System Identification and Data-Driven Modeling

Liu, Donglin

2025

– CENTRUM SCIENTIARUM MATHEMATICARUM –

System Identification and Data-Driven Modeling

DONGLIN LIU

Lund University
Faculty of Engineering
Centre for Mathematical Sciences
Mathematics

System Identification and Data-Driven Modeling

System Identification and Data-Driven Modeling

by Donglin Liu

LUND
UNIVERSITY

Thesis for the degree of Doctor of Philosophy
Thesis advisors: Prof. Alexandros Sopasakis
Faculty opponent: Prof. Markos A. Katsoulakis

To be presented, with the permission of the Faculty of Engineering of Lund University, for public criticism in the MH:H lecture hall (Hörmander) at the Centre for Mathematical Sciences on Friday, the 17th of October 2025 at 13:00.

Organization
LUND UNIVERSITY

Centre for Mathematical Sciences
Box 118
SE–221 00 LUND
Sweden

Document name
DOCTORAL DISSERTATION

Date of disputation
2025-10-17

Sponsoring organization

Author(s)
Donglin Liu

Title and subtitle
System Identification and Data-Driven Modeling

Abstract
This thesis explores the integration of machine learning techniques with classical numerical methods to develop more robust approaches for data-driven modeling. Motivated by challenges such as noise, partial observability, and data scarcity, it proposes hybrid frameworks that combine domain knowledge with learning-based methods. The research focuses on three key areas: parameter estimation, identification of complex dynamical systems from observational data, and synthetic data generation. The first two studies apply neural ordinary differential equations and Bayesian optimization to enable robust parameter estimation and dynamic reconstruction, with applications such as epidemic modeling. The third study advances sparse identification of nonlinear dynamics by incorporating group similarity and Earth Mover's Distance to improve robustness and generalization. The final two studies focus on high-dimensional data settings, developing a graph-enhanced diffusion model for spatiotemporal imputation and employing generative diffusion models to synthesize realistic biometric data, such as fingerprints. These contributions demonstrate the value of generative and hybrid modeling in domains with limited or noisy data and strict privacy constraints.

Key words
generative models, parameter estimation, model identification, data-driven modelling

Classification system and/or index terms (if any)

Supplementary bibliographical information

Language
English

ISSN and key title
1404-0034

ISBN
978-91-8104-667-0 (print)
978-91-8104-668-7 (pdf)

Recipient's notes

Number of pages
164

Price

Security classification

Signature _____ Date 2025-09-10

System Identification and Data-Driven Modeling

by Donglin Liu

LUND
UNIVERSITY

Faculty of Engineering, Centre for Mathematical Sciences

ISBN: 978-91-8104-667-0 (print)
ISBN: 978-91-8104-668-7 (pdf)
LUTFTM-1002-2025

Printed in Sweden by Media-Tryck, Lund University, Lund 2025

To my wife and child —
your love, a steady flame,
guides my heart and lights my way.

Contents

List of publications

This thesis is based on the following publications, referred to by their Roman numerals:

I **A data driven approach for resolving time-dependent differential equations with noise**
D. Liu and A. Sopasakis
IFAC-PapersOnLine, 59(6):379–384, 2025

II **A combined neural ODE-Bayesian optimization approach to resolve dynamics and estimate parameters for a modified SIR model with immune memory**
D. Liu and A. Sopasakis
Heliyon, 10(19):e38276, 2024

III **Enhancing sparse identification of nonlinear dynamics with Earth-Mover distance and group similarity**
D. Liu and A. Sopasakis
Chaos: An Interdisciplinary Journal of Nonlinear Science, 35(3):033139, 03 2025

IV **Graph-enhanced diffusion models for spatiotemporal imputation**
D. Liu, Oskar Åström and A. Sopasakis
Working paper

V **Fingerprint synthesis from diffusion models and generative adversarial networks**
W. Tang, D. A. F. Llamosas, **D. Liu**, K. Johnsson and A. Sopasakis
Advances in Information and Communication, pp. 289–312, Cham, 2025. Springer Nature Switzerland

Additional publications by the author

VI **Traffic predictions using graph neural networks on real-time observations**
J. Hansen, **D. Liu**, and A. Sopasakis
Proceedings of the 7th International Conference on Advances in Signal Processing and Artificial Intelligence, pages 155–160. IFSA Publishing, S. L., 04 2025

VII **Is lung involvement a favorable prognostic factor for pancreatic ductal adenocarcinoma with synchronous liver metastases?–A propensity score analysis**
H. Ouyang, W. Ma, X. Jiang, A. S. Gerdtsson, **D. Liu**, and Z. Pan
Expert Review of Gastroenterology & Hepatology, 17(4):405–412, 2023. PMID: 36803208

VIII **Systemic chemotherapy with or without hepatic arterial infusion chemotherapy for liver metastases from pancreatic cancer: A propensity score matching analysis**
H. Ouyang, W. Ma, T. Si, **D. Liu**, P. Chen, A. S. Gerdtsson, J. Song, Y. Ni, J. Luo, and Z. Yan
Clinical Colorectal Cancer, 2022

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Alexandros Sopasakis, for his invaluable guidance, encouragement, and support throughout the course of my research. His insights and patience have been instrumental in shaping this work.

I am also deeply thankful to my friends and colleagues at the Centre for Mathematical Sciences and beyond. Special thanks to Abolfazl, Alex, Amanda, Anna, Felix, Germán, Georgios, Jaime, Lea, Magnus, Måns, Olof, Oskar, Weiwei, and Yaqing for making daily work life both productive and enjoyable. I would also like to express my sincere appreciation to my teachers and colleagues: Claus Führer, Dragi Anevski, Erik Lindström, Johan Lindström, Jonas Lindemann, Magnus Goffeng, Magnus Wiktorsson, Mikael Nilsson, Nader Tajvidi, Niels Christian Overgaard, Philipp Birken, Stanislav Volkov, Stefan Diehl, Sigmundur Gudmundsson, and Tony Stillfjord. Additionally, I am truly grateful to Mengwu Guo for the many meaningful and thought-provoking discussions we have shared.

I am deeply grateful to my family. My sincere thanks go to my parents for their constant encouragement, understanding, and steadfast support. My son has brought boundless joy into my life and has helped me grow in ways I never imagined. Last but certainly not least, I owe profound thanks to my wife, 任尚上. Her love, patience, and unwavering belief in me have been a steady source of strength and inspiration throughout this journey.

