3.11), the coefficients are distributed as Fig. 3.2(b). After thresholding using (3.12a) with $\varepsilon = 10^{-3}$, $\{\mathbf{u}_t, \mathbf{u} \circ \mathbf{u}_x, \mathbf{u}_{xx}\}$ appearing in the Burgers' equation are selected for all locations in the ROI.

For every location, we build $\tilde{\mathbf{\Phi}}_n^s$ and compute $\tilde{\mathbf{a}}_n$ as (3.12). The coefficients for U_t and UU_x are always nearly identical since $\sum_{n=1}^{N} |\tilde{\mathbf{a}}_n(3) - \tilde{\mathbf{a}}_n(1)| = 1.7 \times 10^{-14}$. The estimated viscosity is $\widehat{\nu}_n = -\frac{\tilde{\mathbf{a}}_n(4)}{\tilde{\mathbf{a}}_n(1)}$, which equals to 0.1 for every n.

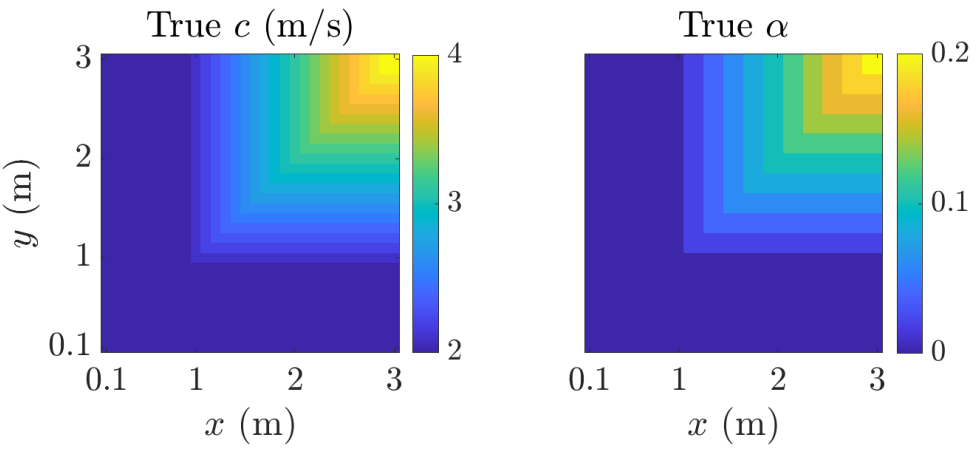

Figure 3.3. The true phase speeds c and attenuating factors α for i_x, i_y in $[1, 30]$ ($\Delta x{=}\Delta y{=}0.1$ m). Waves can not arrive at the places where either i_x or i_y is 0 or 31 because of the boundary condition.

3.3.2 2D spatially-dependent wave equation

In this part, we recover the 2D maps of phase speeds and attenuation from observed propagating waves. A 2D wavefield $\mathbf{U} \in \mathbb{R}^{32\times32\times200}$ in which $\Delta x = \Delta y = 0.1$ m and $\Delta t = 0.01$ s describing waves excited by an initial perturbation and propagating through various media is used for the experiment. The PDE governing it is the wave equation

$$U_{tt} + \alpha U_t - c^2 \nabla^2 U = 0 \,, \tag{3.23}$$

where $\alpha \geq 0$ the attenuating factor, $c > 0$ the phase speed and ∇^2 the Laplacian, i.e., $U_{xx} + U_{yy}$. We adopt the Dirichlet boundary condition, where the measurements on spatial boundaries are zero. The initial perturbation is shaped as a scaled 2D normal distribution PDF, and the phase speeds $2 \leq c \leq 4$ m/s and attenuation $0 \leq \alpha \leq 0.2$ are varying across the domain as shown in Fig. 3.3. Some frames are shown in Fig. 3.4. We choose the ROI to be all the spatial regions without the boundaries and its immediate neighboring points (i.e., $2 \leq i_x \leq 29$, $2 \leq i_y \leq 29$) and the time steps $1 \leq i_t \leq 198$. Therefore $N = 28^2 = 784$, $M = 198$ and $n = 1$ corresponds to $i_x = i_y = 2$. The 2D locations within ROI are indexed from 1 to N in the row-major manner.

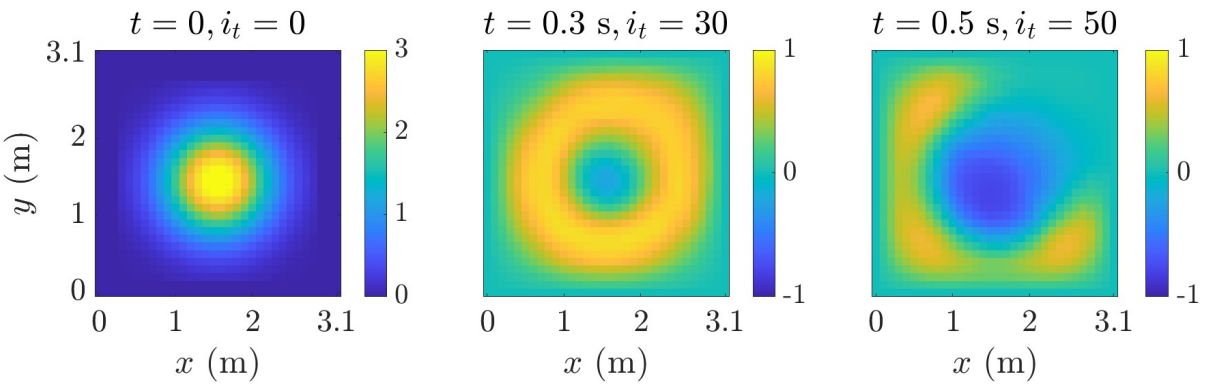

Figure 3.4. The wavefield governed by (3.23) with spatially dependent c and α at 3 selected time points. $\Delta x = \Delta y = 0.1$ m, $\Delta t = 0.01$ s. Therefore i_x, i_y both ranged from 0 to 31.

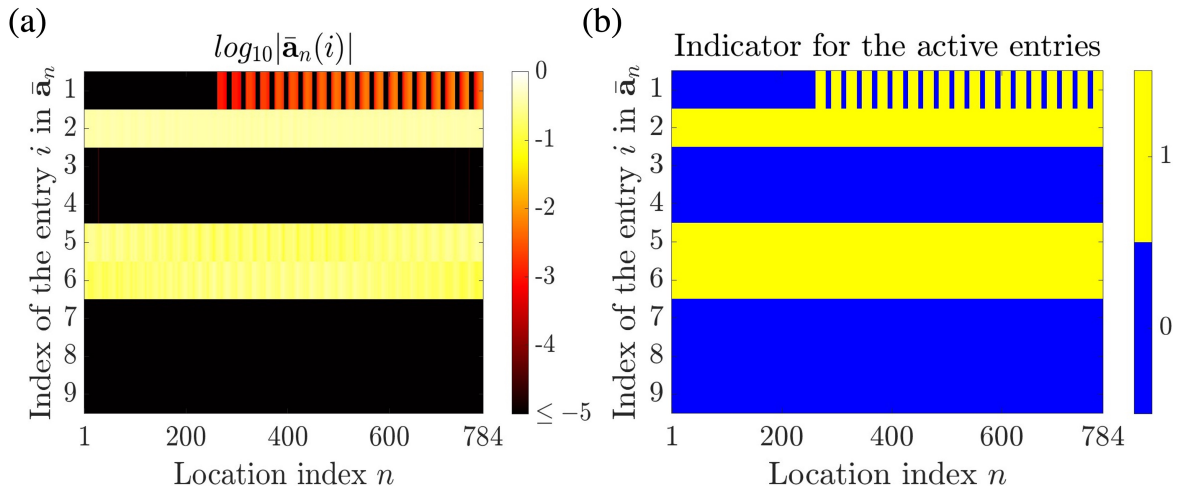

Figure 3.5. For 2D wave equation, (a) $log_{10}|\bar{\mathbf{a}}_n(i)|$ where i corresponds to the indices of columns for $\mathbf{\Phi}_n$ in (3.24) for all $n = 1 \ldots 28^2$; (b) the locations of their active entries after thresholding. The 2D 28×28 locations are indexed from 1 to 784 in a row-major manner.

For this 2D case, we extend the dictionary (3.3) and $\mathbf{s}$ (3.6) to

$$\begin{aligned} \mathbf{\Phi}_n &= [\overset{1}{\mathbf{u}_t}, \overset{2}{\mathbf{u}_{tt}}, \overset{3}{\mathbf{u} \circ \mathbf{u}_x}, \overset{4}{\mathbf{u} \circ \mathbf{u}_y}, \overset{5}{\mathbf{u}_{xx}}, \overset{6}{\mathbf{u}_{yy}}, \overset{7}{\mathbf{u}_{tx}}, \overset{8}{\mathbf{u}_{ty}}, \overset{9}{\sin(\mathbf{u})}] \\ \mathbf{s} &= [1,1,1,1,-1,-1,-1,-1,1]^{\mathrm{T}} . \end{aligned} \tag{3.24}$$

Build dictionaries using (3.24) according to (3.7a), and from (3.11), the $\bar{\mathbf{a}}_1$ to $\bar{\mathbf{a}}_N$ are acquired as Fig. 3.5(a). After thresholding as (3.12a) with $\varepsilon = 10^{-3}$, the kept non-zero entries are indicated in Fig. 3.5(b).

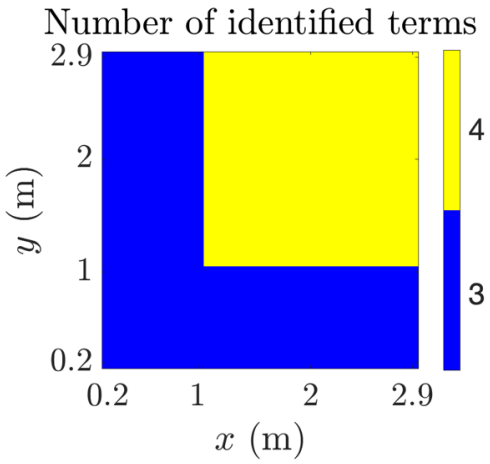

Figure 3.6. Number of identified active PDE terms within the ROI.

Comparing Fig. 3.5(b) to (3.23), the method successfully identifies the PDEs for all 784 locations. Fig. 3.6 shows the number of identified active PDE terms for each location in the ROI, where the 3 terms contain $\{\mathbf{u}_{tt}, \mathbf{u}_{xx}, \mathbf{u}_{yy}\}$, and for 4 terms $\mathbf{u}_t$ is also included. Build $\tilde{\mathbf{\Phi}}_n^s$ and compute $\tilde{\mathbf{a}}_n$ as (3.12), the coefficients for U_{xx} and U_{yy} are nearly identical as $\sum_{n=1}^{N} |\tilde{\mathbf{a}}_n(5) - \tilde{\mathbf{a}}_n(6)| = 3.3 \times 10^{-12}$. The recovered $\widehat{c}_n = \sqrt{-\frac{\tilde{\mathbf{a}}_n(5)+\tilde{\mathbf{a}}_n(6)}{2\tilde{\mathbf{a}}_n(2)}}$ and $\widehat{\alpha}_n = \frac{\tilde{\mathbf{a}}_n(1)}{\tilde{\mathbf{a}}_n(2)}$, which are satisfactory as the RMSE $= 1.1 \times 10^{-14}$ m/s for phase speeds and 1.5×10^{-14} for attenuating factors with respect to the ground truth in Fig. 3.3 across the whole ROI.

3.3.3 2D spatially-dependent wave equation with noise

The field is of size $\mathbf{U} \in \mathbb{R}^{100\times100\times1000}$, $\Delta x = \Delta y = 1$ m, $\Delta t = 0.2$ s. The ROI is selected to be $5 \le i_x < 95, 5 \le i_y < 95$ (8100 locations in total) and for time period use $5 \le i_t < 995$. The field has free boundaries and is excited by two chirp sources located outside the region at $(-6\text{ m}, 18\text{ m})$ and $(109\text{ m}, 80\text{ m})$, and governed by attenuating or non-attenuating wave equations with various coefficients, as indicated in Fig. 3.7(a). One frame is in Fig. 3.8(a).

Without noise, build dictionaries for all locations using (3.24) according to (3.7a) and implement (3.11), we obtain the coefficients distributed as Fig. 3.9. Thresholding them with $\varepsilon = 10^{-2}$, the PDEs are correctly identified for all 8100 locations in the ROI. The $\widehat{c}$ and $\widehat{\alpha}$ are well recovered, with the RMSE $= 1.7 \times 10^{-15}$ m/s for phase speeds and 3.2×10^{-16} for

attenuating factors comparing to the ground truth.

We add additive white Gaussian noise (AWGN) with $\sigma^2 = 2$ (arbitrarily chosen to be $\sim 1\%$ of the signal variance in a frame) to the measurements, see Fig. 3.8(b). Using the same procedure as before without denoising, the $\bar{\mathbf{a}}_n$ is as Fig. 3.10(a). No proper ε can be found to extract the active terms and thus the PDE identification fails. We thus use the integration transformation to build $\mathbf{\Phi}_n^{\text{int}} \in \mathbb{R}^{M_{\text{int}} \times 9}$ for each location as described in Sec. 3.2.2 with its mth row

$$\begin{aligned}\mathbf{\Phi}_n^{\text{int}}(m,:) = \int_{\Omega_m} [-\overset{1}{U}\overset{m}{W_t}, \overset{2}{U}\overset{m}{W_{tt}}, -\frac{1}{2}\overset{3}{U^2}\overset{m}{W_x}, -\frac{1}{2}\overset{4}{U^2}\overset{m}{W_y}, \\ \overset{5}{U}\overset{m}{W_{xx}}, \overset{6}{U}\overset{m}{W_{yy}}, \overset{7}{U}\overset{m}{W_{xt}}, \overset{8}{U}\overset{m}{W_{yt}}, \sin(\overset{9}{U})W^m] \mathrm{d}\Omega_m \,.\end{aligned} \tag{3.25}$$

The center of the integration domains starts from $i_t = 19$ and ends at $i_t = 979$ with $\delta = 10$, so $M_{\text{int}} = 97$. The recovered $\bar{\mathbf{a}}_n$ is in Fig. 3.10(b), a clear improvement of Fig. 3.10(a). Now it is possible to use thresholding to extract active terms. Using $\varepsilon = 0.13$, the recovered result is shown in Fig. 3.7(b) where the terms corresponding to $\{U_{tt}, U_{xx}, U_{yy}\}$ or $\{U_t, U_{tt}, U_{xx}, U_{yy}\}$ are identified at 97.6% of all 8100 locations. Specifically, for the ROI with attenuation (upper right in Fig. 3.7(a)), the four PDE terms $\{U_t, U_{tt}, U_{xx}, U_{yy}\}$ are selected at 86.1% of the 45^2 locations. For each location of the 2.4% region where the wave equation is not identified, the recovered $\widehat{c}$ and $\widehat{\alpha}$ are interpolated using the median value within a window covering 21 locations along y-axis centered at it.

For the phase speed recovery, the sharp edges between distinct true speeds are smoothed. The reason is that the integration domains centered near the edges cover the regions with different speeds, thus the results are affected by both speeds. The recovered speed smoothly changes because the integration domain smoothly slides over the boundary. The integration domain size is important: a larger integration domain leads to more noise-robust estimation and more extensively smoothed edges. Unlike phase speed recovery, when the integration domain centers near the boundary of attenuating and non-attenuating areas, if $\bar{\mathbf{a}}_n(1)$ is kept after thresholding in (3.12a) then $\widehat{\alpha}$ is recovered from least squares regression in (3.12c) and thus near the true α,

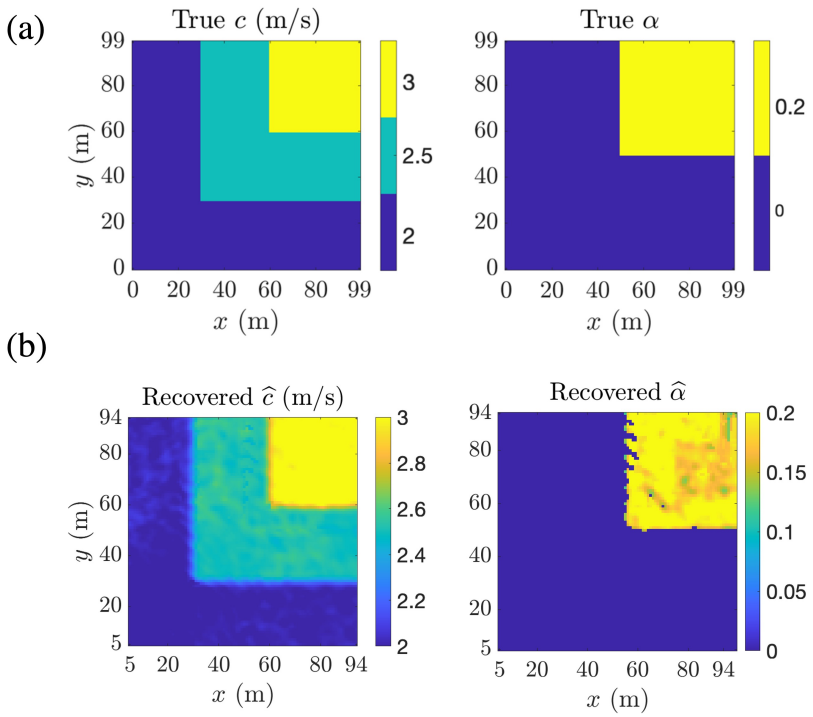

Figure 3.7. (a) The true phase speeds c and attenuating factors α for i_x, i_y in $[0,99]$ ($\Delta x = \Delta y =$ 1 m). (b) The recovered $\hat{c}$ and $\hat{\alpha}$ in the ROI for the noisy measurements using $\varepsilon = 0.13$.