Popular summary in English

Knowledge can guide engineering practice, while data gathered through engineering efforts can, in turn, drive scientific discovery. A classic example of this interplay comes from the history of astronomy. In the 16th century, before telescopes were invented, Tycho Brahe built advanced observatories and collected extensive, high-precision astronomical data. Johannes Kepler used this data, especially on the motion of Mars, to formulate his laws of planetary motion. Later, Isaac Newton drew on Kepler's work to derive the law of universal gravitation in his Principia Mathematica (1687), offering a physical explanation for planetary motion. These milestones laid the groundwork for modern physics and eventually led to more sophisticated observational tools, such as the Hubble Space Telescope.

Today, in a world increasingly shaped by data, the ability to uncover structure in complex systems is more critical than ever. Whether the task is to simulate fingerprints, model disease spread, reconstruct differential equations from noisy signals, or discover governing laws from observed data, a unifying goal remains: to use machine learning and mathematical tools to reveal patterns and rules that are not directly observable.

The development of artificial intelligence—particularly advances in neural networks and machine learning algorithms—has been essential to the rise of modern generative models. These models offer new ways to accelerate the cycle between data and discovery by generating realistic, high-quality synthetic data, especially in fields where data is scarce, costly, or sensitive, such as cybersecurity, healthcare, and finance. Synthetic data augmentation helps balance datasets, enhance model robustness, and reduce the time and expense of data collection. For instance, the fifth article in this thesis focuses on synthesizing fingerprint data, which is notoriously difficult to obtain due to privacy concerns and the need for large, diverse samples. Generating realistic artificial fingerprints enables more efficient and thorough testing and training of biometric authentication systems.

Despite their powerful capabilities, generative models that rely purely on learning from data carry inherent risks. Neural networks, while excellent function approximators, often act as black boxes—capturing statistical patterns without offering insight into the true causal or physical relationships between variables. This limitation can result in unreliable behavior, particularly in time-dependent or scientific systems. To address this, hybrid modeling approaches that integrate domain knowledge have emerged as a promising direction. In the first article of this thesis, for example, a neural ODE is combined with Bayesian optimization

to model epidemic dynamics that incorporate immune memory—a step toward more interpretable and biologically grounded epidemiological models.

Building on the integration of data and knowledge, a natural next step is to use data not just to refine models but to discover them. This area, known as data-driven scientific discovery, focuses on identifying the underlying equations or governing laws of a system directly from data. Techniques such as sparse regression and symbolic regression are used to extract concise, interpretable models that capture the essential structure of complex dynamics, turning raw measurements into mathematical insight. The third study in this thesis extends the sparse regression framework by incorporating Earth Mover's distance and group similarity, allowing for more robust identification of governing laws across families of related systems. By discovering such models, we take a step toward understanding not only what the data predicts, but why it behaves the way it does.

As Galileo observed, "The book of nature is written in the language of mathematics." This thesis treats data not merely as a final outcome, but as a window into the fundamental laws that govern the natural world. Whether through generating synthetic data, modeling dynamic processes, or discovering governing equations, each study aims to bridge empirical observation with mathematical insight. By applying modern machine learning techniques, this work demonstrates how we can infer, reconstruct, and generate models of systems that are complex, noisy, and often hidden from direct view.

中文科普摘要

知识能够指导工程实践，而工程过程中积累的数据又反过来推动科学发现。天文学史上有一个经典例子：16世纪望远镜尚未发明时，提霍 · 布拉赫（Tycho Brahe）建造了先进的天文台，收集了大量高精度观测数据。开普勒（Johannes Kepler）利用这些数据，尤其是火星运动的观测，提出了著名的行星运动三大定律。随后，牛顿（Isaac Newton）基于开普勒的成果，在《自然哲学的数学原理》（Principia Mathematica，1687）中推导出万有引力定律，物理性地解释了行星运动。这些突破奠定了现代物理学的基础，也推动了如哈勃太空望远镜等更精密观测工具的诞生。

当今数据驱动的时代，揭示复杂系统的结构规律比以往更加重要。无论是模拟指纹、建模疾病传播、从噪声信号中重建微分方程，还是从观测数据中发现系统支配规律，核心目标始终如一：借助机器学习和数学工具，揭示肉眼无法直接观察的模式与法则。

人工智能的飞速发展，尤其是神经网络和机器学习算法的突破，为现代生成模型的兴起提供了关键支撑。此类模型加速了数据与发现的循环，特别适合应用于数据稀缺、收集成本高或隐私敏感的领域，如网络安全、医疗和金融。通过生成逼真且高质量的合成数据，数据增强不仅平衡了数据集、提升了模型稳定性，也显著降低了数据采集的时间和成本。例如，本论文第五篇文章聚焦指纹数据合成。由于隐私限制及对大规模多样样本的需求，指纹数据难以获取。通过生成真实感强的人工指纹，能更高效地测试与训练生物识别认证系统。

然而，尽管生成模型功能强大，若完全依赖于数据学习，也存在一些内在风险。神经网络虽是优秀的函数逼近器，却常表现为“黑箱”，仅捕捉统计相关性，难以揭示变量间的因果或物理关系。这种局限可能导致模型在时序动态和科学工程应用中预测失误。为此，融合领域知识的混合建模方法逐渐受到重视。论文第一篇文章结合神经常微分方程与贝叶斯优化，建立了包含免疫记忆机制的传染病动态模型，朝向更具生物合理性和可解释性的建模迈进。

在数据与知识融合的基础上，一个顺理成章的进展方向是利用数据不仅优化模型，更用于发现模型。所谓“数据驱动的科学发现”，旨在从观测数据中识别系统的基本方程或支配规律。稀疏回归、符号回归等技术能提取简洁且可解释的模型，揭示复杂动力系统的内在结构，将原始测量转化为数学洞见。论文第三篇文章通过引入动土距离和群体相似性策略，增强了对一类相关系统通用规律的稳健识别。发现这些模型，使我们不仅提高了预测能力，也更深入理解了系统行为背后的机理。

正如伽利略所言：“自然之书是用数学的语言书写的。”本论文不仅将数据视为最终结果，更将其视为洞察自然界基本规律的窗口。无论是生成合成数据、模拟动态过程，还是挖掘支配系统演化的方程，每一项研究都旨在架起经验观测与

数学洞见之间的桥梁。通过应用现代机器学习技术，本文展示了如何推断、重构乃至生成那些复杂、多噪声、常被直接观测所遮蔽的系统模型。

Populärvetenskaplig sammanfattning på svenska

Kunskap kan vägleda ingenjörspraxis, medan data som samlas in genom ingenjörsarbete i sin tur kan driva vetenskapliga upptäckter. Ett klassiskt exempel på detta samspel kommer från astronomins historia. På 1500-talet, innan teleskopet uppfanns, byggde Tycho Brahe avancerade observatorier och samlade in omfattande och högprecisa astronomiska data. Johannes Kepler använde dessa data, särskilt om Mars rörelse, för att formulera sina lagar om planeternas rörelser. Senare använde Isaac Newton Keplers arbete för att härleda lagen om gravitation i sitt verk Principia Mathematica (1687), vilket gav en fysisk förklaring till planetrörelserna. Dessa milstolpar lade grunden för modern fysik och ledde så småningom till mer sofistikerade observationsverktyg, såsom Hubbleteleskopet.