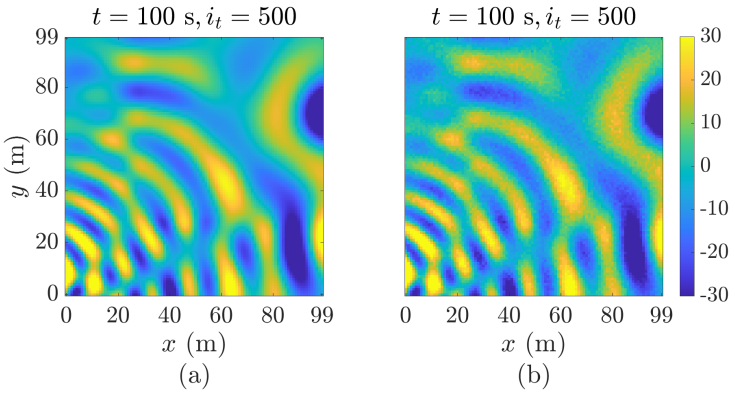

Figure 3.8. A frame of the wavefield governed by (3.23) with spatially dependent c and α in Fig. 3.7(a). The 2 sources are outside the region at $(-6\text{ m}, 18\text{ m})$ and $(109\text{ m}, 80\text{ m})$. (a): clean measurements; (b): noisy measurements with AWGN for $\sigma^2 = 2$.

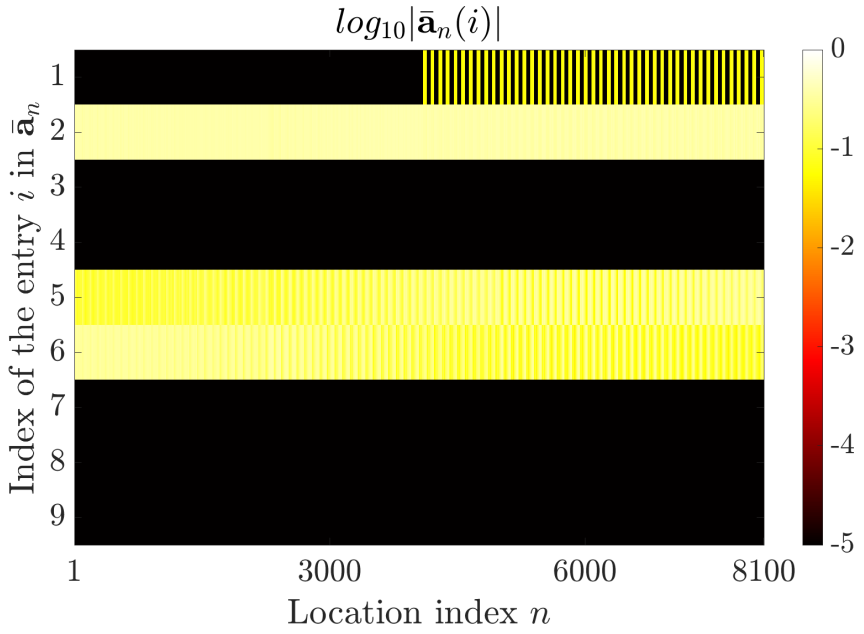

Figure 3.9. For 2D wave equation with spatially dependent c and α in Fig. 3.7(a), $log_{10}|\bar{\mathbf{a}}_n(i)|$ from clean measurements where i corresponds to the indices of columns for $\mathbf{\Phi}_n$ in (3.24).

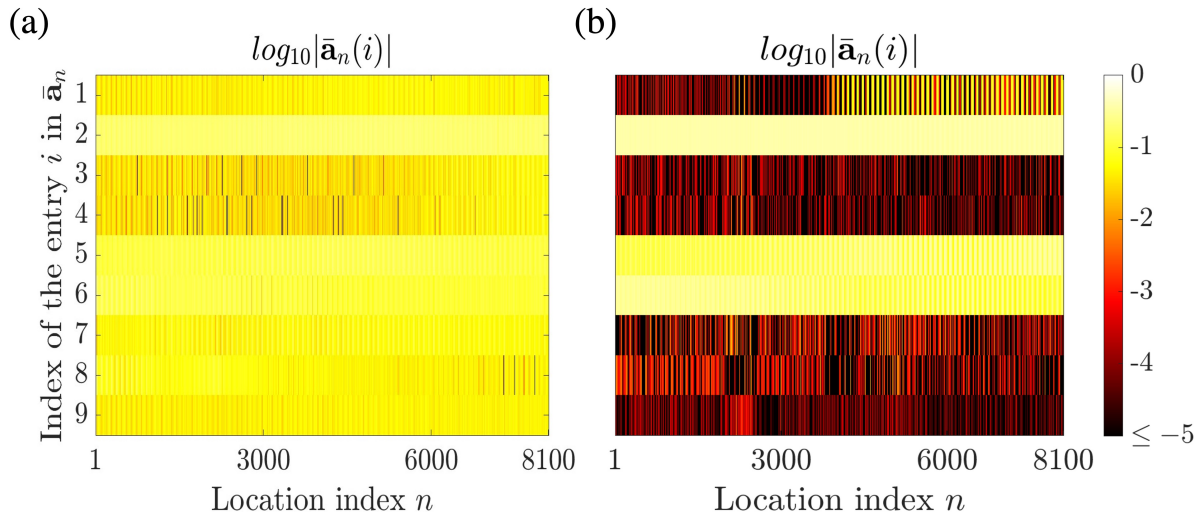

Figure 3.10. For noisy measurements of waves governed by (3.23) with c and α in Fig. 3.7(a): (a) $log_{10}|\bar{\mathbf{a}}_n(i)|$ with $\bar{\mathbf{a}}_n$ recovered from $\mathbf{\Phi}_n$ where i corresponds to the indices of columns for $\mathbf{\Phi}_n$ in (3.24); (b) $log_{10}|\bar{\mathbf{a}}_n(i)|$ with $\bar{\mathbf{a}}_n$ recovered from $\mathbf{\Phi}_n^{\text{int}}$ in (3.25).

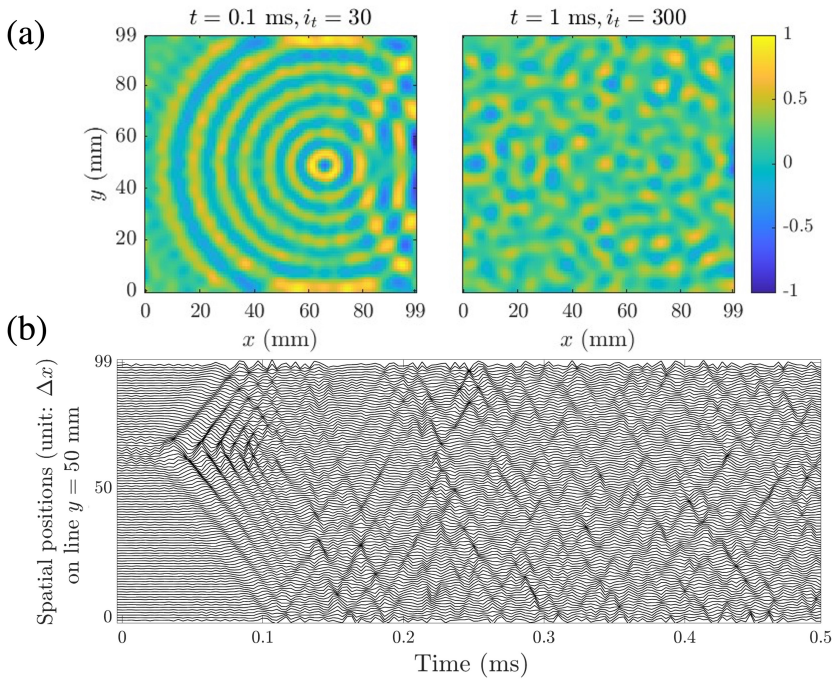

Figure 3.11. The vibrating plate: (a) 2 of the selected 1000 frames, with magnitudes normalized; (b) the traces for locations at $y = 50$ mm in the first 0.5 ms.

otherwise $\tilde{\mathbf{a}}_n(1)$ is set to 0, thus $\widehat{\alpha} = 0$, causing the sharp transition for $\widehat{\alpha}$.

3.4 extracting PDEs for a vibrating plate

3.4.1 Identification from clean measurements

The approach is demonstrated on laser scanned measurements of vibrations of a real aluminum plate sampled at 300 kHz provided by University of Utah [36], see Fig. 3.11. The part taken into consideration is $\mathbf{U} \in \mathbb{R}^{100\times100\times1000}$, i.e., the measurements are collected from 10000 sampling locations uniformly distributed on the square plate (100 rows and 100 columns) with 1000 time steps. The spatial sampling interval $\Delta x = \Delta y = 1$ mm.

The PDEs governing the vibrations of the plate is the wave equation. Since the aluminum plate waves are dispersive [37], i.e., phase speeds c varies across frequencies, we extract the narrow band signals from $\mathbf{U}$ and identify the PDE for every band. Five 6th order Butterworth bandpass filters centered at 30 to 70 kHz stepped by 10 kHz are employed to extract narrowband

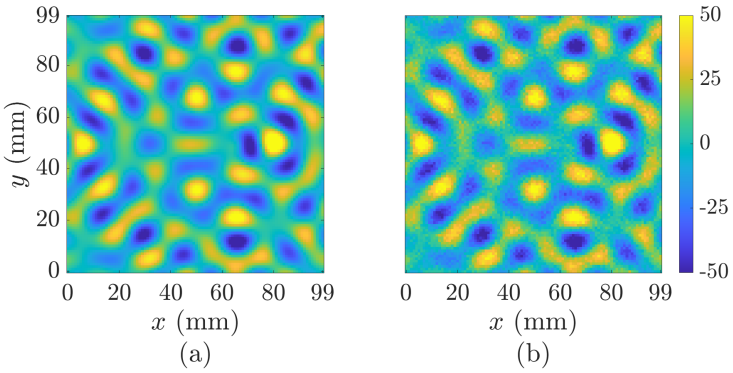

Figure 3.12. For the 300th frame of the signal with frequency band centered at 30 kHz, (a) clean signal; (b) signal corrupted by AWGN with $\sigma^2 = 10^3$.

signals with 2 kHz bandwidth, with one frame of the narrowband signal centered at 30 kHz shown in Fig. 3.12 (a). We drop five neighboring locations on each axis end, so for the ROI $5 \le i_x < 95$, $5 \le i_y < 95$, $5 \le i_t < 995$, and thus $N = 90^2 = 8100$, $M = 990$.

The PDE identification results are summarized in Table 3.1 which is explained in detail as the following. We build the dictionary $\mathbf{\Phi}_n \in \mathbb{R}^{990\times 9}$ for each location as (3.24), and build $\bar{\mathbf{\Phi}}_n^s$ using the terms and $\mathbf{s}$ in (3.24) according to (3.7a) for every $1 \le n \le N$. From (3.11), the coefficients for $\{\mathbf{u}_{tt}, \mathbf{u}_{xx}, \mathbf{u}_{yy}\}$ are found to have significant greater magnitudes, as shown in Fig. 3.13 (a) for the 70 kHz case as an example. We threshold $\bar{\mathbf{a}}_n(i)$ as (3.12a) using $\varepsilon = 0.2$ at each location that gives the active PDE terms. Counting the number of locations with the PDE terms correctly identified (either $\{\mathbf{u}_{tt}, \mathbf{u}_{xx}, \mathbf{u}_{yy}\}$ or these including $\mathbf{u}_t$) and normalizing it with N gives the "Success rate". The mean phase speed and mean absolute deviation (MAD, computed by $\frac{1}{N}\sum_{n=1}^{N} |\widehat{c}_n - (\frac{1}{N}\sum_{n=1}^{N}\widehat{c}_n)|$ where $\widehat{c}_n$ the recovered phase speed at location n) over these "success" locations are calculated. The recovered speeds $\widehat{c}_n = \sqrt{-\frac{\bar{\mathbf{a}}_n(5)+\bar{\mathbf{a}}_n(6)}{2\bar{\mathbf{a}}_n(2)}}$ are shown in Fig. 3.14. For each location in the region where the wave equation is not identified (3.2% of the ROI at most, at 70 kHz), the recovered $\widehat{c}$ is interpolated using the median value within a window covering 21 locations along y-axis centered at it.

The mean phase speeds in Table 3.1 are close to the recovered phase speeds when the PDE coefficients are assumed constant across the space and recovered from one dictionary built

Table 3.1. Success rate (using $\mathbf{s}$ as (3.24) or $\mathbf{s} = \mathbf{1}$ in $\mathbf{\Phi}_n^s$) of the PDE identification and the recovered speeds for various frequency bands.

Center Freq.(kHz)	30	40	50	60	70
Success rate using correct $\mathbf{s}$ in (3.24) (%)	99.5	98.1	99.1	99.4	96.8
Success rate using $\mathbf{s} = \mathbf{1}$ (%)	0	0	0	0	0
Mean speed (m/s)	467	532	586	628	663
MAD (m/s)	4.9	8.7	6.9	7.7	6.1
MAD/Mean (%)	1.1	1.6	1.2	1.2	0.9
$\widehat{c}_{all}$ (m/s)	463	531	587	632	668

from the measurements at all locations (a big dictionary $\mathbf{\Phi} = [\mathbf{\Phi}_1^{\mathrm{T}}\ \mathbf{\Phi}_2^{\mathrm{T}}\ \ldots\ \mathbf{\Phi}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{NM\times D}$) in Ref. [3], as given in $\widehat{c}_{all}$. The relative MAD is $\sim 1\%$, indicating the recovered speed is nearly a constant across the plate for a narrow frequency band, which coincides with our physical setting.

The correct $\mathbf{s}$ selected from physical knowledge is a key for the successful identification. If we use $\mathbf{1}$ as $\mathbf{s}$ in (3.24), no wave equations are identified at any location for any frequency band. Using $\mathbf{1}$, the $\bar{\mathbf{a}}_n$ for the band centered at 70 kHz is shown in Fig. 3.13 (b). The $\mathbf{1}$ used as $\mathbf{s}$ indicates the coefficients for U_{tt}, U_{xx} and U_{yy} have the same sign. Since this is not true, these terms are suppressed (comparing Fig. 3.13 (b) with Fig. 3.13 (a)), and the energy that should appear in $\bar{\mathbf{a}}_n(2), \bar{\mathbf{a}}_n(5)$ and $\bar{\mathbf{a}}_n(6)$ is redistributed to other entries to make the combination of selected columns in $\bar{\mathbf{\Phi}}_n^s$ still fit $\mathbf{e}$. This can not be remedied by tuning ε since the incorrect terms have larger coefficients than correct ones.

Table 3.2. PDE identification success rate with or without integration transformation for the noisy narrowband signal at 30 kHz. Values in the 2nd column is the direct quotient, not percentage.

Noise Variance	Noise/Signal Variance	Without Integration (%)	With Integration (%)
10^4	65.56	2.2	9.2
10^3	6.56	0.3	62.1
10^2	0.66	1.2	98.1
10^1	6.56×10^{-2}	25.3	99.5
10^0	6.56×10^{-3}	90.9	99.5

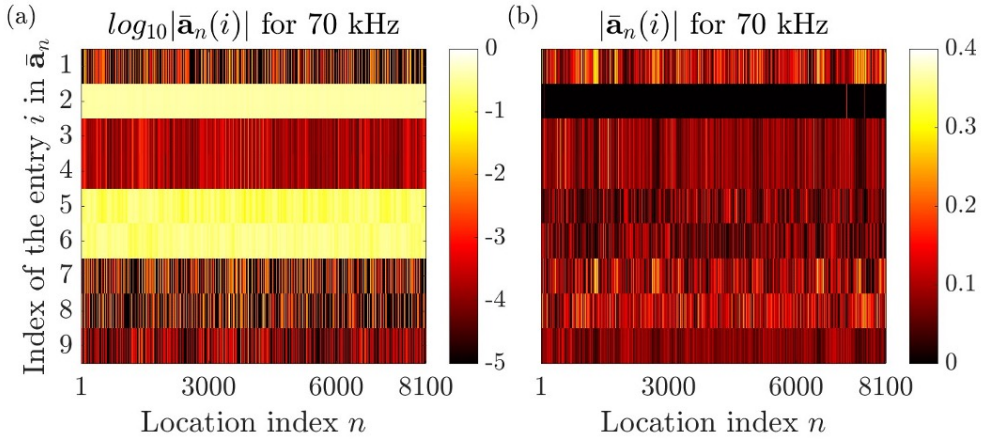

Figure 3.13. Magnitudes of $\bar{\mathbf{a}}_n(i)$ where i corresponds to the indices of columns for $\mathbf{\Phi}_n$ in (3.24) for all $n = 1 \ldots 90^2$ from clean measurements, (a) using $\mathbf{s}$ in (3.24); (b) using $\mathbf{1}$ as the $\mathbf{s}$.

Table 3.3. Success rate of the PDE identification from noisy measurements aided by integration transformation and the recovered speeds for various frequency bands. The correct $\mathbf{s}$ is from (3.24) and other rows in the dictionary are the normalized terms of (3.25).