Idag, i en värld som allt mer formas av data, är förmågan att urskilja struktur i komplexa system viktigare än någonsin. Oavsett om uppgiften är att simulera fingeravtryck, modellera sjukdomsspridning, rekonstruera differentialekvationer från brusiga signaler eller upptäcka styrande lagar från observerade data, finns ett gemensamt mål: att med hjälp av maskininlärning och matematiska verktyg avslöja mönster och regler som inte är direkt observerbara.

Utvecklingen inom artificiell intelligens –särskilt framstegen inom neurala nätverk och algoritmer för maskininlärning –har varit avgörande för framväxten av moderna generativa modeller. Dessa modeller erbjuder nya sätt att påskynda cykeln mellan data och upptäckt genom att generera realistiska, högkvalitativa syntetiska data, särskilt inom områden där data är knapp, dyr eller känslig –som cybersäkerhet, hälso- och sjukvård samt finans. Syntetisk dataförstärkning hjälper till att balansera datamängder, förbättra modellernas robusthet och minska tids- och kostnadskraven för datainsamling. I det femte arbetet i denna avhandling fokuseras till exempel på att syntetisera fingeravtrycksdata, vilket är notoriskt svårt att samla in på grund av integritetsfrågor och behovet av stora, varierade datamängder. Att kunna generera realistiska konstgjorda fingeravtryck möjliggör mer effektiv och grundlig testning och träning av biometriska autentiseringssystem.

Trots sina kraftfulla möjligheter medför generativa modeller som enbart lär sig från data vissa risker. Neurala nätverk är visserligen utmärkta funktionsapproximerare, men de fungerar ofta som "svarta lådor"–de fångar statistiska mönster utan att ge insikt i de verkliga kausala eller fysikaliska samband som råder mellan variabler. Denna begränsning kan leda till opålitligt beteende, särskilt i tidsberoende eller vetenskapliga system. För att hantera detta har hybrida modeller som integrerar domänkunskap fått ökad uppmärksamhet. I det första arbetet i

denna avhandling kombineras till exempel en neural ODE med Bayesiansk optimering för att modellera epidemidynamik som tar hänsyn till immunologiskt minne –ett steg mot mer tolkbara och biologiskt förankrade epidemiologiska modeller.

Med denna integration av data och kunskap som grund är ett naturligt nästa steg att använda data inte bara för att förbättra modeller utan för att upptäcka dem. Detta område, som kallas datadriven vetenskaplig upptäckt, fokuserar på att identifiera de underliggande ekvationerna eller styrande lagarna i ett system direkt från data. Tekniker som gles regressionsanalys och symbolisk regression används för att extrahera koncisa, tolkbara modeller som fångar den väsentliga strukturen i komplex dynamik –och omvandlar råa mätningar till matematisk insikt. Den tredje studien i denna avhandling vidareutvecklar ramen för gles regression genom att införa Earth Mover's Distance och gruppsimilaritet, vilket möjliggör mer robust identifiering av styrande lagar över familjer av relaterade system. Genom att upptäcka sådana modeller närmar vi oss förståelsen av inte bara vad datan förutspår, utan varför den beter sig som den gör.

Som Galileo konstaterade: Naturens bok är skriven på matematikens språk. Denna avhandling ser inte data enbart som ett slutresultat, utan som ett fönster mot de grundläggande lagar som styr den naturliga världen. Oavsett om det handlar om att generera syntetiska data, modellera dynamiska processer eller upptäcka styrande ekvationer, syftar varje studie till att överbrygga klyftan mellan empirisk observation och matematisk insikt. Genom att tillämpa moderna maskininlärningstekniker visar arbetet hur vi kan härleda, rekonstruera och generera modeller av system som är komplexa, brusiga och ofta dolda för direkt observation.

1. Introduction

Machine learning (ML) has emerged as an indispensable tool in scientific computing and engineering, fundamentally altering the methodologies employed for the analysis, modeling, and optimization of complex systems. Historically, scientific inquiry relied predominantly on first-principles modeling, characterized by physics-derived equations, domain-specific expertise, and analytical techniques. Early computational strategies, encompassing numerical analysis and statistical methods such as linear regression and Bayesian inference, underpinned data-driven investigations. At the same time, methods involving Monte Carlo simulations and Gaussian processes advanced the field of uncertainty quantification [37, 59, 61, 65]. Early work in artificial intelligence, such as Rosenblatt's perceptron [68], demonstrated the promise of machine learning. However, scientific modeling continued to rely primarily on physical principles and mathematical rigor, with machine learning serving only a supporting role.

The rise of abundant data and computational power has since propelled ML methods into the heart of scientific computing. Techniques such as Support Vector Machines (SVMs) [21], Random forests [13], and early neural network architectures like Convolutional Neural Networks (CNNs) [43] and Long Short-Term Memory (LSTM) units [33] enhanced tasks like classification and regression across scientific domains. During this period, ML methods served primarily as a complementary tool, augmenting traditional physics-based models by excelling at pattern recognition and data analysis. The subsequent advent of deep learning marked a pivotal shift within ML. With its capacity to handle large-scale, complex datasets, deep learning help introduce innovative architectures such as Transformers [84] and Graph Neural Networks (GNNs) [41], alongside generative frameworks like diffusion models [74], thereby enabling tackling complex systems with higher accuracy.

Due to data scarcity and the complexity of real-world scenarios, ML models often suffer from low accuracy. Moreover, since these models rely on statistical

correlations rather than genuine understanding or common sense, they tend to lack robustness. Consequently, applying ML methods directly to raw data may lead to suboptimal performance. In contrast, incorporating domain knowledge into the modeling process can enrich the information available to the model, making ML approaches more practical and effective in real-world applications [36].

Domain knowledge can be integrated into machine learning models through two primary strategies: knowledge embedding and knowledge discovery. Knowledge embedding involves explicitly encoding expert insights into the model architecture or training process—using techniques such as feature engineering, informative priors, structural constraints, or regularization terms. These techniques allow models to incorporate established scientific understanding, improving generalization and interpretability.