Center Freq.(kHz)	30	40	50	60	70
Success rate using correct $\mathbf{s}$ (%)	98.1	99.3	99.2	93.4	96.6
Mean speed (m/s)	473	548	613	675	731
MAD (m/s)	5.8	4.2	5.5	9.6	5.4
MAD/Mean (%)	1.2	0.8	0.9	1.4	0.7
$\widehat{c}_{cl}$ (m/s)	476	556	610	667	722

3.4.2 Identification from noisy measurements

To test the robustness against AWGN, we add the AWGN with variance $\sigma^2 = 10^4, 10^3, 10^2$, 10 and 1 respectively to the signal with frequency band centered at 30 kHz (1 frame shown in Fig. 3.12), and identify PDEs from the dictionary (3.25) where the center of the integration domains starts from $i_t = 19$ and ends at $i_t = 979$ with $\delta = 20$ (so $M_{\text{int}} = 49$). The successful identification rate is in Table 3.2, in which the "Signal Variance" in the 2nd column is the average variance across all 1000 frames of the clean signal centered at 30 kHz in the ROI. Table 3.2 shows the integration method significantly increases the robustness against AWGN. In reality, the method should provide a high success rate for the sensors satisfying basic quality requirements. From Table 3.2 when the noise variance is 10^2, i.e. the noise variance is 66% of the signal

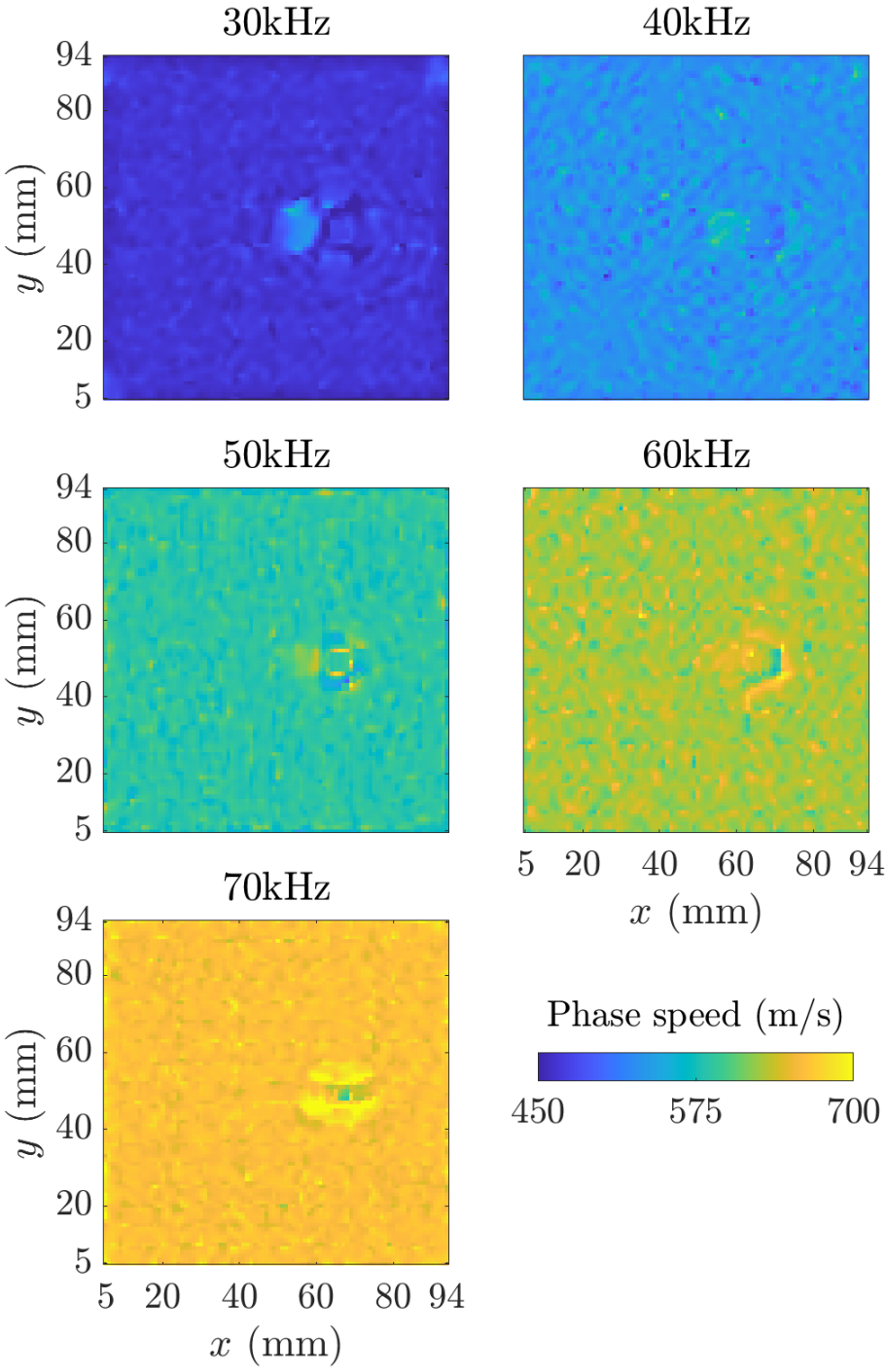

Figure 3.14. The recovered phase speeds for various frequency bands on the plate from clean measurements.

variance and thus a SNR $= 1.83$ dB, the success rate is larger than 98%.

Add AWGN with $\sigma^2 = 100$ to all 5 narrowband signals and use the integration transformation to assist PDE identification, the PDE coefficients are recovered with the example for 70 kHz shown in Fig. 3.15. The coefficients for $\{U_{tt}, U_{xx}, U_{yy}\}$ have larger coefficients at most locations and thus can be extracted by thresholding. Use $\varepsilon = 0.2$ to extract active terms and the recovered speeds $\widehat{c}_n = \sqrt{-\frac{\tilde{\mathbf{a}}_n(5)+\tilde{\mathbf{a}}_n(6)}{2\tilde{\mathbf{a}}_n(2)}}$ are shown in Fig. 3.16 and recorded in Table 3.3. For each location in the region where the wave equation is not identified (6.6% of the ROI at most, at 60 kHz), the recovered $\widehat{c}$ is interpolated using the median value within a window covering 21 locations along y-axis centered at it.

The mean speeds in Table 3.1 for high frequencies are underestimated because of insufficient sampling along time [3]. For 70 kHz, if we assume the speed being 700 m/s, the wavelength becomes 10^{-2} m $= 10\Delta x$, while the period is 1.43×10^{-5} s $\approx 4\Delta t$. Since finite difference evaluates $\partial_{tt}\mathbf{U}(i_x, i_y, i_t)$ based on slopes of the line segments connecting $\mathbf{U}(i_x, i_y, i_t)$ with $\mathbf{U}(i_x, i_y, i_t - 1)$ and $\mathbf{U}(i_x, i_y, i_t + 1)$ respectively, when Δt is not sufficiently small, these slopes can be far from the slope of the tangent line passing $\mathbf{U}(i_x, i_y, i_t)$, causing significant bias. In comparison, the mean speeds in Table 3.3 are not underestimated because of the 9 points interpolation between each neighboring time steps. The result is similar to a classic phase speed estimation based on Fourier transform by first finding the primary spatial frequency $\widehat{\xi}$ for each frequency f, and then $\widehat{c}_{cl} = f/\widehat{\xi}$ [38, 39] assuming the PDE is spatially independent wave equation and the waves are isotropic. The $\widehat{c}_{cl}$ computed from clean measurements is also shown in Table 3.3, indicating that the surface wave is strongly dispersive.

3.5 Efficiency

We emphasize the efficiency of the proposed method by comparing its CPU time to the PDE identification using sparse Bayesian learning (SBL) [14], cross-validation (CV) based method[3] and exhaustive search (Exhaust). Since methods in Refs. [14, 3] only work for

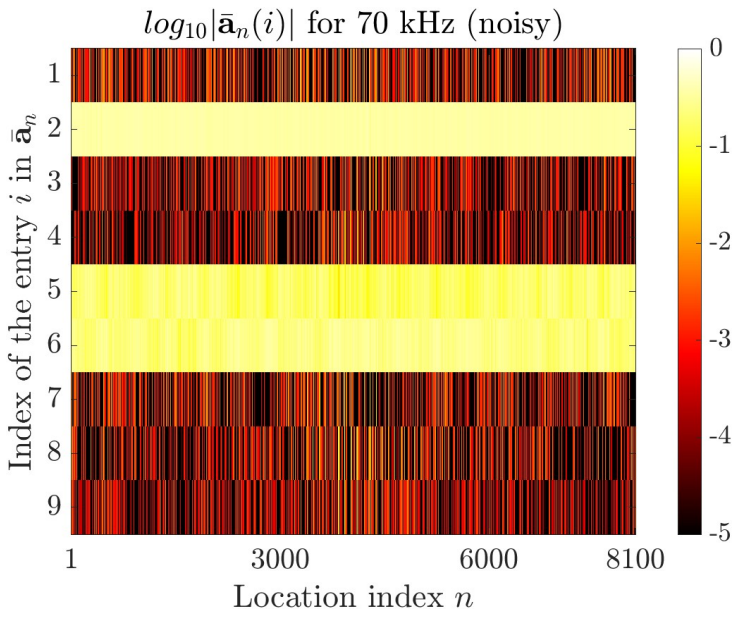

Figure 3.15. From noisy measurements of the vibrating plate centered at 70 kHz, $|\bar{\mathbf{a}}_n(i)|$ where i corresponds to the indices of columns for $\boldsymbol{\Phi}_n^{\text{int}}$ in (3.25) for all $n = 1 \dots 90^2$.

spatially-independent PDEs, we use 3 datasets describing such PDEs for the experiments: (a) dataset in Sec. 3.3.1; (b) non-attenuating 2D wave equation (3.23) with $\alpha = 0, c = 2.5$ m/s; (c) attenuating 2D wave equation (3.23) with $\alpha = 0.025, c = 2.5$ m/s. All other settings for (b)–(c) are the same as in Sec. 3.3.2.

As in Refs. [14, 3], since the PDE is identical for all locations we concatenate $\boldsymbol{\Phi}_n$ (defined by (3.3) or (3.24)) for each column to build the dictionary

$$\boldsymbol{\Phi} = [\boldsymbol{\Phi}_1^{\mathrm{T}}, \boldsymbol{\Phi}_2^{\mathrm{T}}, \dots, \boldsymbol{\Phi}_N^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{NM \times D} \tag{3.26}$$

for Refs. [14, 3] and exhaustive search, and the proposed method utilizes normalized (3.26) appended by $\mathbf{s}$ as its last row.

All the methods successfully identify the PDEs, and the CPU time shows the superior efficiency of our proposed method as demonstrated in Table 3.4. From Table 3.4, the proposed method outperforms others significantly. The exhaustive search is efficient for the dictionary with 6 terms, but time-consuming for 9 terms.

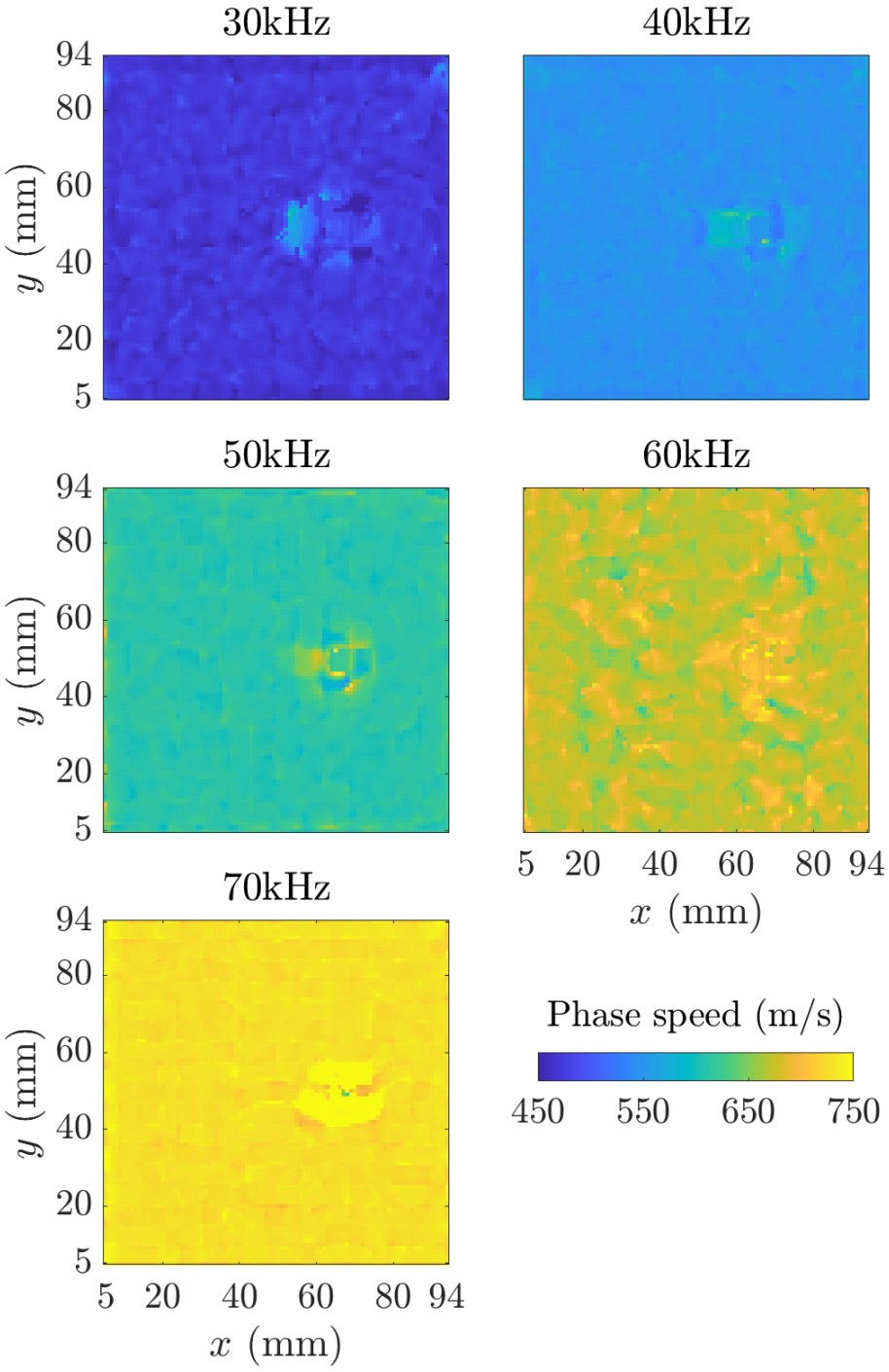

Figure 3.16. The recovered phase speeds for various frequency bands on the plate from noisy measurements.

Table 3.4. The average CPU time (s) for 10 trials on a MacBook Pro to correctly identify spatially independent PDEs on 3 datasets using sparse Bayesian learning (SBL) [14], the cross-validation (CV) based method[3], exhaustive search (Exhaust), and the proposed method.

	SBL[14]	CV[3]	Exhaust	Proposed
Burgers' eq.	0.69	0.16	0.06	0.002
Non-atten. wave eq.	1.7	15	35	0.028
Atten. wave eq.	1.0	4.3	9.4	0.010

Besides efficiency, the proposed method and the baselines in Table 3.4 have similar performance for correctly identifying the PDEs. With the same success rate, the higher efficiency makes our approach suitable for recovering spatially-dependent PDEs in a large ROI, in which the PDE identification is repeated for every spatial location.

3.6 Conclusion

We proposed a technique to efficiently recover the spatial variations of physical properties via spatially-dependent PDEs identification given observations, and validated it by recovering various acoustical properties for the medium of propagating waves.

The identification employs a constrained ℓ_1-norm minimization, which encourages sparsity and is solved via Lasso, to select active PDE terms from a dictionary of all potential terms. It is computationally efficient due to not requiring iterative assumptions of active PDE terms and the implementation of a fast computing scheme for Lasso. Using an integration transformation to transfer the derivatives on the noisy measurement to a smooth pre-defined function, the method can also identify spatially-dependent PDEs from highly noisy measurements.

3.7 Acknowledgments

The text of Chapter 3, in full, is a reprint of the material as it appears in R. Liu, P. Gerstoft, M. Bianco, B. D. Rao, "Recovery of Spatially Varying Acoustical Properties via Automated PDE Identification", The Journal of the Acoustical Society of America, 153(6), 3169-3180, 2023. The dissertation author was the primary researcher and author of Chapter 3. The coauthors listed

in this publication directed and supervised the research.

Chapter 4

SD-PINN: Deep Learning based Spatially Dependent PDEs Recovery

4.1 Introduction

Lots of natural phenomena find their mathematical representation in partial differential equations (PDEs), which are inherently composed of multiple terms and coefficients. A PDE describing the dynamics of field U can be written as

$$\mathsf{N}[U] = a_1 U_x + a_2 U_y + a_3 U_t + a_4 U_{tt} + \dots \tag{4.1}$$

where the partial derivatives $U_x, U_y, U_t, \dots$ are the PDE terms and the $a_1, a_2, \dots$ are PDE coefficients. The coefficients are often related to the physical properties of the medium and thus are of great interest in many applications. For example, in mechanical vibrations, the coefficients in the wave equation are related to the elastic properties of the medium [63]; in electromagnetics, the coefficients in Maxwell's equations are related to the electrical properties of the medium [64]. The spatial variation of the physical properties, like the various elasticities due to the various densities of the medium at different locations, leads to spatially-dependent PDE coefficients (e.g., in (4.1), the coefficients become $a_1(x,y), a_2(x,y)$, etc.). Thus by recovering the spatially-dependent PDE coefficients from observations (i.e., measurements of the dynamical field), we can obtain the spatial distribution of the physical properties of the medium.

The recent developments in computing power have enabled data-driven approaches to identify the PDEs directly from measurements [7, 5, 6, 15, 1, 16, 17, 18, 11, 67, 27, 14, 3, 65, 66]. Within these methods, the Physics Informed Neural Network (PINN) [5, 6, 7] has garnered considerable scholarly interest due to its notable resilience against measurement noise. Given the type of PDE which delineates the active PDE terms, PINN can learn the representation of the function mapping the spatiotemporal coordinate $(\boldsymbol{x}_m, t_j)$ (where for spatially 2D cases $\boldsymbol{x}_m$ is a vector) to its measurement u_{mj} by a fully connected feed-forward neural network (FNN) [68][69] and recover the PDE coefficients. However, the PINN has limitations when the coefficients for the PDEs are spatially dependent, as it assumes the coefficients are identical across the whole region of interest (ROI).

We propose a Spatially Dependent Physics Informed Neural Network (SD-PINN) which can recover spatially-dependent PDEs using only one neural network (in contrast to previous works which use two networks, one for PDE coefficients recovery and the other for solving the PDE [70, 80, 81]), without requiring domain-specific physical knowledge (in contrast to the prior arts which employ prior knowledge for specific situations, e.g., the stress-strain relationship [70]), and offers improved robustness against noise in input signals compared to previous methods [4]. Meanwhile, exploiting the low-rank assumption of the spatial variations for the PDE coefficients, the method can recover the coefficients at locations without available signals. A preliminary version of this work was in the conference paper [72].

Notations: The 2D or 3D matrices are given in bold capitalized letters, the vectors are in bold lowercase letters, and the scalars are in plain letters. For any variable $\mathbf{X}$ (or $\boldsymbol{x}$, x), its estimation is denoted by $\widehat{\mathbf{X}}$ (or $\widehat{\boldsymbol{x}}$, $\widehat{x}$). The entry at the ith row and jth column of matrix $\mathbf{X}$ is denoted by $\mathbf{X}(i,j)$, and $\mathbf{X}^{\mathrm{T}}(i,j)$ denotes the entry at the ith row and jth column of $\mathbf{X}^{\mathrm{T}}$ (the transpose of $\mathbf{X}$). $\mathscr{P}_{\Omega}(\mathbf{X})$ denotes the span of matrices vanishing outside a region Ω so that the (i,j)th component of $\mathscr{P}_{\Omega}(\mathbf{X})$ equals to $\mathbf{X}(i,j)$ if $(i,j) \in \Omega$ and zero otherwise. The number of entries within Ω is denoted by $|\Omega|$.

4.2 Theory of SD-PINN

With the type of PDE governing the field of interesting dynamics U in the ROI (with M spatial locations and T time steps) assumed known, we recover the spatially dependent coefficients for each term in the assumed PDE within the ROI. There are true PDE coefficients at only a few locations in the ROI given, the coefficients at all other locations, which consist the majority of the ROI, are recovered from the measurements of U.

The sign information (non-positive or non-negative) of each coefficient is known from the assumed type of PDE, which is determined by the physical background of the PDE and is the same at all locations. For example, in the wave equation [73]

$$U_{tt} + \alpha U_t - c^2 \nabla^2 U = 0 \tag{4.2}$$

the coefficient $-c^2$ for $\nabla^2 U$ (the Laplacian of U, i.e., $U_{xx} + U_{yy}$) must be non-positive since c is a real number for the phase speed of the wave, and α which is the factor for attenuation must be non-negative for a system without input energy from external sources.

In an overview of this work, an FNN denoted by a function Net_θ used to predict the observation $\widehat{u}_{mj}$ given its coordinates $(\boldsymbol{x}_m, t_j)$ as Fig. 4.1 is trained, where θ is the parameters (weights and bias) of the FNN. Then PDE terms (i.e., partial derivatives) are computed by automatic differentiation of Net_θ. The spatially-dependent PDE coefficients are then recovered using these partial derivatives computed at various locations. The details are described below.

4.2.1 Formulation of spatially-dependent PDEs

We focus on time-invariant homogeneous PDEs, i.e., there is no source in the ROI and the coefficients do not change with time.

The PDE is written with one term on the left-hand side (LHS) equaling other terms on the right-hand side (RHS). The coefficient of the one term in the LHS is set to one at every location,

e.g., for (4.2),

$$U_{tt} = -\alpha U_t + c^2 \nabla^2 U \ . \tag{4.3}$$

Our task is to recover the coefficients for all terms in the RHS for all locations.

We denote the LHS at the location $\boldsymbol{x}_m$ and time step t_j by ℓ_m^j. The RHS r_m^j contains K terms r_{mk}^j, each of which is a product of a time-invariant coefficient λ_{mk} and a PDE term d_{mk}^j. So the LHS equaling RHS gives:

$$\ell_m^j = r_m^j = \sum_{k=1}^{K} r_{mk}^j = \sum_{k=1}^{K} \lambda_{mk} d_{mk}^j \ . \tag{4.4}$$

For example, the wave equation (4.3) is rewritten as

$$(U_{tt})_m^j = -\alpha_m (U_t)_m^j + c_m^2 (\nabla^2 U)_m^j = \sum_{k=1}^{K} \lambda_{mk} d_{mk}^j \tag{4.5}$$

where $K = 2$, $\lambda_{m1} = -\alpha_m$, $\lambda_{m2} = c_m^2$, $d_{m1}^j = (U_t)_m^j$ and $d_{m2}^j = (\nabla^2 U)_m^j = (U_{xx})_m^j + (U_{yy})_m^j$. Thus the PDEs at all locations and time steps are written as

$$\ell_m^j = r_m^j = \sum_{k=1}^{K} \lambda_{mk} d_{mk}^j \ , \ \forall m, \ \forall j \ . \tag{4.6}$$

From (4.4), we can write the RHS for all the locations and PDE terms at time t_j in a matrix as

$$\begin{bmatrix} r_{11}^j & \cdots & r_{1K}^j \\ r_{21}^j & \cdots & r_{2K}^j \\ \vdots & \cdots & \vdots \\ r_{M1}^j & \cdots & r_{MK}^j \end{bmatrix} = \begin{bmatrix} \lambda_{11} & \cdots & \lambda_{1K} \\ \lambda_{21} & \cdots & \lambda_{2K} \\ \vdots & \cdots & \vdots \\ \lambda_{M1} & \cdots & \lambda_{MK} \end{bmatrix} \circ \begin{bmatrix} d_{11}^j & \cdots & d_{1K}^j \\ d_{21}^j & \cdots & d_{2K}^j \\ \vdots & \cdots & \vdots \\ d_{M1}^j & \cdots & d_{MK}^j \end{bmatrix} \tag{4.7}$$

where the MK unknown λ_{mk} are the coefficients to be recovered and $\circ$ is the Hadamard product. This differs from the conventional PINN, where only a vector of coefficients $[\lambda_1, \ldots, \lambda_K]$ is recovered since the PDE is assumed to be spatially independent. The SD-PINN is demonstrated using the wave equation (4.5) as an example, but it works the same way for other PDEs.

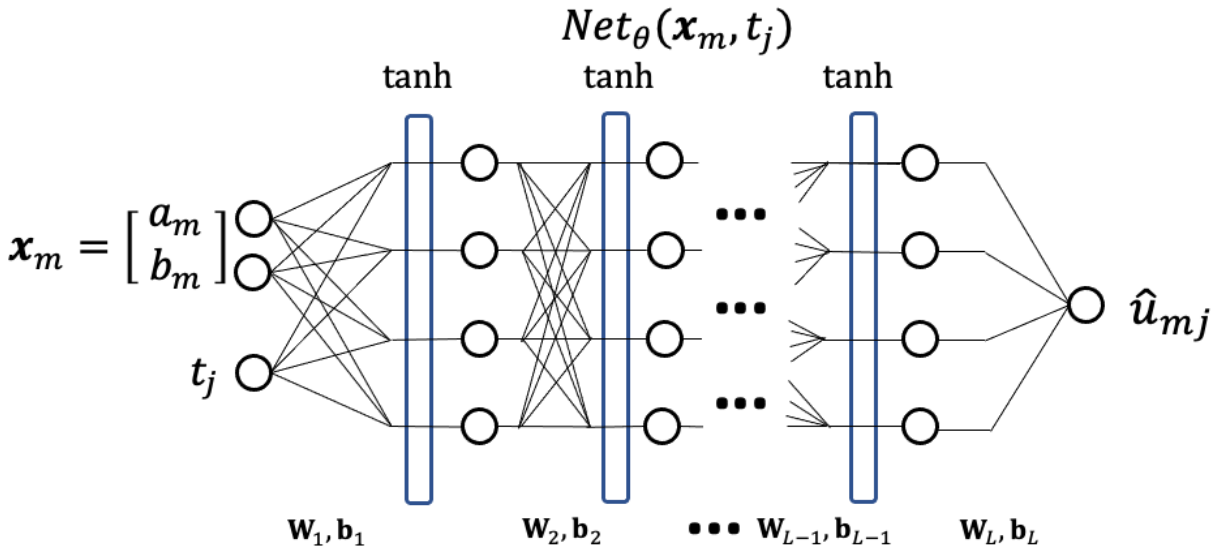

Figure 4.1. The FNN used in this work is denoted by a function $\widehat{u}_{mj} = Net_\theta(\boldsymbol{x}_m, t_j)$.

4.2.2 Low-rank assumption for the spatial variation of coefficients

In this work, we consider spatially 2D cases and assume the ROI to be a rectangular area with $M = M_1 M_2$, thus the measurements of the dynamical field acquired at T time steps are stored in a 3D matrix $\mathbf{U} \in \mathbb{R}^{M_1 \times M_2 \times T}$. By reshaping the $M \times 1$ vector for the coefficients of the kth term in (4.7) into an $M_1 \times M_2$ matrix and moving the index of the PDE term k to the 3rd dimension, the coefficients will be stored in a 3D matrix containing K slices in $\mathbb{R}^{M_1 \times M_2}$ with its kth slice being the spatially-dependent coefficients of the kth PDE term denoted as:

$$\mathbf{\Lambda}_k = \begin{bmatrix} \lambda_{11k} & \cdots & \lambda_{1M_2k} \\ \lambda_{21k} & \cdots & \lambda_{2M_2k} \\ \vdots & \cdots & \vdots \\ \lambda_{M_1 1k} & \cdots & \lambda_{M_1 M_2 k} \end{bmatrix}. \tag{4.8}$$

The $\boldsymbol{x}_m$ represents the mth location in the ROI and is a vector containing a row index and a column index. The objective of SD-PINN is to find an estimation $\widehat{\mathbf{\Lambda}}_k$ for all entries of $\mathbf{\Lambda}_k$ for all k based on a few given entries.

In real-world scenarios, the spatial variations of the physical properties of the medium for the dynamics are not random, as the properties at a certain point are influenced by the surrounding medium. The decreased degrees of freedom are represented by a lower rank for $\mathbf{\Lambda}_k$, which is smaller than $\min(M_1, M_2)$.

For $\mathbf{\Lambda}_k \in \mathbb{R}^{M_1 \times M_2}$ with rank r_k, there exist two smaller matrices with r_k columns whose multiplication is $\mathbf{\Lambda}_k$[74]:

$$\mathbf{\Lambda}_k = \mathscr{U}_k \mathscr{V}_k^{\mathrm{T}} \tag{4.9}$$

where $\mathscr{U}_k \in \mathbb{R}^{M_1 \times r_k}$ and $\mathscr{V}_k \in \mathbb{R}^{M_2 \times r_k}$.

The method aims to find $\widehat{\mathscr{U}}_k \in \mathbb{R}^{M_1 \times r_k}$ and $\widehat{\mathscr{V}}_k \in \mathbb{R}^{M_2 \times r_k}$ for all k which satisfy $\widehat{\mathscr{U}}_k \widehat{\mathscr{V}}_k^{\mathrm{T}} = \widehat{\mathbf{\Lambda}}_k$, such that $\widehat{\mathbf{\Lambda}}_k \approx \mathbf{\Lambda}_k$ and specifically $\mathscr{P}_\Omega(\widehat{\mathbf{\Lambda}}_k) = \mathscr{P}_\Omega(\mathbf{\Lambda}_k)$ by exploiting the information from measurements $\mathbf{U}$ of the dynamical field governed by PDEs parameterized by $\{\mathbf{\Lambda}_k | \forall k = 1, \ldots, K\}$, where Ω covers the few locations for the given coefficients.

Instead of assuming the exact rank of $\mathbf{\Lambda}_k$, we assume a reasonable upper limit for that and use it as r_k, which not only represents a weaker assumption that is empirically viable but also provides a better recovery as detailed in the following sections. Since the rank of $\mathbf{\Lambda}_k$ can be smaller than r_k, the column vectors in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ are not necessarily linear independent.

By denoting $\widehat{\mathbf{\Lambda}}_k = \widehat{\mathscr{U}}_k \widehat{\mathscr{V}}_k^{\mathrm{T}}$ we relate the entries in $\widehat{\mathbf{\Lambda}}_k$ by the vectors in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$. We thus decrease the number of unknowns to be recovered from KM_1M_2 to $\sum_k (M_1 + M_2) r_k$ and can use the measurements from only a part of the ROI to recover the properties in the whole ROI. This is valuable when the sensors are insufficient, or there are areas within the ROI where sensors can not be placed.

4.2.3 Loss functions

The used neural network Net_θ parametrized by θ is an FNN with L layers as shown in Fig. 4.1, whose inputs are the spatial-temporal coordinates $(\boldsymbol{x}_m, t_j)$ where $\boldsymbol{x}_m = [a_m, b_m]^{\mathrm{T}}$ is

a vector describing the location indexed by m in the ROI, and outputs are the corresponding estimated measurements $\widehat{u}_{mj}$. During the training of the SD-PINN, we minimize the overall loss $loss$ as expressed in Eq. (4.10):

$$loss = loss_{\mathrm{u}} + w_{\mathrm{f}} \times loss_{\mathrm{f}} + w_{\mathrm{g}} \times loss_{\mathrm{g}} + w_{\mathrm{si}} \times loss_{\mathrm{si}} \ , \tag{4.10}$$

which is a linear combination of four individual losses: $loss_{\mathrm{u}}, loss_{\mathrm{f}}, loss_{\mathrm{g}}, loss_{\mathrm{si}}$ with their weights being $1, w_{\mathrm{f}}, w_{\mathrm{g}}$ and w_{si} respectively. These losses can be classified into three categories: (i) the data fitting loss $loss_{\mathrm{u}}$ is a function of only the neural network parameters θ (weights and bias); (ii) the functional loss $loss_{\mathrm{f}}$ is a function of both θ and the PDE coefficients λ (which stands for all entries subjected to recovery in $\mathbf{\Lambda}_k, \ \forall k$); and (iii) the given coefficients loss $loss_{\mathrm{g}}$ and sign loss $loss_{\mathrm{si}}$ are functions of only the PDE coefficients λ.

The $loss$ (4.10) is minimized via Adam [79]. At the beginning of the network training, all entries in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ for all k are randomly initialized together with θ. The details of these losses are provided below, in which the $\widehat{\mathbf{\Lambda}}_k$ is an intermediate variable and during training the gradients are used to update the $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ essentially. In the optimization related to $\widehat{\mathbf{\Lambda}}_k$ (which involves Sec. 4.2.3, 4.2.3, 4.2.3), we do not include the substitution of $\widehat{\mathbf{\Lambda}}_k = \widehat{\mathscr{U}}_k \widehat{\mathscr{V}}_k^{\mathrm{T}}$ to maintain concise formulaic representation.

Data fitting loss

Given the training samples $\{\boldsymbol{x}_m, t_j, u_{mj}\}$ selected from measurements $\mathbf{U}$, the FNN Net_θ adjusts its parameters (wights and bias) θ to learn the mapping from coordinates $(\boldsymbol{x}_m, t_j)$ to its corresponding measurement u_{mj} by minimizing the $loss_{\mathrm{u}}$:

$$loss_{\mathrm{u}}(\theta) = \sum_{\boldsymbol{x}_m \in \Omega_{\mathrm{u}}} \sum_{j=1}^{T} (Net_\theta(\boldsymbol{x}_m, t_j) - u_{mj})^2 \tag{4.11}$$

where Ω_{u} the set of locations where the measurements are used as training samples to minimize $loss_{\mathrm{u}}$.

Functional loss

After $\widehat{u}_{mj}$ is computed by $Net_\theta(\boldsymbol{x}_m, t_j)$, we compute the PDE terms $\widehat{\ell}_m^j$ and $\widehat{d}_{mk}^j$ by automatic differentiation [8]. For example, the $(U_t)_m^j$ is computed as $\left.\frac{\partial Net_\theta(\boldsymbol{x},t)}{\partial t}\right|_{\boldsymbol{x}=\boldsymbol{x}_m, t=t_j}$, which is a function of $(\boldsymbol{x},t)$ parametrized by θ. It can also be deemed as a function of θ parametrized by $\{\boldsymbol{x}=\boldsymbol{x}_m, t=t_j\}$ if we want to optimize θ using it, and thus $\{\widehat{\ell}_m^j, \widehat{d}_{mk}^j\}$ can be written as $\{\widehat{\ell}_m^j(\theta), \widehat{d}_{mk}^j(\theta)\}$.

The computation of $\widehat{\ell}_m^j$ and $\widehat{d}_{mk}^j$ allows us to introduce $loss_{\mathrm{f}}$, by minimizing which we recover the PDE coefficients λ and prevent the Net_θ from overfitting the measurements when there is noise in the training samples. The $loss_{\mathrm{f}}$ is

$$\begin{aligned} loss_{\mathrm{f}}(\theta,\lambda) &= \sum_{j\in I_{\mathrm{t}}} \sum_{m\in I_{\mathrm{m}}} (\widehat{\ell}_m^j(\theta) - (\sum_{k=1}^{K} \widehat{\lambda}_{mk} \widehat{d}_{mk}^j(\theta)))^2 \\ &= \sum_{t\in I_t} \|\widehat{\mathbf{L}}^t(\theta) - \sum_k \widehat{\mathbf{\Lambda}}_k \circ \widehat{\mathbf{D}}_k^t(\theta)\|_F^2 \end{aligned} \tag{4.12}$$

with

$$\widehat{\mathbf{L}}^t(\theta) = \begin{bmatrix} \widehat{\ell}_{11}^t(\theta) & \cdots & \widehat{\ell}_{1M_2}^t(\theta) \\ \vdots & \cdots & \vdots \\ \widehat{\ell}_{M_1 1}^t(\theta) & \cdots & \widehat{\ell}_{M_1 M_2}^t(\theta) \end{bmatrix}, \tag{4.13}$$

$$\widehat{\mathbf{D}}_k^t(\theta) = \begin{bmatrix} \widehat{d}_{11k}^t(\theta) & \cdots & \widehat{d}_{1M_2k}^t(\theta) \\ \vdots & \cdots & \vdots \\ \widehat{d}_{M_1 1k}^t(\theta) & \cdots & \widehat{d}_{M_1 M_2 k}^t(\theta) \end{bmatrix} \tag{4.14}$$

where I_{m} is the set of location indices m corresponding to all $\boldsymbol{x}_m$ used in $loss_{\mathrm{f}}$. As indicated by (4.14), I_m covers all $M_1 M_2$ locations within the ROI for our experiments. The I_{t} is the set of time steps used for $loss_{\mathrm{f}}$, and is chosen as all time steps from 1 to T.