Recent advances illustrate the growing synergy between machine learning and mathematical modeling. For example, the transition from finite-depth architectures like Residual Neural Networks (ResNet) [31] to continuous-depth formulations such as Neural Ordinary Differential Equations (Neural ODEs) [19] reflects a shift toward integrating dynamical systems theory into deep learning, improving scalability and expressiveness. Likewise, the progression from Generative Adversarial Networks (GANs) [29] to diffusion- and score-based generative models [32, 74] demonstrates how probabilistic modeling and differential equations can drive advances in high-fidelity data generation and simulation. Physics-Informed Neural Networks (PINNs) [64] further extend this paradigm by embedding governing physical laws directly into the neural architecture, yielding more robust and physically consistent predictions.

In contrast, knowledge discovery focuses on using machine learning to uncover new patterns or relationships within the data, leading to novel insights that can refine or expand existing domain knowledge. This approach is particularly effective for solving inverse problems, identifying latent variables from noisy data, as well as estimating parameters in complex systems. Techniques like symbolic regression [5] or the Sparse Identification of Nonlinear Dynamics (SINDy) [15] enable ML architectures to autonomously uncover hidden governing laws, accelerating advancements across scientific disciplines.

This thesis proposes hybrid methodologies that integrate mathematical modeling with machine learning to analyze complex dynamical systems in the presence of incomplete, noisy, or limited data. At its core, the work advances techniques that extract structure, recover parameters, or generate data by embedding domain knowledge into data-driven models. Papers I and II explore hybrid ap-

proaches that integrate machine learning with numerical solvers for parameter estimation and model discovery in differential equations. Paper III extends this line by refining the SINDy framework with statistical tools to improve robustness and accuracy in identifying governing laws. Building on these foundations, Papers IV and V address generative modeling in data-scarce regimes: the former proposes imputation strategies for incomplete traffic datasets, while the latter synthesizes fingerprint data where labeled examples are limited. Together, these contributions offer a unified framework for scientific modeling across inference, completion, and generation.

The structure of the thesis is as follows. Sections 2–5 introduce the key theoretical and methodological foundations underlying Papers I–V. Section 2 covers fundamental concepts in machine learning, including both supervised and unsupervised learning, which serve as the basis for the more advanced methods explored later. Section 3 focuses on fully data-driven approaches, with an emphasis on generative models for complex data. Section 4 shifts toward hybrid modeling paradigms that integrate machine learning with domain knowledge—highlighting applications such as noise-perturbed ordinary differential equations and inverse problems involving parameter estimation. Section 5 addresses the use of machine learning in scientific discovery, particularly for inferring governing equations from observational data. Finally, Section 6 synthesizes the main research contributions, draws conclusions, and discusses avenues for future research.

2. Machine learning techniques

One of the core strengths of machine learning (ML) is its ability to uncover hidden patterns, relationships, and structures in complex datasets. Traditional scientific and engineering methods typically depend on explicit mathematical models and predefined assumptions about system behavior. In contrast, ML employs data-driven computational techniques to uncover meaningful patterns directly from observations—often revealing structures that are difficult or even impossible to detect using conventional approaches. These capabilities are especially evident in both supervised and unsupervised learning, which serve complementary roles in extracting predictive and descriptive insights from complex datasets.

2.1 Supervised learning

Supervised learning seeks to infer a mapping function f that relates input features $\mathbf{X} \in \mathcal{X}$ to output labels $\mathbf{Y} \in \mathcal{Y}$, enabling the prediction of outputs for previously unseen data. Here, $\mathcal{X}$ denotes the feature space, and $\mathcal{Y}$ the label space. Formally, the goal is to learn a function $f : \mathcal{X} \to \mathcal{Y}$ from a given training dataset,

$$\mathcal{D} = \{(\mathbf{X}_i, \mathbf{Y}_i)\}_{i=1}^{N},$$

where N is the number of training samples, $\mathbf{X}_i \in \mathbb{R}^d$ represents an input feature vector, and $\mathbf{Y}_i \in \mathbb{R}^k$ is the corresponding target output. The target $\mathbf{Y}_i$ may be a discrete label in classification tasks or a continuous value in regression problems.

A fundamental assumption in supervised learning is that the training data are independently and identically distributed (i.i.d.) according to an unknown but fixed joint probability distribution $P(\mathbf{X}, \mathbf{Y})$. The objective of learning is to

approximate the conditional expectation $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$, which characterizes the dependency between inputs $\mathbf{X}$ and outputs $\mathbf{Y}$ as encoded in the joint distribution $P(\mathbf{X}, \mathbf{Y})$, by identifying a suitable function f. We assume that candidate functions $f_\theta : \mathbb{R}^d \to \mathbb{R}^k$ are drawn from a measurable hypothesis space $\mathcal{F}$, where each function is parameterized by a model parameter $\theta \in \Theta$, and Θ denotes the set of all admissible parameter configurations.

To determine a good predictive function from the given dataset $\mathcal{D}$, we introduce a loss function $L(f_\theta(\mathbf{X}), \mathbf{Y})$ to quantify the discrepancy between predictions and true labels, which measures how closely f_θ approximates f. The objective is to minimize the risk functional,

$$R(\theta) = \int_{\mathcal{X} \times \mathcal{Y}} L(\mathbf{Y}, f_\theta(\mathbf{X})) dP(\mathbf{X}, \mathbf{Y}). \tag{2.1}$$

Since the true distribution $P(\mathbf{X}, \mathbf{Y})$ is generally unknown, we approximate the expected risk using the empirical distribution from a finite dataset $\mathcal{D}$. This leads to the empirical risk minimization principle [83]. Instead of integrating over the true probability distribution, we approximate it using the empirical distribution,

$$R_N(\theta) = \frac{1}{N} \sum_{i=1}^{N} L(\mathbf{Y}_i, f_\theta(\mathbf{X}_i)). \tag{2.2}$$

This empirical risk serves as a practical approximation of the true risk and forms the basis for learning algorithms.

2.2 Unsupervised learning

Unsupervised learning aims to uncover patterns, structures, or representations from unlabeled data. Unlike supervised learning, where we have input-output pairs $(\mathbf{X}, \mathbf{Y})$, unsupervised learning works solely with inputs $\mathbf{X}$ without any corresponding labels. Given a dataset of N samples,

$$\mathcal{D} = \{\mathbf{X}_i\}_{i=1}^N,$$

where $\mathbf{X}_i \in \mathbb{R}^d$ is a d-dimensional feature vector. Our objective is to discover intrinsic structures, such as clusters, manifolds, or latent representations. We assume that the observed data are sampled from an unknown but fixed probability distribution $P(\mathbf{X})$. The objective of unsupervised learning is to learn a function: $f : \mathbb{R}^d \to \mathbb{R}^k$, where $f \in \mathcal{F}$ belongs to a hypothesis space containing candidate functions, and k is a latent space dimension. The function f is

learned and its objective it to capture hidden structures in the data. Since unsupervised learning operates solely on unlabeled data, both the learning strategy and the choice of loss function must be carefully adapted to the specific learning objective. In this context, we distinguish between two key categories that are particularly relevant to this study: *unsupervised structure learning*, which aims to uncover latent structures or representations within the data, and *density learning*, which focuses on estimating the underlying probability distribution that generated the data.

2.2.1 Unsupervised Structure Learning

Unsupervised structure learning seeks to uncover meaningful representations, latent structures, or geometric patterns within data in the absence of labeled information. By revealing the intrinsic organization of complex datasets, it enables more effective data analysis, visualization, and feature extraction. In the context of this work, two prominent approaches to unsupervised structure learning are particularly relevant: *clustering*, which groups data points based on similarity, and *sparse representation learning*, which identifies compact and informative representations that capture the essential structure of the data.

Clustering techniques aim to group similar data points based on intrinsic similarities. Traditional clustering methods such as k-means and Gaussian Mixture Models (GMMs) rely on distance metrics and probability distributions to cluster data [12, 49]. However, these methods require the number of clusters to be predefined, which is often unknown. In such cases, adaptive clustering techniques are needed to measure the similarity between data points and dynamically determine the clusters [47]. Hierarchical clustering approaches address this challenge by building a nested hierarchy of clusters. The main idea relies on recursively merging pairs of clusters based on a linkage criterion that measures the distance between them. This approach results in a more interpretable structure, as it does not require prior knowledge of the number of clusters [9].

Sparse representation learning is a technique in machine learning and signal processing that aims to represent data using a small number of non-zero coefficients selected from a predefined or learned dictionary. The underlying assumption is that high-dimensional data often lie near a lower-dimensional manifold, allowing for efficient representation using only a few active basis elements.

A common approach to constructing such representations involves regularized least squares (RLS), which promotes sparsity through penalty terms. In applications where the data evolves over time and space, sparse learning methods

have proven particularly useful for scientific discovery. Such methods enable the identification of governing equations or latent functional relationships that drive the observed dynamics.

A notable example is the use of the L_1 regularizer (Lasso), as demonstrated in [71], where a dictionary built from nonlinear functions of the data and its spatial partial derivatives was used to discover partial differential equations (PDEs) from data. In [14], the author introduced the Sequential Thresholded Least Squares (STLS) algorithm as part of the Sparse Identification of Nonlinear Dynamics (SINDy) framework. This method iteratively combines ridge regression with hard thresholding to promote sparsity in the identified model. The details of this approach will be discussed further in Chapter 5.

2.2.2 Density estimation

Density learning aims to estimate the underlying probability distribution $P(\mathbf{X})$ that governs the observed data. Classical approaches, such as kernel density estimation (KDE) and histograms, attempt to approximate the probability density function (PDF) directly from data, typically without imposing strong assumptions on its form. Although these methods offer useful descriptive insights, they are generally limited in their capacity to generate new samples from the learned distribution, which restricts their applicability in generative modeling tasks.

Generative models extend the concept of density estimation by not only estimating the underlying data distribution but also enabling the generation of new data points that follow the same distribution. Advances in deep learning, particularly Variational Autoencoders (VAEs) [40], Generative Adversarial Networks (GANs) [29], Energy-Based Models (EBMs) [55], and diffusion models [32, 75], have significantly improved sample generation. Notably, diffusion models overcome key challenges: misalignment of posterior distributions in VAEs, the computational complexity of Markov chain Monte Carlo (MCMC) in EBMs, and instability in GAN training. A detailed discussion follows in Chapter 3.

2.3 Neural networks

A neural network is a computational model composed of interconnected nodes, or "neurons," inspired by the structure and function of biological neural networks in the brain [67]. While loosely based on neuroscience, modern neural networks are primarily mathematical constructs optimized for function approximation,

pattern recognition, and data-driven learning. A neural network comprises multiple layers that process input data, learning hierarchical representations to approximate a target function [10].

Formally, we let Φ be a neural network that learns a function f_θ from a hypothesis set $\mathcal{F}$. The network consists of layers that process and propagate information through structured transformations. A key aspect of this process is the propagation operator $\mathcal{P}$, which defines how information flows across layers based on the underlying data structure. We define $\sigma : \mathbb{R} \to \mathbb{R}$ as the activation function, and $\Phi^{(l)}$ as the output of the l-th hidden layer. The trainable parameters θ consist of weight matrices $W^{(l)} \in \mathbb{R}^{N_{l-1} \times N_l}$ and bias vectors $b^{(l)} \in \mathbb{R}^{N_l}$ for each layer l. The transformations for the first hidden layer and the l-th hidden layer (for $l > 1$) are given by,

$$\begin{aligned} \Phi^{(1)}(\mathbf{X}, \theta) &= \sigma^{(1)}\left(\mathcal{P}(\mathcal{S}, \mathbf{X})W^{(1)} + b^{(1)}\right), \\ \Phi^{(l)}(\mathbf{X}, \theta) &= \sigma^{(l)}\left(\mathcal{P}(\mathcal{S}, \Phi^{(l-1)}(\mathbf{X}, \theta))W^{(l)} + b^{(l)}\right), \end{aligned} \tag{2.3}$$

where $\mathcal{S}$ encodes structural information, and $\mathcal{P}(\mathcal{S}, \cdot)$ defines the aggregation mechanism applied before the linear transformation.