In addition to recovering PDE coefficients λ, the $loss_{\mathrm{f}}$ also benefits the training of neural network parameters θ by encouraging Net_θ to provide the correct partial derivatives as the PDE

terms. If we only use the $loss_{\mathrm{u}}$ to train the network, although we can quickly make the neural network predict the dynamic field itself more accurately, the field's partial derivatives computed by automatic differentiation (AD) are not sufficiently accurate. This is because there are multiple neural network parameters θ that can make $\widehat{u}_{mj} = Net_{\theta}(\boldsymbol{x}_m, t_j)$ approximately equal to the true u_{mj}, but for different θ, the AD (for example, $\frac{\partial Net_{\theta}(\boldsymbol{x},t)}{\partial t}\Big|_{\boldsymbol{x}=\boldsymbol{x}_m, t=t_j}$) are different.

In addition to recovering the PDE coefficients λ, the $loss_{\mathrm{f}}$ also encourages θ to be the one that makes the AD of Net_{θ} work well as the PDE terms. Without $loss_{\mathrm{f}}$, the AD based on the θ optimized purely on $loss_{\mathrm{u}}$ can not simulate the true partial differentiation of U.

Given coefficients loss

Let there be p_k entries in $\boldsymbol{\Lambda}_k$ from a sub-region Ω of the ROI known *a priori*, we thus have $loss_{\mathrm{g}}$ (where g stands for "given") as

$$loss_{\mathrm{g}}(\lambda) = \sum_{k} \sum_{(a,b)\in\Omega} (\widehat{\boldsymbol{\Lambda}}_k(a,b) - \boldsymbol{\Lambda}_k(a,b))^2 \tag{4.15}$$

where a and b are the row and column indices to enforce all entries within Ω to be identical between the recovered $\widehat{\boldsymbol{\Lambda}}_k$ and true $\boldsymbol{\Lambda}_k$.

Sign loss

The sign (non-negative or non-positive) for the coefficients in the given type of the PDE is unchanged across the ROI. Thus we can encourage the recovered coefficients to have their assumed signs by minimizing the sign loss

$$loss_{\mathrm{si}}(\lambda) = \sum_{m=1}^{M} \sum_{k=1}^{K} \mathrm{ReLU}(-\mathrm{sign}(\lambda_{mk})\widehat{\lambda}_{mk}) \tag{4.16}$$

where ReLU is the Rectified Linear Unit defined as $\mathrm{ReLU}(x) = x$ for $x > 0$ and 0 otherwise, and $\mathrm{sign}(\lambda_{\mathrm{mk}})$ is 1 for $\lambda_{mk} > 0$ or -1 for $\lambda_{mk} < 0$ depending only on the assumed sign of true λ_{mk} and is irrelevant to its approximation $\widehat{\lambda}_{mk}$. Further, the $\mathrm{sign}(\lambda_{\mathrm{mk}})$ depends only on k because the

sign for a given PDE term is assumed the same in the PDE recovered at any location m. From (4.8), the sign loss (4.16) is rewritten as

$$loss_{\text{si}}(\lambda) = \sum_{k} \sum_{(a,b)\in \text{ROI}} \text{ReLU}(-\text{sign}(\mathbf{\Lambda}_k(a,b))\widehat{\mathbf{\Lambda}}_k(a,b)) \tag{4.17}$$

where the value of $\text{sign}(\mathbf{\Lambda}_k(a,b))$ is entirely determined by k.

For example, for the wave equation (4.3) where $\mathbf{\Lambda}_1$ denotes $-\alpha$ (non-positive) and $\mathbf{\Lambda}_2$ denotes c^2 (non-negative), $loss_{\text{si}}$ is

$$loss_{\text{si}}(\lambda) = \sum_{(a,b)\in \text{ROI}} \text{ReLU}(\widehat{\mathbf{\Lambda}}_1(a,b)) + \text{ReLU}(-\widehat{\mathbf{\Lambda}}_2(a,b)) \ . \tag{4.18}$$

Note that the $\mathbf{\Lambda}_k$ and $\widehat{\mathbf{\Lambda}}_k$ stand for both the magnitude of the coefficient and its assumed sign. For example, in (4.3), the $\mathbf{\Lambda}_1$ and $\widehat{\mathbf{\Lambda}}_1$ are for $-\alpha$ instead of α.

4.2.4 Coefficient recovery as a matrix completion problem

The spatially dependent PDE coefficients recovery can be performed as a matrix completion problem [74, 75, 76, 77, 78]. Assuming that for $\mathbf{\Lambda}_k$ there are p_k entries known with their spatial locations covered by Ω (a sub-region of the ROI), the goal of coefficients recovery is to reconstruct the matrix $\mathbf{\Lambda}_k$ from these known entries subject to the constraint $\text{rank}(\mathbf{\Lambda}_k) \leq r_k$.

We discuss two factors that affects the coefficients recovery: the locations of given coefficients and the number of columns of $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ (i.e., r_k).

Locations of given coefficients

For $\widehat{\mathscr{U}}_k \in \mathbb{R}^{M_1 \times r_k}$ and $\widehat{\mathscr{V}}_k \in \mathbb{R}^{M_2 \times r_k}$ subjected to recovery, the equation $\widehat{\mathscr{U}}_k \widehat{\mathscr{V}}_k^{\text{T}} = \widehat{\mathbf{\Lambda}}_k$ where $\widehat{\mathbf{\Lambda}}_k = \Lambda_k$ at p_k specified entries defines a collection of p_k equations with several variables

which are a part of the entries in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$:

$$\sum_{i=1}^{r_k} \widehat{\mathscr{U}}_k(a_j,i)\widehat{\mathscr{V}}_k^{\mathrm{T}}(i,b_j) = \widehat{\mathbf{\Lambda}}_k(a_j,b_j) = \mathbf{\Lambda}_k(a_j,b_j), \quad \forall (a_j,b_j) \in \Omega, \;\; j = 1,\ldots,p_k \tag{4.19}$$

for Ω with $|\Omega| = p_k$.

The number of entries of $\widehat{\mathscr{U}}_k$ involved in these equations is r_k times the number of distinct rows covered by Ω: for example, when $p_k = 2$, in (4.19), if $a_1 = a_2$, r_k entries of $\widehat{\mathscr{U}}_k$ are involved; otherwise, $2r_k$ entries are involved. Similarly, the number of entries in $\widehat{\mathscr{V}}_k$ involved is r_k multiplying the number of distinct columns covered by Ω. Thus, for a fixed number (i.e., p_k) of equations, the more distinct rows and columns covered by Ω, the more entries of $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ are affected by these p_k known coefficients. If the locations in Ω are concentrated in too few distinct rows and columns, the recovery is difficult because the contribution of the known coefficients is constrained within too few entries of $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$.

Redundant columns of $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$

If the specified r_k which is the number of columns in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ exceeds the true rank of $\mathbf{\Lambda}_k$ (denoted by r_k^0), this will be an advantage because more degrees of freedom are allowed for the recovery. This is intuitive because when $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ have r_k columns, their ranks can be smaller or equal to r_k. Thus, the potential $\widehat{\mathbf{\Lambda}}_k$ generated by $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ with more columns encompasses the $\widehat{\mathbf{\Lambda}}_k$ derived from $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ with fewer columns. In other words, the potential $\widehat{\mathbf{\Lambda}}_k$ recovered with a higher upper limit of its rank includes those recovered with a lower upper limit, but the reverse is not true.

Meanwhile, the coefficient recovery does not monotonically improve with the increase in the number of columns r_k. If r_k is too large, there are so many degrees of freedom for entries in $\widehat{\mathbf{\Lambda}}_k$ that the information of recovered entries at locations with available measurements is insufficient to confidently determine the values of entries at other locations.

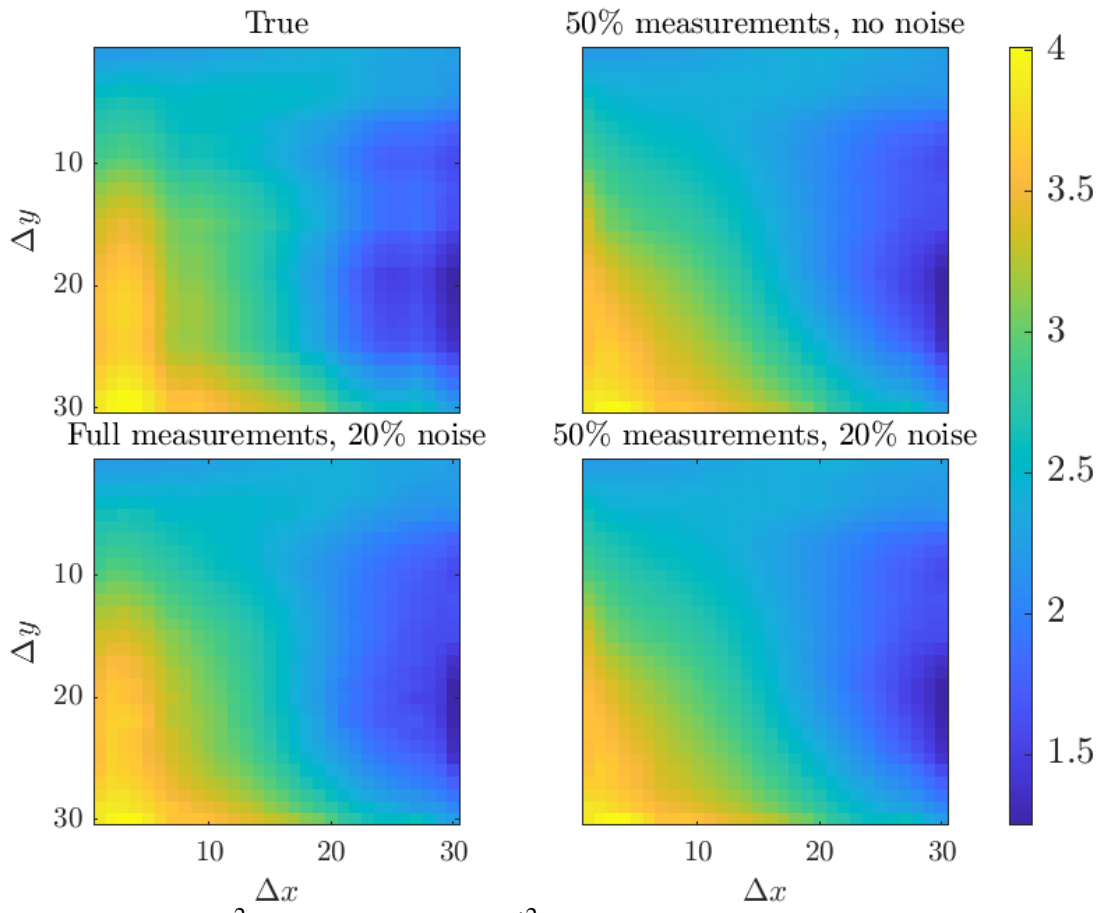

Figure 4.2. The true c^2 and some recovered $\widehat{c}^2$ in the experiments with $r_1 = 5$ (at epoch = 4000).

4.3 Experiments for SD-PINN

In this section, we explore the PDE coefficients recovery as a matrix completion problem. Various datasets **U** containing the measurements of wave fields governed by the wave equation (4.5) are used, all of which are in the shape of $\mathbb{R}^{30\times30\times198}$. The distances between neighboring coordinates are $\Delta x = \Delta y = 0.1$ m and $\Delta t = 0.01$ s. They may be noise-free or noisy and may be complete (i.e., all entries are given) or incomplete to various extents.

All data in **U** with their spatiotemporal coordinates are used to train Net_θ and $\{\widehat{\mathscr{U}}_k, \widehat{\mathscr{V}}_k\}$ are recovered while training. Thus, the input is the concatenation of $\boldsymbol{x}_m$ and t_j which is a three-component vector and the output is real number $\widehat{u}_{mj}$ which is used to be compared with the true observation u_{mj}.

Table 4.1. The root mean square error (RMSE) of recovered $\widehat{c}$ (at epoch = 4000) for various experiments using the waves without attenuation. The small number in the right bottom corner is the epoch at which the $\widehat{c}^2_m$ is extracted.

Signals	Noise level	RMSE$_{c^2}$ ($r_1 = 3$)	RMSE$_{c^2}$ ($r_1 = 5$)
All	0	0.140	0.128
All	10%	0.144	0.140
All	20%	0.132	0.131
50%	0	0.136	0.115
50%	10%	0.131	0.116
50%	20%	0.137	0.135

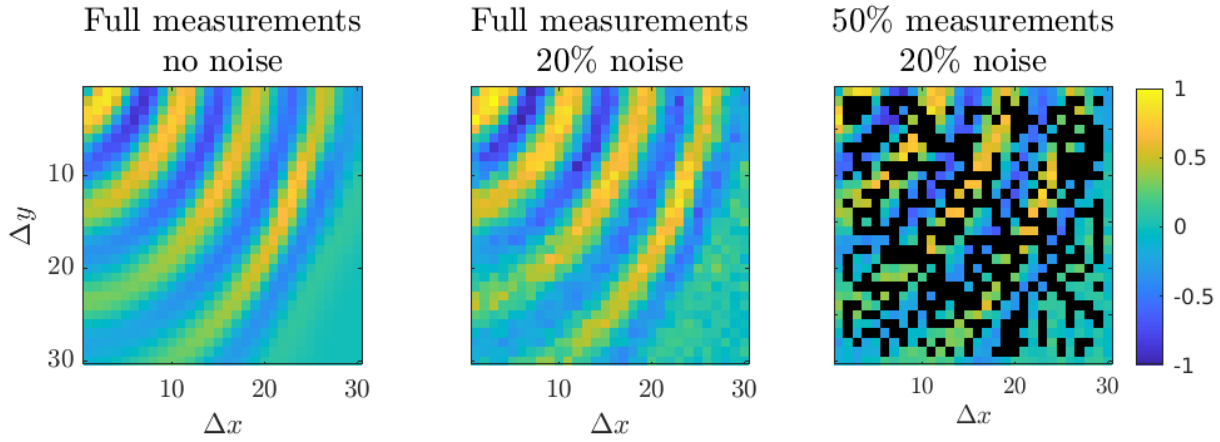

Figure 4.3. The clean and noisy signal at frame 100 with full measurements and 50% measurements. The black pixels denote places without available signals, which are randomly selected.

In all experiments, we use $L = 5$ layers for the FNN as shown in Fig. 4.1. The activation function tanh is applied in the 1∼4 layers. For the weights $\mathbf{W}_l$ of layer l, $\mathbf{W}_l \in \mathbb{R}^{200\times 200}$ for $2 \le l \le 4$, while $\mathbf{W}_1 \in \mathbb{R}^{200\times 3}$ and $\mathbf{W}_5 \in \mathbb{R}^{1\times 200}$ accommodate the input and output sizes. All weights are initialized by He initializer [82]. All biases $\boldsymbol{b}_l$ are in $\mathbb{R}^{200}$ except at the last layer where it is a scalar, and they are initialized as zero. The entries in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ are initialized as samples drawn from zero-mean Gaussian random distribution with a standard deviation of 0.1. We set $w_{\mathrm{f}} = 0.1, w_{\mathrm{si}} = w_{\mathrm{g}} = 1$ for (4.10). The Ω_{u} for $loss_{\mathrm{u}}$ in (4.11) is set to be all locations where measurements are available. As measurements at certain locations may be unavailable, the Ω_{u} is not necessarily the whole ROI. For (4.12), I_{m} is for all the M locations in the ROI, and I_{t} is for all T time steps.

Table 4.2. The RMSEs between the true and recovered PDE coefficients for different settings of locations for given coefficients.

Locations of given coefficients	RMSE_α	RMSE_{c^2}
Diagonal	1.466	0.328
Grid	4.228	2.444
Random	1.810	0.194

4.3.1 Non-attenuating waves

We recover the PDEs for non-attenuating waves described in $\mathbf{U} \in \mathbb{R}^{30\times30\times198}$ here. The PDE is the spatially-dependent wave equation (4.5) where $\alpha_m = 0$ and c_m^2 is distributed as the "True" subplot of Figure 4.2 (rank $= 3$). The unit for the phase speed c is m/s, and for the attenuation factor α is s^{-1}.