For fully connected feedforward neural networks (FCNNs), no structural constraints are imposed by $\mathcal{S}$. As a result all neurons in one layer are connected to all other neurons in the next layer. In this case, $\mathcal{P}(\mathcal{S}, \cdot)$ reduces to the identity mapping, $\mathcal{P}(\mathcal{S}, \Phi^{(l-1)}) = \Phi^{(l-1)}(\mathbf{X}, \theta)$, leading to the standard layer update,

$$\Phi^{(l)}(\mathbf{X}, \theta) = \Phi^{(l-1)}(\mathbf{X}, \theta)W^{(l)} + b^{(l)}. \tag{2.4}$$

For message-passing Graph Neural Networks (GNNs), the structure $\mathcal{S}$ is represented by the adjacency matrix $\mathbf{A}$ or graph Laplacian, which encodes the connectivity between nodes in the graph. GNNs update each node's embedding by aggregating features from its neighbors, while also incorporating edge attributes if available. This aggregation and update mechanism enables the network to learn representations that capture the graph's topology [28]. A key example is the Graph Convolutional Network (GCN) [42], where $\mathcal{P}(\mathcal{S}, \Phi^{(l-1)}) = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\Phi^{(l-1)}(\mathbf{X}, \theta)$. Here, $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ represents the adjacency matrix with self-loops added, and $\tilde{\mathbf{D}}$ is the corresponding degree matrix, where each diagonal entry is given by $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. The term $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$ normalizes the adjacency matrix to prevent feature explosion during message passing. Using this propagation rule, the update equation for GCN becomes,

$$\Phi^{(l)}(\mathbf{X}, \theta) = \tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\Phi^{(l-1)}(\mathbf{X}, \theta)W^{(l)} + b^{(l)}. \tag{2.5}$$

By embedding structural information within $\mathcal{P}$, neural networks, particularly GNNs, can leverage underlying data structures for more efficient learning, faster convergence, and improved physical interpretability.

When training a neural network Φ, the goal is to optimize its parameters θ such that the network minimizes a given loss function L on the training dataset. Since Φ is typically differentiable, gradient-based optimization methods can be employed. In its simplest form, gradient descent computes the gradient of the loss over the entire dataset and updates the parameters according to,

$$\theta \leftarrow \theta - \eta \nabla_\theta L, \tag{2.6}$$

where $\eta > 0$ is the learning rate, which determines the step size of the parameter update. However, computing gradients over the full dataset can be computationally expensive and memory-intensive, especially for large datasets. To alleviate this, Stochastic Gradient Descent (SGD) [87] is commonly used, where the gradient is estimated using a small random subset (mini-batch) of the data. Several variants of SGD have been proposed to improve speed of convergence and stability. Notable examples include RMSProp[81], AdaGrad[24], Adam[38].

3. Generative models

Density estimation plays a central role in situations where the underlying data distribution is either unknown or heavily corrupted by noise. In Papers IV–V, we explore the use of generative models to mitigate challenges related to data scarcity and to perform imputation in highly noisy environments. The datasets considered range from biological measurements, such as fingerprint data, to spatiotemporal signals, including traffic flow and air quality time series. This chapter provides a concise overview of generative modeling approaches, with particular emphasis on Generative Adversarial Networks (GANs) and diffusion-based models.

3.1 Generative adversarial network

GANs are a class of generative models introduced by Ian Goodfellow et al. [29]. They rely on a game-theoretic framework where two neural networks, the generator and the discriminator, iteratively compete against each other in order to improve their individual performance.

The generator (G) creates synthetic data that resembles real samples, while the discriminator (D) learns to distinguish real data from generated samples. The training process starts with G mapping a random latent vector z, sampled from a prior distribution (e.g., Gaussian or uniform), to the data space to generate $G(z)$. Meanwhile, D evaluates whether a sample which may come from the dataset or from the generator is real or fake. The adversarial interaction between G and D is formalized through the following objective function,

$$L_{\text{GAN}}(G, D) = E_{x \sim p_r} log[D(x)] + E_{z \sim p_z} log[1 - D(G(x))], \tag{3.1}$$

where p_r represents the real data distribution and p_z denotes the prior distribution in the latent space z. The training process involves solving the minimax

optimization problem,

$$\min_G \max_D L_{GAN}(G, D). \tag{3.2}$$

The objective is for the generator G to produce samples that are indistinguishable from real data by the discriminator D. Goodfellow et al. [29] theoretically demonstrated that, under ideal conditions with infinite model capacity and perfect optimization, this adversarial game reaches a Nash equilibrium (NE) when the generator's output distribution matches the real data distribution. However, in practice, applications, GAN training often suffers from instability and does not reliably converge to this theoretical equilibrium [25].

Similarly, GANs commonly suffer from many other issues such as mode collapse, vanishing gradients, and overall training instability [35]. To address these challenges, researchers have proposed a range of strategies, including alternative loss functions and redesigned network architectures, aimed at improving training stability and enhancing the fidelity of generated data.

3.1.1 Wasserstein GAN

A significant advancement in addressing vanishing gradients and mode collapse in GANs was proposed by Arjovsky, Chintala, and Bottou, who introduced the Wasserstein GAN (WGAN) framework [6, 7]. Unlike traditional GANs that rely on the Jensen–Shannon (JS) divergence, which often leads to unstable training dynamics, WGANs replace the discriminator with a critic that outputs real-valued scores rather than probabilities. By minimizing the Wasserstein distance (also known as Earth Mover's distance) between the real and generated data distributions, WGANs provide a smoother loss landscape and more stable gradients, thereby improving convergence and sample diversity. The WGAN objective function is defined as,

$$L_{WGAN}(G, D) = \mathbb{E}_{x\sim p_r}[D(x)] - \mathbb{E}_{z\sim p_z}[D(G(z))] \tag{3.3}$$

where D is 1-Lipschitz continuous function. To enforce this Lipschitz continuity, the original WGAN paper introduces a simple yet effective technique: weight clipping. After each gradient update, the parameters of D are clamped within a small window, ensuring that the function D remains within the required bounds, thus preserving its Lipschitz continuity. This approach enhances training stability and helps mitigate mode collapse.

However, the original WGAN relies on weight clipping to enforce the 1-Lipschitz constraint on the critic, which can lead to optimization issues such as vanishing or exploding gradients if not carefully tuned. To address this, Wasserstein GAN

with Gradient Penalty (WGAN-GP) [30] introduces a more robust method by penalizing deviations from a unit gradient norm. This gradient penalty (GP) ensures that the critic remains close to 1-Lipschitz without requiring explicit weight clipping, thereby enabling stable training even for deeper or more complex networks.

The objective function for WGAN-GP is given by:

$$L_{WGAN-GP}(G, D) = \mathbb{E}_{x\sim p_r}[D(x)] - \mathbb{E}_{z\sim p_z}[D(G(z))] + \lambda \mathbb{E}_{\hat{x}\sim p_{\hat{x}}}[(||\nabla_{\hat{x}} D(\hat{x})||_2 - 1)^2], \tag{3.4}$$

where $\hat{x} = \epsilon x + (1-\epsilon)G(z)$, with $\epsilon \sim \mathcal{U}(0,1)$, interpolates between real samples $x \sim p_r$ and generated samples $G(z)$. The hyperparameter λ controls the strength of the gradient penalty term and is typically set to a constant ($\lambda = 10$) in practice. This gradient penalty term ensures that the norm of the gradient of D is close to 1, enforcing the 1-Lipschitz condition without requiring weight clipping.