In this case, the only coefficient we are recovering is c_m^2 for all m, and thus $K = 1$, the only coefficient matrix $\mathbf{\Lambda}_1$ to be recovered in (4.8) is for the phase speeds and has a rank $r_1^0 = 3$. The PDE term corresponding to c_m^2 is $\nabla^2 U$, which is computed as the sum of U_{xx} and U_{yy}, both of which are computed by automatic differentiation of Net_θ. We carry out 12 experiments with no noise, 10% noise, 20% noise, all measurements available, 50% measurements available (see Figure 4.3), $r_1 = 3$ and $r_1 = 5$ respectively. The noise is additive zero-mean Gaussian noise. The "10%" or "20% noise" means the standard deviation (STD) of the Gaussian noise is 10% or 20% of the STD of the measurements. The "50% measurements available" means the available measurements are from all time steps and 50% spatial locations (randomly selected) of the unknown region in the ROI (i.e., the ROI excluding Ω where Ω are the four boundaries, denoted by Ω^c). We measure the recovery results by the root mean square error (RMSE) which is

$$\text{RMSE}_{c^2} = \sqrt{\frac{\sum_{m\in\Omega^c} |\widehat{c}_m^2 - c_m^2|^2}{|\Omega^c|}} \tag{4.20}$$

where $|\Omega^c|$ is the cardinality of set Ω^c and summarize them in Table 4.1. The results of the recovery for some experiments are shown in Figure 4.2. Both the RMSEs and the graphical

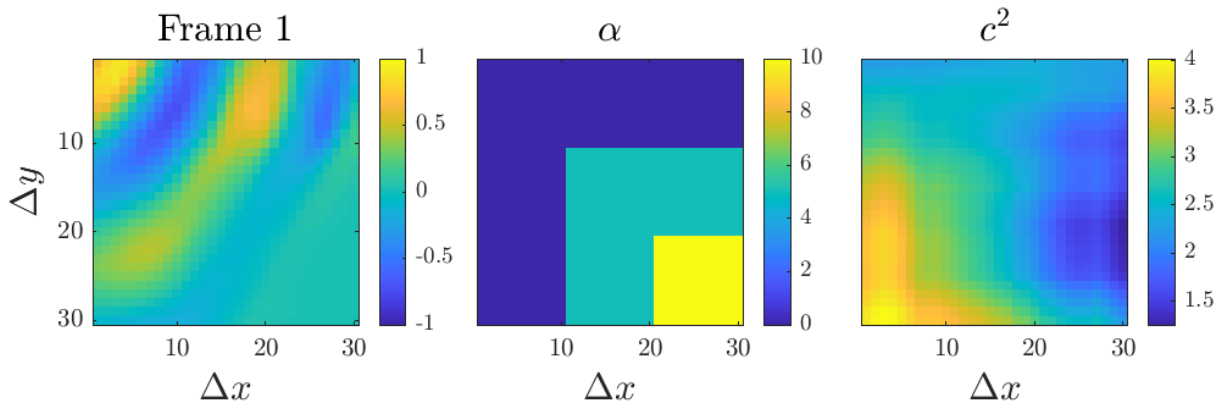

Figure 4.4. One frame of the wavefield for the attenuating wave with its true α and c^2, where $\max(\alpha) = 10$.

demonstrations show that the recovery is satisfactory.

4.3.2 Attenuating waves

Locations of given coefficients

We experiment with a dataset $\mathbf{U} \in \mathbb{R}^{30\times30\times198}$ showing an attenuating wavefield. One frame of the field together with its true α and c^2 is shown in Fig. 4.4. The spatial variation of c^2 is the same as the dataset in Sec. 4.3.1 so that its rank is 3, and the rank for α is 2. The PDE is the wave equation (4.5) and thus $K = 2$, the $\mathbf{\Lambda}_1$ is for $-\alpha$ and $\mathbf{\Lambda}_2$ is for c^2. The true rank for $\mathbf{\Lambda}_1$ is $r_1^0 = 2$ and for $\mathbf{\Lambda}_2$ is $r_2^0 = 3$. Although $\mathbf{\Lambda}_1$ stands for $-\alpha$, we show α in the subsequent figures as it more directly represents the physical properties of the medium.

For this wavefield, we first conduct three experiments with different settings of the locations for the given PDE coefficients. Unlike before, the coefficients on the boundaries are unknown here. In the overall 900 spacial locations within the ROI, the set of locations Ω for given coefficients covers 30 entries, which are on the diagonal, evenly spaced grids, and randomly selected locations respectively. We set $r_1 = 2, r_2 = 3$ to run the recovery, the same as true ranks. After 6000 epochs, the coefficients recovery results are summarized in Fig. 4.5. Compared to Fig. 4.4, it is visibly evident that the recovery of "diagonal" is approximately

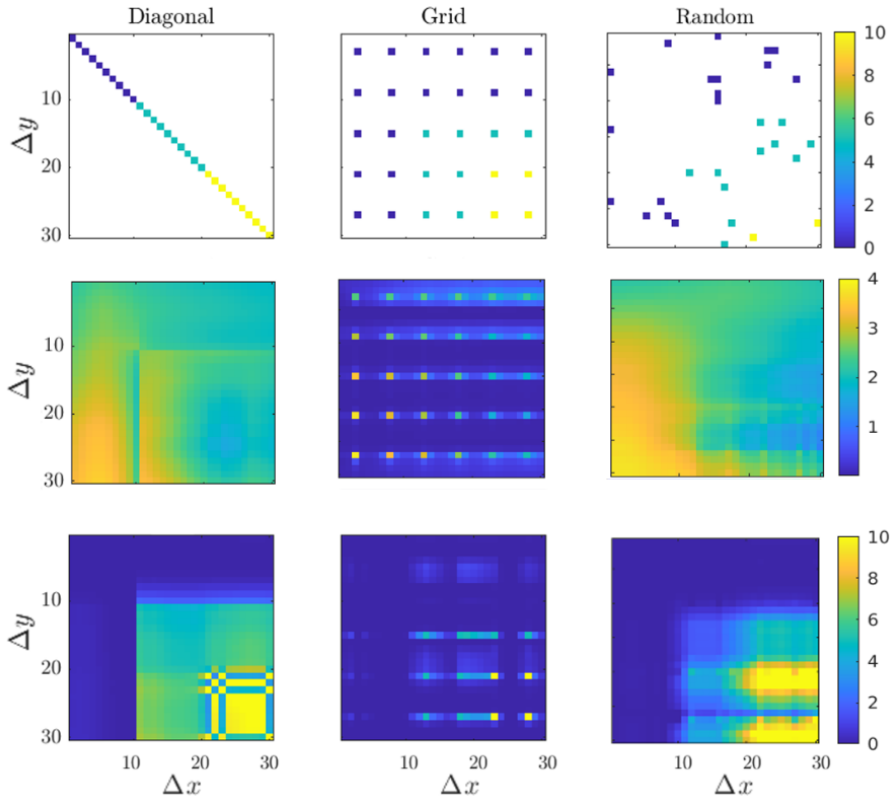

Figure 4.5. Recovery of PDE coefficients (at 6000th epoch) with 30 entries given: 1st row: the locations of the given entries with the colors representing true α (white pixels are for locations without given coefficients, i.e., Ω^c); 2nd row: recovered $\widehat{c^2}$; 3rd row: recovered $\widehat{\alpha}$.

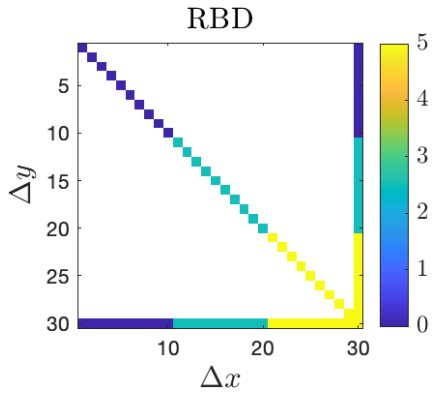

Figure 4.6. The locations of the given coefficients which include the right, bottom boundaries, and the diagonal (RBD). There are 88 locations in total. The colors indicate the α at these locations, and white pixels are for locations without given coefficients.

Table 4.3. RMSEs between true and recovered PDE coefficients for different settings of ranks, signal availability, and noise conditions.

Experimental settings	r_1	r_2	RMSE_α	RMSE_{c^2}
Full measurements no noise	5	5	0.366	0.145
75% measurements no noise	5	5	0.369	0.125
50% measurements no noise	5	5	0.371	0.140
50% measurements no noise	2	3	0.495	0.186
Full measurements 10% noise	5	5	0.359	0.153
75% measurements 10% noise	5	5	0.356	0.150
75% measurements no noise	2	3	0.496	0.192
75% measurements 10% noise	2	3	0.495	0.198
Full measurements 20% noise	5	5	0.400	0.134
50% measurements 20% noise	5	5	0.398	0.139

equivalent to "random", and both are significantly superior to "grid". The RMSEs between the true and recovered coefficients are given in Table 4.2, where RMSE_α is defined as

$$\text{RMSE}_\alpha = \sqrt{\frac{\sum_{m\in\Omega^c} |\widehat{\alpha}_m - \alpha_m|^2}{|\Omega^c|}} \tag{4.21}$$

The results coincide with our conjecture that when the locations of given coefficients are too concentrated in a few distinct rows and columns, the recovery is hard. For the "diagonal", "random" and "grid", the numbers of distinct rows where the coefficients are given are 30, 20, and 5; and the numbers of distinct columns are 30, 18, and 6 respectively.

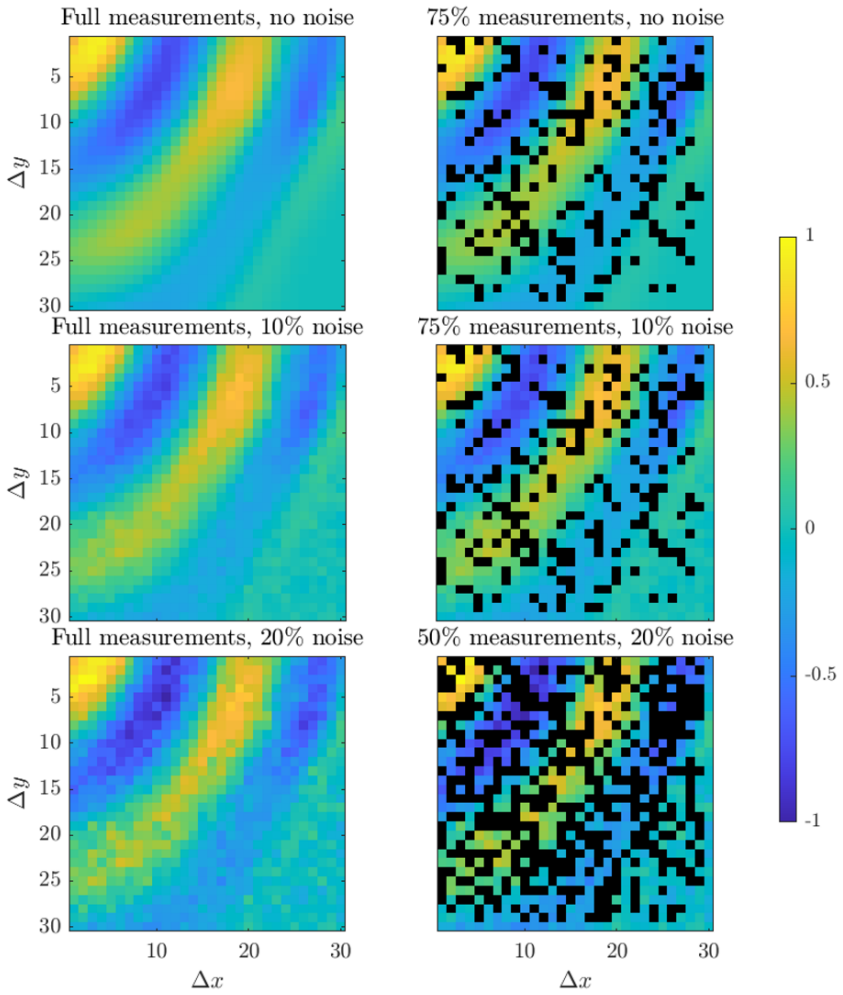

Figure 4.7. One frame of the wave field with various percentages of observations and noise levels. The randomly selected black pixels stand for locations without available measurements.

Redundant columns of $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$

In this section, we recover the PDE coefficients of an attenuating wavefield with one frame shown in Fig. 4.7 and the true coefficients shown in Fig. 4.8. For the coefficients, everything is the same as the dataset in 4.3.2 except that the attenuation is halved. There are $|\Omega| = 88$ locations of given coefficients on the right boundary + bottom boundary + diagonal (RBD) as shown in Fig. 4.6. We carry out 10 experiments where the measurements at all locations are available or at 75%, 50% locations are available, and the signal is noise-free or polluted by Gaussian noise whose STD is 10% or 20% of the signal's STD. The frame of the signals with various noise

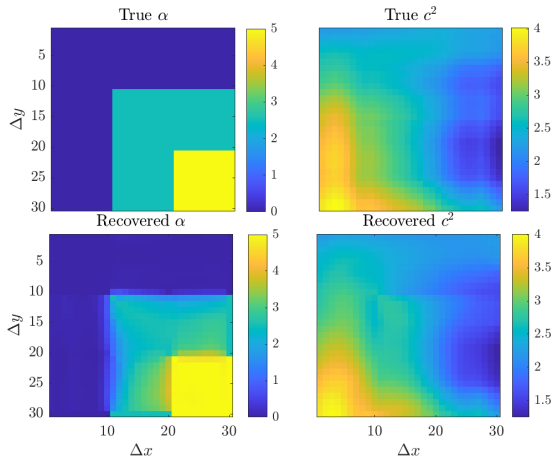

Figure 4.8. Recovered $\widehat{\alpha}$ and $\widehat{c^2}$ at epoch 5000 given 88 entries on the right, bottom and diagonal for the ground truth ($r_1^0 = 2$ for α, $r_2^0 = 3$ for c^2), using $r_1 = r_2 = 5$ and fully-measured noise-free signals.

levels and availabilities are shown in Fig. 4.7.

Setting the ranks to be $r_1 = r_1^0 = 2$ and $r_2 = r_2^0 = 3$, the recovery results for 75% measurements at 5000th epoch are shown in Fig. 4.9. Setting the ranks to be $r_1 = r_2 = 5$ which are greater than the true ranks, the recovery results at the 5000th epoch are shown in Fig. 4.10 and 4.11. Comparing Fig. 4.8, Fig. 4.9, Fig. 4.10 and 4.11, allowing additional ranks obviously benefit the coefficients recovery. The RMSEs between the true and recovered coefficients are in Table 4.3.

From Table 4.3, we see that the recovery using $r_1 = r_2 = 5$ is better than using $r_1 = 2, r_2 = 3$. From the table and Fig. 4.12, the recovery using $r_1 = r_2 = 5$ is satisfactory even for the case with noisy data (noise STD = 20% of signal STD) and 50% measurements. But when $r_1 = 2$ and $r_2 = 3$, the recovery is problematic for the 50% measurements case as indicated in Fig. 4.13.

4.4 Comparison with two baseline methods

We compare the coefficient recovery result between the SD-PINN and two baseline methods. Given the noise-free and 50% measurements of the field in Sec. 4.3.2 with the

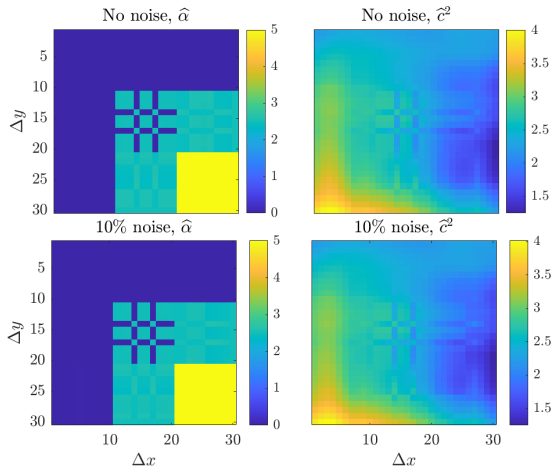

Figure 4.9. For the 75%-measured noise-free and noisy signals, the recovered $\widehat{\alpha}$ and $\widehat{c}^2$ at epoch 5000 given 88 entries on the right, bottom and diagonal, using $r_1 = 2, r_2 = 3$.

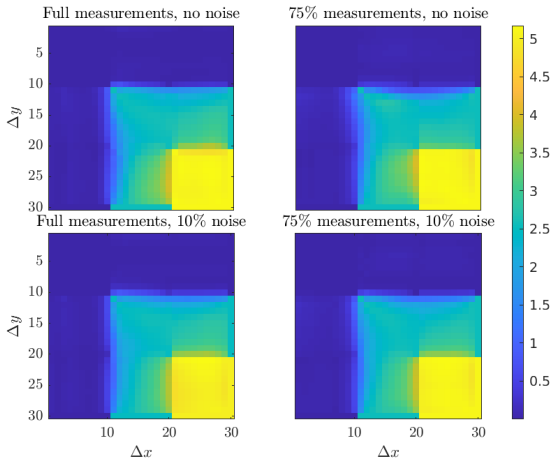

Figure 4.10. The recovered $\widehat{\alpha}$ at epoch 5000 for various settings with $r_1 = r_2 = 5$.

measurements sampled at the same 50% locations as in Fig. 4.7, the coefficients within all the ROI are recovered by: (baseline-1) first interpolating the measurements by spline interpolation [83] to obtain full measurements, and then recover the coefficients for every location iteratively based on the interpolated signals [4][84]; (baseline-2) first recover the coefficients at a few locations with sufficient measurements, and then use the matrix completion approach [85] to recover the coefficients at other locations.

Before diving into the baseline methods, we outline the PDE coefficients recovery by finite difference (FD) [9] with ordinary least squares regression (OLS) which is used in both baseline methods and its limitation. Given the measurements at three consecutive locations along

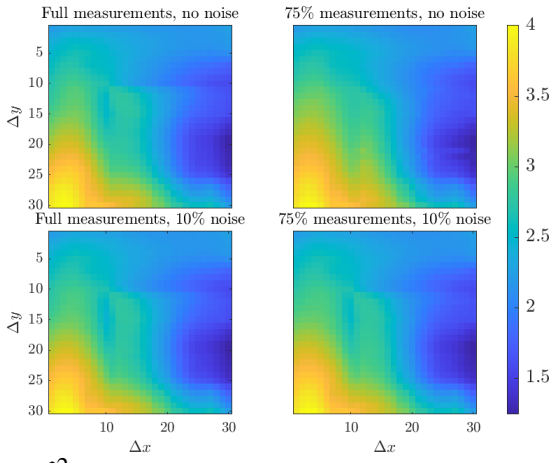

Figure 4.11. The recovered $\widehat{c}^2$ at epoch 5000 for $r_1 = r_2 = 5$.