3.1.2 CycleGAN

CycleGAN [88] is a generative framework that enables image-to-image translation between two domains $\mathcal{X}$ and $\mathcal{Y}$ without the need for paired training data. The core concept underlying CycleGAN is *cycle consistency*: if a sample $x \in \mathcal{X}$ is translated into domain $\mathcal{Y}$ and then mapped back into domain $\mathcal{X}$, the result should closely resemble the original sample x. This forward-backward transformation ensures that the learned mappings preserve the essential characteristics of the input data.

This property makes CycleGAN particularly effective in settings where paired data is scarce or unavailable. One example is in the transformation of live fingerprint images into various spoof fingerprint styles. Since spoof fingerprint samples are often limited or costly to obtain, CycleGAN can be used to synthesize spoof variants from readily available live fingerprints, without requiring direct live–spoof pairs during training [79].

In CycleGAN, two generator networks are defined: $G_f : \mathcal{X} \to \mathcal{Y}$ for forward translation, and $G_r : \mathcal{Y} \to \mathcal{X}$ for reverse translation. The core idea of cycle consistency requires that a sample translated to the other domain and back should closely resemble the original input. For a sample $x \in \mathcal{X}$, the forward cycle consistency condition is,

$$x \to G_f(x) \to G_r(G_f(x)) \approx x.$$

Similarly, for a sample $y \in \mathcal{Y}$, the backward cycle consistency condition is,

$$y \to G_r(y) \to G_f(G_r(y)) \approx y.$$

To enforce this behavior during training, CycleGAN introduces a cycle consistency loss, defined as,

$$L_{cycle}(G_f, G_r) = \mathbb{E}_{x \sim p_x}[||G_r(G_f(x)) - x||_1] + \mathbb{E}_{y \sim p_y}[||(G_r(y)) - y||_1], \quad (3.5)$$

where $\|\cdot\|_1$ denotes the L1 norm, promoting accurate reconstruction by penalizing large deviations from the original inputs.

The adversarial loss for mapping from domain $\mathcal{X}$ to domain $\mathcal{Y}$ is defined using a generator G and a discriminator $D_{\mathcal{Y}}$ as,

$$\mathcal{L}_{\text{GAN}}(G, D_{\mathcal{Y}}, \mathcal{X}, \mathcal{Y}) = \mathbb{E}_{y \sim p_{\mathcal{Y}}} \left[\log D_{\mathcal{Y}}(y)\right] + \mathbb{E}_{x \sim p_{\mathcal{X}}} \left[\log \left(1 - D_{\mathcal{Y}}(G(x))\right)\right], \quad (3.6)$$

where $p_{\mathcal{X}}$ and $p_{\mathcal{Y}}$ denote the data distributions in domains $\mathcal{X}$ and $\mathcal{Y}$, respectively.

The full CycleGAN objective combines two adversarial losses (one loss for each domain mapping) with the cycle consistency loss. It is given by,

$$\begin{aligned}\mathcal{L}(G_f, G_r, D_{\mathcal{X}}, D_{\mathcal{Y}}) = \mathcal{L}_{\text{GAN}}(G_f, D_{\mathcal{Y}}, \mathcal{X}, \mathcal{Y}) + \mathcal{L}_{\text{GAN}}(G_r, D_{\mathcal{X}}, \mathcal{Y}, \mathcal{X}) \\ + \lambda \mathcal{L}_{\text{cycle}}(G_f, G_r),\end{aligned} \quad (3.7)$$

where λ is a hyperparameter that controls the relative importance of the cycle consistency loss. The final optimization objective is a minimax game between the generators and discriminators,

$$\min_{G_f, G_r} \max_{D_{\mathcal{X}}, D_{\mathcal{Y}}} \mathcal{L}(G_f, G_r, D_{\mathcal{X}}, D_{\mathcal{Y}}). \quad (3.8)$$

3.2 Generative diffusion models

WGAN seeks to minimize the Wasserstein distance between the real data distribution and the distribution induced by the generator. However, computing this distance directly is intractable in high-dimensional spaces, making its practical approximation computationally demanding. Training WGANs also requires enforcing the 1-Lipschitz constraint on the discriminator, which introduces additional optimization challenges and can lead to instability. In contrast, diffusion models learn the data distribution by reversing a fixed stochastic diffusion process, which progressively corrupts data with noise and then trains a model to

invert this process for sample generation. Current research on diffusion models is primarily organized around two closely related formulations: a) denoising diffusion probabilistic models (DDPMs) [32, 56], which describe the diffusion process in discrete time, and b) score-based generative models (SGMs) [77, 78], which extend this framework to continuous time using stochastic differential equations (SDEs). Both DDPMs and SGMs operate in continuous vector spaces, enabling smooth, differentiable transitions between data states. This continuous formulation leads to more stable gradients during training, avoiding the abrupt shifts in optimization that are common in GANs. Recent theoretical advances have shown that diffusion models can be interpreted as solving regularized optimal transport problems, specifically through the lens of Schrödinger bridges [20, 23]. A Schrödinger bridge is an entropic interpolation between distributions. This regularization results in a smoother optimization landscape and improved training stability.

3.2.1 Denoising diffusion probabilistic models (DDPMs)

A denoising diffusion probabilistic model (DDPM) generates data by simulating the reversal of a gradual noise corruption process. The method is built on two sequential Markov chains. First, a fixed forward process adds noise to the data over time, and second, a learned reverse process that reconstructs data from noise. In the forward process, data samples are progressively corrupted by Gaussian noise across a finite number of time steps. This process is carefully designed to map any data distribution into a simple, tractable prior distribution (typically a standard Gaussian distribution) while maintaining mathematical tractability. The reverse process is parameterized by a neural network that learns to denoise the corrupted inputs, approximating the reverse transitions of the forward chain. New data samples are generated by first drawing a random vector from the prior distribution and then gradually denoising it step by step through the learned reverse Markov chain.