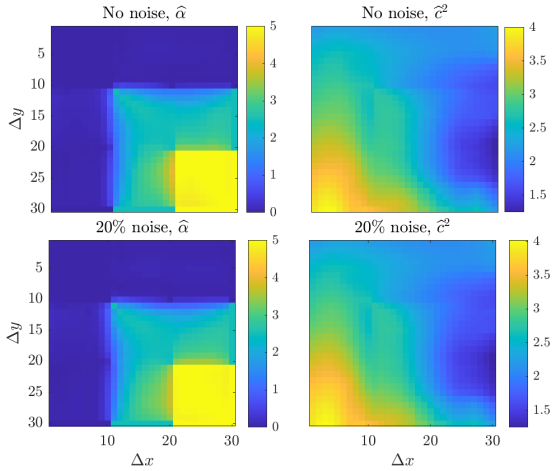

Figure 4.12. For the 50%-measured noise-free and noisy signals, the recovered $\widehat{\alpha}$ and $\widehat{c}^2$ at epoch 5000 given 88 entries on the right boundary, bottom boundary, and the diagonal, using $r_1 = r_2 = 5$.

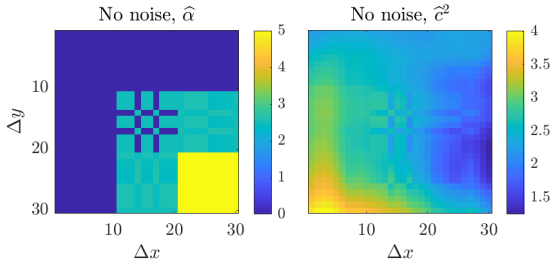

Figure 4.13. For the 50%-measured noise-free signals, the recovered $\widehat{\alpha}$ (left) and $\widehat{c}^2$ (right) at epoch 5000 given 88 entries on the right boundary, bottom boundary, and the diagonal, using $r_1 = 2, r_2 = 3$.

x-axis centered at i with y-coordinate j and time step k: $\{\mathbf{U}(i-1,j,k), \mathbf{U}(i,j,k), \mathbf{U}(i+1,j,k)\}$, the first order spatial derivative along x at (i,j) is computed as $[\mathbf{U}(i+1,j,k) - \mathbf{U}(i-1,j,k)]/2\Delta x$ and the 2nd order derivative is $[\mathbf{U}(i+1,j,k) - 2\mathbf{U}(i,j,k) + \mathbf{U}(i-1,j,k)]/\Delta x^2$. Such calculations can be repeated at all time steps, and thus for location (i,j), we can obtain vectors containing the numerical partial derivatives along x-axis as

$$\begin{aligned} \mathbf{u}_x^{(i,j)} &= \frac{\mathbf{U}(i+1,j,:) - \mathbf{U}(i-1,j,:)}{2\Delta x} \\ \mathbf{u}_{xx}^{(i,j)} &= \frac{\mathbf{U}(i+1,j,:) - 2\mathbf{U}(i,j,:) + \mathbf{U}(i-1,j,:)}{\Delta x^2} \end{aligned} . \tag{4.22}$$

The partial derivatives at (i,j) along the y-axis are computed similarly. For the partial derivatives along time, only the measurements at (i,j) are sufficient:

$$\begin{aligned} \mathbf{u}_t^{(i,j)}(k) &= \frac{\mathbf{U}(i,j,k+1) - \mathbf{U}(i,j,k-1)}{2\Delta t} \\ \mathbf{u}_{tt}^{(i,j)}(k) &= \frac{\mathbf{U}(i,j,k+1) - 2\mathbf{U}(i,j,k) + \mathbf{U}(i,j,k-1)}{\Delta t^2} \end{aligned} \tag{4.23}$$

where $2 \leq k \leq$ number of time steps $- 1$.

From (4.22) and (4.23), we do not consider the FD evaluated at the boundaries of $\mathbf{U}$ which is defined differently and subjected to larger errors. For the considered dataset $\mathbf{U} \in \mathbb{R}^{30\times30\times198}$ in Sec. 4.3.2, $\{\mathbf{u}_x, \mathbf{u}_{xx}, \mathbf{u}_y, \mathbf{u}_{yy}\}$ all of which are in $\mathbb{R}^{198}$ can be computed at all locations except the spatial boundaries, so there are $28^2 = 784$ locations in total. The $\mathbf{u}_t$ and $\mathbf{u}_{tt}$ are also computed at these locations, and according to (4.23), $\{\mathbf{u}_t^{(i,j)}(k), \mathbf{u}_{tt}^{(i,j)}(k)\}$ are well-defined for $2 \leq k \leq 197$. So for each (i,j), we drop first and last entries of $\{\mathbf{u}_t^{(i,j)}, \mathbf{u}_{tt}^{(i,j)}\}$ to make them in $\mathbb{R}^{196}$. Similarly, the first and last entries of the spatial derivative vectors are also dropped. Then for (i,j) we construct a matrix as

$$\mathbf{\Phi}^{(i,j)} = [-\mathbf{u}_t^{(i,j)}, \mathbf{u}_{xx}^{(i,j)} + \mathbf{u}_{yy}^{(i,j)}] \in \mathbb{R}^{196\times2} \tag{4.24}$$

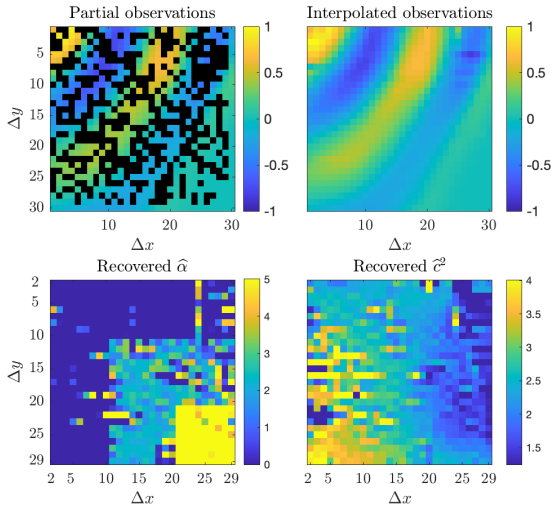

Figure 4.14. One frame of the 50% sampled noise-free signal and its interpolation result, and the recovered PDE coefficients via baseline-1. For this method, the recovered coefficients are located within [2,29] for both axes.

and then the coefficients at (i, j) are recovered by OLS as

$$[\widehat{\alpha}(i,j),\ \widehat{c}(i,j)^2]^{\mathrm{T}} = \mathbf{\Phi}^{(i,j)\dagger}\mathbf{u}_{tt}^{(i,j)} \tag{4.25}$$

where † denotes pseudo-inverse.

From the above discussion, the limitation of the FD+OLS method is that for location (i, j), the measurements at all its neighbors $\{(i-1, j), (i+1, j), (i, j-1), (i, j+1)\}$ must exist. This is not true when there are many sensors out of work, e.g., the 50% measurements as in Fig. 4.7. To recover all coefficients using only partial observations, the two methods are detailed below.

Details of the baseline-1 method: (1) For every frame of the 50% measurements, first do spline interpolation row by row, and then do the interpolation again column by column, and in the end average these two interpolation results to be the interpolated signals at this frame. (2) Except for the four boundaries, for each of the 28^2 locations in the ROI, use the above-mentioned FD+OLS method to recover the coefficients. One frame of the noise-free 50% measurements with its interpolation result and the recovered coefficients are shown in Fig. 4.14.

Details of the baseline-2 method: (1) In addition to the known coefficients on the bottom boundary, right boundary, and the diagonal, as indicated in Fig. 4.7, there are a few other locations eligible for the spatial derivatives to be computed by FD (for such a location, the measurements are available at itself and all its top, bottom, left and right neighbors). We first recover the coefficients by FD+OLS at these locations. (2) Based on the coefficients that are given and recovered at the few locations mentioned above, we recover the coefficients at other locations via matrix completion by nuclear norm minimization (NNM) [85]:

$$\widehat{\mathbf{\Lambda}}_k = \arg\min_{\mathbf{X}} \tau\|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X}\|_F^2 \quad \text{s.t. } \mathscr{P}_\Omega(\mathbf{X}) = \mathscr{P}_\Omega(\mathbf{\Lambda}_k) \tag{4.26}$$

where the nuclear norm $\|\mathbf{X}\|_*$ is the sum of its singular values and $\|\mathbf{X}\|_F$ the Frobenius norm. The (4.26) is solved iteratively from $\mathbf{Y}^0 = \mathbf{0} \in \mathbb{R}^{M_1 \times M_2}$ with step δ by

$$\begin{cases} \mathbf{X}^i = \mathscr{D}_\tau\left(\mathbf{Y}^{i-1}\right) \\ \mathbf{Y}^i = \mathbf{Y}^{i-1} + \delta\,\mathscr{P}_\Omega\left(\mathbf{\Lambda}_k - \mathbf{X}^i\right) \end{cases} \tag{4.27}$$

where $\mathscr{D}_\tau$ the singular value shrinkage operator, i.e., suppose the singular value decomposition (SVD) [71] of $\mathbf{Y}$ with rank r is

$$\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\mathrm{T}}, \quad \mathbf{\Sigma} = \mathrm{diag}\left(\{\sigma_i\}_{1\le i\le r}\right), \tag{4.28}$$

then

$$\mathscr{D}_\tau(\mathbf{Y}) := \mathbf{U}\mathscr{D}_\tau(\mathbf{\Sigma})\mathbf{V}^{\mathrm{T}}, \quad \mathscr{D}_\tau(\mathbf{\Sigma}) = \mathrm{diag}\left(\{\sigma_i - \tau\}_+\right) \tag{4.29}$$

with $\{t\}_+ = \max(0,t)$. From (4.26), the rank r_k of recovered $\widehat{\mathbf{\Lambda}}_k$ is adjustable: as τ increases, r_k decreases in general. Multiple experiments are carried out using various τ, but none of them provide satisfactory coefficient recovery. Among them, the recovered $\widehat{\mathbf{\Lambda}}_1$ with rank 2 (for $\widehat{\alpha}$) and $\widehat{\mathbf{\Lambda}}_2$ with rank 3 (for $\widehat{c}^2$) together with the known α and c^2 on which the recovery is based

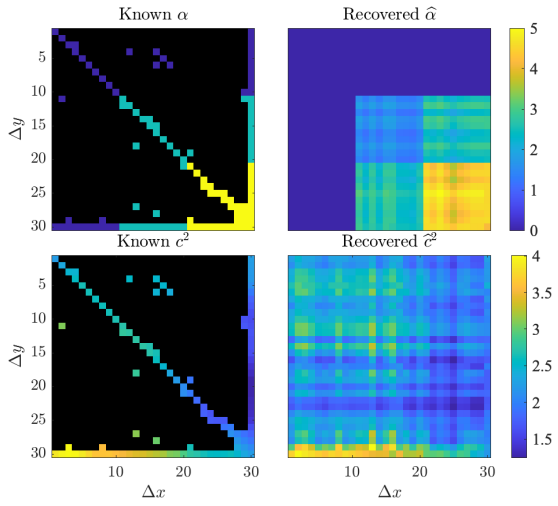

Figure 4.15. Known and recovered PDE coefficients for the baseline-2 method. The known coefficients include the given ones on the right, bottom boundaries and the diagonal as well as the recovered ones via FD+OLS at a few eligible locations. Black pixels are locations without known coefficients.

Table 4.4. RMSEs between the true and recovered PDE coefficients by two baseline methods from noise-free 50% measurements sampled at locations indicated in Fig. 4.7, the corresponding SD-PINN result with $r_1 = r_2 = 5$ from Table 4.3 is included for comparison.

Method	RMSE$_\alpha$	RMSE$_{c^2}$
Baseline-1	1.379	0.929
Baseline-2	0.381	0.810
SD-PINN	0.371	0.140

(including given coefficients and the recovered coefficients via FD+OLS) are shown in Fig. 4.15.

Visual examinations of Fig. 4.14 and 4.15 suggest that the PDE coefficients recovery by the two baseline methods is far poorer than SD-PINN, as shown in Fig. 4.12. The RMSEs for the two baseline methods are in Table 4.4. Compared to Table 4.3, except for the recovery of $\widehat{\alpha}$ by baseline-2 which is slightly worse than SD-PINN ($r_1 = r_2 = 5$ case), all other recoveries are much worse than SD-PINN.

4.5 Conclusion

We propose a spatially-dependent physics-informed neural network (SD-PINN) method to recover the spatially-dependent PDE coefficients from the observations. The PDE coefficients

are recovered as the entries of the matrices and the recovery is formulated as a matrix completion problem with low rank constraints which is solved by a neural network. The experiments show that the proposed method can successfully recover the PDE coefficients and the recovery is robust to noise and poor availability of measurements. The recovery performance is better when the locations of given coefficients are not constrained to too few distinct rows and columns, and is affected by the assumed ranks of the coefficient matrices.

4.6 Acknowledgments

The text of Chapter 4, in full, is a reprint of the material as it appears in R. Liu, P. Gerstoft, "SD-PINN: Deep Learning based Spatially Dependent PDEs Recovery", which is submitted to IEEE Access. A preliminary version of this work is published as R. Liu, P. Gerstoft, "SD-PINN: Physics informed neural networks for spatially dependent PDEs", *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023. The dissertation author was the primary researcher and author of Chapter 4. The coauthors listed in this publication directed and supervised the research.

Chapter 5

Conclusion

In this dissertation, sparse modeling and deep learning methods are utilized to solve problems within two areas of PDE recovering: the identification of PDE terms for unknown PDEs and the recovery of PDE coefficients for given PDE terms. Both spatially-independent and spatially-dependent PDEs are considered, and the latter case can be used for material properties recovery which is valuable for multiple modern applications [70, 80].

Each study in this dissertation demonstrates the capability of using machine learning methods to recover the PDEs from observations. Compared to traditional PDE derivation methods based on theoretical reasoning, machine learning based PDE recovery methods require significantly less expertise in mathematics and physics, and their higher demands for computational power is fulfilled by the substantial increase of the available computational resources in recent times. Furthermore, machine learning based PDE recovery is more adaptable to various situations: a single machine learning model can yield different PDEs based on different input data from various dynamical system observations, whereas the classical physics derivation process varies for each dynamical system. Overall, the machine learning based PDE recovery shows distinctive advantages and the research into this topic will flourish.

Our machine learning based PDE recovery research has progressed through several stages. We start our machine learning based PDE recovery research from the work in Chapter 2, in which only the spatially-independent PDE can be identified from all measurements in the ROI.

The work in Chapter 3 exceeds it by allowing spatially-dependent PDEs to be identified. One limitation of Chapter 3 is that the PDE can only be identified at where the measurements are available. This is overcome in the work of Chapter 4 in which we can recover the spatially dependent PDE coefficients at all locations from partly observed measurements. Alghouth in Chapter 4 we assume the correct type of PDE, this assumption can be easily dropped as will be discussed in Sec. 5.4.

We have implemented a series of methods across different chapters for the PDE recovery. In the work of Chapter 2, a sparse regression method (cross-validation + sparsity penalty) is developed to identify active PDE terms from a redundant dictionary, in which each atom is computed by numerical differentiation (like finite difference) using measurements at all spatio-temporal coordinates in the ROI. In the work of Chapter 3, another sparse regression method (lasso) is employed to identify active PDE terms from many dictionaries, each of which is computed by numerical differentiation using measurements at only one spatial location, and the extracted active PDE terms are the terms in the spatially-dependent PDE at that location. If there is no measurement for a location, then the dictionary at that point can not be computed and thus the PDE can not be identified. In the work of Chapter 4, a modified physics informed neural network SD-PINN is used to recover the spatially-dependent PDE coefficients using the measurements at only parts of the ROI, and thus is superior to the method in Chapter 3 which requires full measurements to achieve satisfactory recovery.

In consideration of the noise conditions, the work in Chapter 2 only deals with noise-free measurements. With an integration transformation, the method in Chapter 3 can recover the PDE from noisy measurements using a transformed dictionary of PDE terms. The SD-PINN in Chapter 4 is also robust to noise, which attributes to the constraint from the physical background and the accurate derivatives computation via automatic differentiation.

We summarize the three papers within this dissertation and discuss our future work as below.

5.1 Automated Partial Differential Equation Identification

A sparse modeling approach to extract unknown spatially-independent PDEs directly from observations is proposed. Given the spatio-temporal measurements of a dynamical field, a dictionary $\mathbf{\Phi}$ with D PDE terms (which is redundant) is generated with its terms computed numerically by finite difference (FD) or pseudo-spectral (PS) method. Without any prior knowledge of the active PDE terms, the method iteratively takes one term from $\mathbf{\Phi}$ as the left hand side (LHS) of the PDE. Then for each assumed LHS term, the method try selecting terms from $\mathbf{\Phi}$ to form the RHS of the PDE, with the sparsity (i.e., number of terms) of RHS varying iteratively from 1 to $D-1$. The choice of RHS terms with the assumed LHS term is made when a loss function combining a data-fitting loss and a sparsity loss is minimized. The data-fitting loss is a squared ℓ_2-norm based on cross-validation, and the sparsity loss is proportional to the number of terms in the RHS. Every assumed LHS with its corresponding selected RHS forms a candidate PDE, and the identified PDE as the method's output is the candidate who fits the observations best. The synthetic fields governed by wave equation, Helmholtz equation and Burgers equation are used to test the method. The method is also implemented to identify the PDEs governing the vibrations of a real aluminum plate and successfully depicts its dispersion properties.

5.2 Recovery of Spatially Varying Acoustical Properties via Automated PDE Identification

A sparse modeling approach, to be specific, constrained ℓ_1-norm minimization is used to identify unknown spatially-dependent PDEs from observations. This method utilizes the sparsity promotion property of ℓ_1-norm minimization [48, 49] to extract the active PDE terms within a dictionary with redundant terms. For each location, a dictionary is computed from the observations measured there. So the spatially-dependent PDEs for all the locations are recovered by extracting the PDE terms iteratively from every dictionary for every location. Thanks to

the innovative structure of the dictionary whose last row representing the sign information of the PDE coefficients, this method can identify the PDE in the absence of any assumed active PDE term efficiently without assuming its LHS iteratively. To accelerate the process, lasso [55], which is the Lagrangian of the constrained ℓ_1-norm minimization is employed. The method is extended by incorporating an integration transformation [12] to enhance its resistance to the noise in observations.

5.3 SD-PINN: Deep Learning based Spatially Dependent PDEs Recovery

An extension of the physics informed neural network (PINN) [7], SD-PINN, is proposed to recover the spatially dependent coefficients of a given type of PDE from noisy and partial observations. Thanks to the mutual constraints between the loss responsible for fitting training data and the loss for fitting the PDE, as well as the precision of automatic differentiation, SD-PINN possesses a strong capability to identify PDEs from noisy observations. Although the PDEs to be recovered are spatially-dependent, the spatial variation of the PDE coefficients is not totally free, because the physical property at one location of the material, on which the dynamics are observed, is related to the property in its neighborhood. We encode this as a low rank constraint of the matrix containing the PDE coefficients at all locations (suppose the region of interest is rectangular), and thus reformulate the PDE recovery as a low-rank matrix completion problem. In this way, the spatially-dependent PDEs at locations without available observations are also recovered.

5.4 Future Work

Our future work will be focused on SD-PINN, as it can recover the spatially-dependent PDE coefficients everywhere from noisy observations at only a small part of the ROI, which is particularly useful in many applications including but not limited to material diagnostics and

geological survey. The potential capabilities of SD-PINN can encompass and exceed all aspects of the previous two papers, as its limitations can be overcome easily as discussed as follows.

In the work of SD-PINN, we have the assumption that the upper limit for the rank r_k of coefficient matrix $\mathbf{\Lambda}_k$ is properly estimated. Thus one limitation comes when this upper limit is difficult to evaluate *a priori*. In the future work, cross-validation may be used to solve this problem. To be specific, we divide the given samples of $(\boldsymbol{x}_m, t_j, u_{mj})$ into a training set and a validation set, e.g., each of which consists of observations at all locations with available signals and a half of the time steps (to keep the size of $\mathbf{\Lambda}_k$ unchanged after dividing the samples, the division should be made on the time axis). We thus train the SD-PINN on the training set using various r_k, and for each choice, we compute $loss_{\mathrm{u}}$ which compares the $\widehat{u}_{mj}$ estimated by $Net_{\boldsymbol{\theta}}$ trained on the training set to the true u_{mj} in the validation set. Then we choose the r_k which provides the minimal $loss_{\mathrm{u}}$ to be the assumed r_k. While training $Net_{\boldsymbol{\theta}}$, we may add noises to the training samples so that when the r_k is assumed too large, the $Net_{\boldsymbol{\theta}}$ will overfit the observations and thus it will produce a higher $loss_{\mathrm{u}}$ when used to compare its estimations with the noise-free observations in the validation set.

Another concern is that how will the SD-PINN work for higher AWGN noise. From (4.10), the methods robustness against noise is adjusted by w_{f}. As the noise is higher, w_{f} should be set greater. In addition, for a higher noise we should use a higher r_k, because in this case for the training samples the correct relationship from $(\boldsymbol{x}_m, t_j)$ to u_{mj} is interrupted more severely. This makes the $Net_{\boldsymbol{\theta}}$ more difficult to extract useful information and thus more likely to be misguided during optimization. Allowing for more columns in $\widehat{\mathscr{U}}_k$ and $\widehat{\mathscr{V}}_k$ will help SD-PINN decrease the error in the recovered coefficients by adding a matrix which is the product of the additional columns as a compensation to the recovered coefficients with unwanted shifts due to the noise. The exploration of the impacts that various levels of noise can have on the above-mentioned cross-validation based rank selection and on the eventual PDE coefficients recovery results is a useful future research direction. Another future research focus will be the effects of other kinds of noise (possibly structured noise) to the SD-PINN.

There is also an assumption that the type of PDE is known for SD-PINN. This assumption can be easily dropped by adding more PDE terms involved in all potential PDEs for the dynamical system of interests. We can use the same procedure to train SD-PINN, and the recovered coefficients for inactive PDE terms should approach zero. By thresholding the coefficients and keeping the PDE terms with only large coefficients, we can recover the spatially-dependent coefficients for an unknown PDE using SD-PINN.

Bibliography

[1] S. H. Rudy, S. L. Brunton, J. L. Proctor, and J. N. Kutz. Data-driven discovery of partial differential equations. *Sci. Adv.*, 3(4):e1602614, 2017.

[2] W. Zhou, H. Zhang, and J. Wang. Sparse bayesian learning based on collaborative neurodynamic optimization. *IEEE Transactions on Cybernetics*, 52(12):13669–13683, 2021.

[3] R. Liu, M. Bianco, and P. Gerstoft. Automated partial differential equation identification. *J. Acoust. Soc. Am.*, 150(4):2364–2374, 2021.

[4] R. Liu, M. J. Bianco, P. Gerstoft, and B. D. Rao. Data-driven spatially dependent PDE identification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3383–3387. IEEE, 2022.

[5] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[7] M. Raissi, P. Perdikaris, and G. E. Karniadakis. 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.

[8] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18:1–43, 2018.

[9] W. F. Ames. *Numerical methods for partial differential equations*. Academic press, 2014.

[10] R. Chartrand. Numerical differentiation of noisy, nonsmooth data. *Int. Sch. Res. Notices*, 2011, 2011.

[11] Steven L Brunton and J Nathan Kutz. *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2019.

[12] P. A.K. Reinbold, D. R. Gurevich, and R. O. Grigoriev. Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Phys. Rev. E*, 101(1):010203, 2020.

[13] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi, and P. Gentine. Enforcing analytic constraints in neural networks emulating physical systems. *Phys. Rev. Lett.*, 126(9):098302, 2021.

[14] S. Zhang and G. Lin. Robust data-driven discovery of governing physical laws with error bars. *Proc. Math. Phys. Eng. Sci.*, 474(2217):20180305, 2018.

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

[16] H. Schaeffer, G. Tran, and R. Ward. Extracting sparse high-dimensional dynamics from limited data. *SIAM J. Appl. Math.*, 78(6):3279–3295, 2018.

[17] Z. Long, Y. Lu, X. Ma, and B. Dong. PDE-Net: Learning PDEs from data. *Proceedings of the 35th International Conference on Machine Learning*, 80:3208-3216, 2018.

[18] Z. Long, Y. Lu, and B. Dong. Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys*, 399:108925, 2019.

[19] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[20] J. B. Harley and J. M.F. Moura. Sparse recovery of the multimodal and dispersive characteristics of lamb waves. *J. Acoust. Soc. Am.*, 133(5):2732–2745, 2013.

[21] J. Bongard and H. Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.*, 104(24):9943–9948, 2007.

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

[23] P. Gerstoft, C. F. Mecklenbräuker, W. Seong, and M. Bianco. Introduction to compressive sensing in acoustics. *J. Acoust. Soc. Am.*, 143(6):3731–3736, 2018.

[24] M. J. Bianco and P. Gerstoft. Travel time tomography with adaptive dictionaries. *IEEE Trans. Comput. Imag.*, 4(4):499–511, 2018.

[25] S. Khatiry Goharoodi, P. Nguyen Phuc, L. Dupré, and G. Crevecoeur. Data-driven discovery of the heat equation in an induction machine via sparse regression. In *ICIT*, 2019.

[26] D. Bhattacharya, L. K. Cheng, and W. Xu. Sparse machine learning discovery of dynamic differential equation of an esophageal swallowing robot. *IEEE Trans. Ind. Electron.*, 2019.

[27] R. Liu, M. J. Bianco, and P. Gerstoft. Wave equation extraction from a video using sparse modeling. In *Proc.53th Asilomar Conf. on Circuits, Systems and Computers.*, pages 2160–2165. IEEE, 2019.

[28] J. Carcione. A generalization of the fourier pseudospectral method. *Geophysics*, 75(6):A53–A56, 2010.

[29] H. Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proc. Math. Phys. Eng. Sci.*, 473(2197):20160446, 2017.

[30] B. Fornberg. High-order finite differences and the pseudospectral method on staggered grids. *SIAM J Numer. Anal.*, 27(4):904–918, 1990.

[31] C. M. Bishop. *Pattern recognition and machine learning*. springer, 2006.

[32] T. T. Cai and L. Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information theory*, 57(7):4680–4688, 2011.

[33] B. Qiao, X. Zhang, C. Wang, H. Zhang, and X. Chen. Sparse regularization for force identification using dictionaries. *J. Sound Vib.*, 368:71–86, 2016.

[34] B. Fornberg. The pseudospectral method: Comparisons with finite differences for the elastic wave equation. *Geophysics*, 52(4):483–501, 1987.

[35] K. Atkinson, W. Han, and D. E Stewart. *Numerical solution of ordinary differential equations*, volume 108. John Wiley & Sons, 2011.

[36] K. S. Alguri, J. Melville, and J. B. Harley. Baseline-free guided wave damage detection with surrogate data and dictionary learning. *J. Acoust. Soc. Am.*, 143(6):3807–3818, 2018.

[37] S. M. Ziola and M. R. Gorman. Source location in thin plates using cross-correlation. *J. Acoust. Soc. Am.*, 90(5):2551–2556, 1991.

[38] D. Alleyne and P. Cawley. A two-dimensional fourier transform method for the measurement of propagating multimode signals. *J. Acoust. Soc. Am.*, 89(3):1159–1168, 1991.

[39] D. H. Johnson and D. E. Dudgeon. *Array signal processing: concepts and techniques*. Simon & Schuster, Inc., 1992.

[40] M. van Berkel, G. Vandersteen, E. Geerardyn, R. Pintelon, H. Zwart, and M. de Baar. Frequency domain sample maximum likelihood estimation for spatially dependent parameter estimation in pdes. *Automatica*, 50(8):2113–2119, 2014.

[41] S. Kramer and E. M. Bollt. Spatially dependent parameter estimation and nonlinear data assimilation by autosynchronization of a system of partial differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 23(3):033101, 2013.

[42] F. C. Lin, M. H. Ritzwoller, and R. Snieder. Eikonal tomography: surface wave tomography by phase front tracking across a regional broad-band seismic array. *Geophys. J. Int*, 177(3):1091–1110, 2009.

[43] I. D. Khurjekar and J. B. Harley. Closing the sim-to-real gap in guided wave damage detection with adversarial training of variational auto-encoders. In *Proc. 2022 IEEE Int. Conf. Acoust. Speech Signal Process., ICASSP*, pages 3823–3827. IEEE, 2022.

[44] C. Huang, K. Wang, R. W. Schoonover, L. V. Wang, and M. A. Anastasio. Joint reconstruction of absorbed optical energy density and sound speed distributions in photoacoustic computed tomography: a numerical investigation. *IEEE Trans. Comput. Imag.*, 2(2):136–149, 2016.

[45] P. S. Fuchs, S. Mandija, P. R. S. Stijnman, W. M. Brink, Cornelis A.T. van den Berg, and R. F. Remis. First-order induced current density imaging and electrical properties tomography in MRI. *IEEE Trans. Comput. Imag.*, 4(4):624–631, 2018.

[46] J. Rubinstein. Sine-gordon equation. *J. Math. Phys.*, 11(1):258–266, 1970.

[47] M. J. Ablowitz, D. J. Kaup, A. C. Newell, and H. Segur. Method for solving the sine-Gordon equation. *Phys. Rev. Lett.*, 30(25):1262, 1973.

[48] D. Malioutov, M. Cetin, and A. S. Willsky. A sparse signal reconstruction perspective for source localization with sensor arrays. *IEEE Trans. Signal Process.*, 53(8):3010–3022, 2005.

[49] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via $\ell 1$ minimization. *Proc. Natl. Acad. Sci.*, 100(5):2197–2202, 2003.

[50] B. D. Rao. Signal processing with the sparseness constraint. In *Proc. 1998 IEEE Int. Conf. Acoust., Speech and Signal Process., ICASSP*, volume 3, pages 1861–1864. IEEE, 1998.

[51] R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Process. Mag.*, 24(4):118-121, 2007.

[52] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Trans. Inf. Theory*, 56(4):1982–2001, 2010.

[53] M. A. Salman, and J. Romberg. Sparse recovery of streaming signals using L1-homotopy. *IEEE Trans. Signal Process.*, 62(16):4209–4223, 2014.

[54] A. Xenaki, P. Gerstoft and M. Mosegaard. Compressive beamforming. *J. Acoust. Soc. Am.*, 136(1):260–271, 2014.

[55] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[56] M. Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using $\ell 1$ -constrained quadratic programming (lasso). *IEEE Trans. Inf. Theory*, 55(5):2183–2202, 2009.

[57] C. F. Mecklenbräuker, P. Gerstoft, A. Panahi, and M. Viberg. Sequential bayesian sparse signal reconstruction using array data. *IEEE Trans. Signal Process.*, 61(24):6344–6354, 2013.

[58] P. Gerstoft, A. Xenaki, and C. F. Mecklenbräuker. Multiple and single snapshot compressive beamforming. *J. Acoust. Soc. Am.*, 138(4):2003–2014, 2015.

[59] C. F. Mecklenbräuker, P. Gerstoft, and E. Zöchmann. c–lasso and its dual for sparse signal estimation from array data. *Signal Process.*, 130:204–216, 2017.

[60] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, 33(1):1, 2010.

[61] Y. C. Hon, and X. Z. Mao. An efficient numerical scheme for Burgers' equation. *Appl. Math. Comput.*, 95(1):37–50, 1998.

[62] H. Mitome. An exact solution for finite-amplitude plane sound waves in a dissipative fluid. *J. Acoust. Soc. Am.*, 86(6):2334–2338, 1989.

[63] M. M. Sigalas and E. N. Economou. Elastic waves in plates with periodically placed inclusions. *J. Appl. Phys.*, 75(6):2845–2850, 1994.

[64] F. Assous, P. Degond, E. Heintze, P. A. Raviart, and J. Segré. On a finite-element method for solving the three-dimensional maxwell equations. *J. Comput. Phys.*, 109(2):222–237, 1993.

[65] H. Xu, H. Chang, and D. Zhang. DL-PDE: Deep-learning based data-driven discovery of partial differential equations from discrete and noisy data. *Commun. Comput. Phys.*, 29:698–728, 2021.

[66] P. Pilar and N. Wahlström. Physics-informed neural networks with unknown measurement noise. *arXiv preprint arXiv:2211.15498*, 2022.

[67] Camps-Valls G., Martino L., Svendsen D. H., Campos-Taberner M., Muñoz-Marí J., Laparra V., Luengo D., and García-Haro F. J. Physics-aware gaussian processes in remote sensing. *Appl. Soft Comput.*, 68:69–82, 2018.

[68] Eldan, R. and Shamir, O. The power of depth for feedforward neural networks. *Conference on learning theory*, 907–940, 2016.

[69] Svozil, D., Kvasnicka, V., and Pospichal, J. Introduction to multi-layer feed-forward neural networks. *Chemometr. Intell. Lab. Syst.*, 39(1):43–62, 1997.

[70] E. Zhang, M. Yin, and G. E. Karniadakis. Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging. *arXiv preprint arXiv:2009.04525*, 2020.

[71] M. E. Wall, A. Rechtsteiner, and L. M. Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[72] R. Liu and P. Gerstoft. SD-PINN: Physics informed neural networks for spatially dependent pdes. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[73] M. J. Buckingham. On the transient solutions of three acoustic wave equations: van wijngaarden's equation, stokes' equation and the time-dependent diffusion equation. *J. Acoust. Soc. Am*, 124(4):1909–1920, 2008.

[74] Y. Chen and Y. Chi. Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization. *IEEE Signal Process. Mag.*, 35(4):14–31, 2018.

[75] M. A. Davenport and J. Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE J. Sel. Top. Signal Process.*, 10(4):608–622, 2016.

[76] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[77] D. I. Bernstein, G. Blekherman, and R. Sinn. Typical and generic ranks in matrix completion. *Linear Algebra Appl.*, 585:71–104, 2020.

[78] O. Klopp. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20:282–303, 2014.

[79] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[80] Y. Zhang, H. Fu, Y. Qin, K. Wang, and J. Ma. Physics-informed deep neural network for inhomogeneous magnetized plasma parameter inversion. *IEEE Antennas Wirel. Propag. Lett.*, 21(4):828–832, 2022.

[81] A. Kamali, M. Sarabian, and K. Laksari. Elasticity imaging using physics-informed neural networks: Spatial discovery of elastic modulus and Poisson's ratio. *Acta Biomater.*, 155:400–409, 2023.

[82] K. He, X. Zhang, S. Ren and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE international conference on computer vision*, 1026–1034, 2015.

[83] H. Akima. A new method of interpolation and smooth curve fitting based on local procedures. *J. ACM.*, 17(4):589–602, 1970.

[84] R. Liu, P. Gerstoft, M. J. Bianco, and B. D. Rao. Recovery of spatially varying acoustical properties via automated partial differential equation identification. *J. Acoust. Soc. Am.*, 153(6):3169–3180, 2023.

[85] J. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.*, 20(4):1956–1982, 2010.