Formally, we let $x_0 \sim q_r(x_0)$ denote a sample from the real data distribution. The forward diffusion process is defined as a Markov chain that gradually perturbs the data with Gaussian noise. At each time step t, the Gaussian transition probability is given by,

$$q(x_t|x_{t-1}) := \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t \boldsymbol{I}), \tag{3.9}$$

where β_1, ...,β_T is a predefined sequence of noise variances that control the rate of diffusion. The full forward process from x_0 to x_T is given by the composition

of these transitions,

$$\begin{aligned} q(x_{0:T}) &:= q(x_0)q(x_{1:T}|x_0), \\ q(x_{1:T}|x_0) &:= \prod_{t=1}^{T} q(x_t|x_{t-1}). \end{aligned} \tag{3.10}$$

Here we let $\alpha_t := 1-\beta_t$ and define the cumulative product $\bar{\alpha}_t := \prod_{s=0}^{t} \alpha_s$. Then, the marginal distribution of x_t given x_0 has the closed-form,

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\boldsymbol{I}). \tag{3.11}$$

This allows efficient sampling of x_t directly from x_0 using the reparameterization trick [39, 58],

$$x_t = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, \boldsymbol{I}). \tag{3.12}$$

As $t \to T$ and $\bar{\alpha}_t \to 0$, the sample x_t becomes increasingly noisy. In the limit, the final latent variable x_T approaches a standard Gaussian. The corresponding marginal distribution is given by,

$$q(x_T) := \int q(x_T \mid x_0)\, q(x_0)\, dx_0 \approx \mathcal{N}(0, \boldsymbol{I}). \tag{3.13}$$

The reverse process in DDPM aims to gradually remove noise from a sample $x_T \sim \mathcal{N}(0, \boldsymbol{I})$, ultimately recovering a sample x_0 from the original data distribution. An important theoretical result from [27, 74] shows that when the forward process uses sufficiently small noise levels β_t, the reverse-time process is also a Markov process and has a similar form to the forward process. Specifically, the reverse conditional distribution $p(x_{t-1}|x_t)$ is also Gaussian. Thus, the reverse process can be modeled by parameterizing its mean and variance with a neural network. We define the learnable reverse transition kernel as,

$$p_\theta(x_{t-1}|x_t) := \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)), \tag{3.14}$$

where μ_θ and Σ_θ are the neural network outputs representing the mean and variance at each time step t. The full reverse process is then defined as,

$$p_\theta(x_{0:T}) := p(x_T) \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t), \tag{3.15}$$

where $p(x_T) \approx \mathcal{N}(0, \boldsymbol{I})$ is the prior distribution that approximates the terminal distribution of the forward process $q(x_T)$.

The training objective is to make this generative process match the data distribution. This is done by minimizing the KL divergence between the forward

process $q(x_{0:T})$ and the model $p_\theta(x_{0:T})$. Using variational inference, we derive the evidence lower bound (ELBO),

$$\begin{aligned} p_\theta(x_0) &= \int p_\theta(x_{0:T})\, dx_{1:T} \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] \\ &\geq \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right], \end{aligned} \tag{3.16}$$

where the final inequality follows from Jensen's inequality. We now expand the integrand,

$$\begin{aligned} &\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}\right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log p(x_T) + \sum_{t=2}^{T}\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})} + \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log p(x_T) + \sum_{t=2}^{T}\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)}\frac{q(x_{t-1}|x_0)}{q(x_t|x_0)} + \log \frac{p_\theta(x_0|x_1)}{q(x_1|x_0)}\right] \\ &= \mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p_\theta(x_T)}{q(x_T|x_0)} + \sum_{t=2}^{T}\log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t,x_0)} + \log p_\theta(x_0|x_1)\right]. \end{aligned} \tag{3.17}$$

Finally, taking the negative and interpreting each term as a KL divergence, we obtain,

$$\begin{aligned} -\log p_\theta(x_0) \leq \mathbb{E}_q\Bigg[&\underbrace{D_{KL}(q(x_T|x_0)||p(x_T))}_{L_T} + \\ &\sum_{t=2}^{T}\underbrace{D_{KL}(q(x_{t-1}|x_t,x_0)||p_\theta(x_{t-1}|x_t))}_{L_{t-1}}\underbrace{-\log p_\theta(x_0|x_1)}_{L_0}\Bigg], \end{aligned} \tag{3.18}$$

where L_T is the prior loss, L_0 is the reconstruction loss, and the terms $L_{1:T-1}$ represent the divergence between the forward posterior and the reverse model at each step. To further simplify the training, [32] adopts a reparameterization approach where the model predicts the noise ϵ used to generate x_t from x_0 (see Eq. 3.14). This leads to the following simplified training objective, often referred to as L_{simple},

$$L_{simple} := \mathbb{E}_{t\sim\mathcal{U}[1,T],x_0,\epsilon}\left[\lambda(t)||\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon)||^2\right], \tag{3.19}$$

where $\lambda(t)$ is a positive weighting function, x_t is constructed via Eq. (3.12), $\mathcal{U}[1,T]$ is a uniform distribution over the set $\{1,2,...,T\}$, and ϵ_θ is a neural network that predicts the noise ϵ given x_t and t.

3.2.2 Score-based generative models (SGMs)

Score-based generative models (SGMs), also known as diffusion probabilistic models in continuous time, generalize DDPMs by using stochastic differential equations (SDEs) to model the forward (noising) and reverse (denoising) dynamics of data [78]. In this framework, the data is gradually perturbed into noise via a continuous-time diffusion process governed by the Itô SDE,

$$dx = f(x,t)dt + g(t)dw, \tag{3.20}$$

where $f(x,t)$ is the drift coefficient, $g(t)$ is the diffusion coefficient, and w is a standard Wiener process. For DDPMs, the forward process is a discretization of the following variance-preserving SDE (VP-SDE),

$$dx = -\frac{1}{2}\beta(t)xdt + \sqrt{\beta(t)}dw, \tag{3.21}$$

which ensures the marginal variance of $x(t)$ is preserved over time. The discrete-time DDPM parameters β_t are related to the continuous-time form by the scaling $\beta(t/T) = T\beta_t$, which connects the discrete and continuous formulations as the number of steps $T \to \infty$. A key theoretical insight by Anderson [3] shows that any forward diffusion process defined by Eq. (3.20) can be exactly reversed in time, under mild regularity assumptions. The reverse-time process is governed by the reverse-time SDE,

$$dx = [f(x,t) - g(t)^2\nabla_x \log q_t(x)]dt + g(t)d\bar{w}, \tag{3.22}$$

where $\bar{w}$ denotes a standard Wiener process defined with respect to backward time, and q_t denotes the distribution of x_t in the forward process.

To approximate the reverse process, we parameterize a time-dependent neural network $s_\theta(x_t,t)$ to estimate the score function $\nabla_x \log q_t(x)$. This leads to the following training objective,

$$\mathbb{E}t \sim \mathcal{U}(0,T),\ x_0 \sim q_r(x_0),\ x_t \sim q(x_t|x_0)\left[\lambda(t), |s\theta(x_t,t) - \nabla_x \log q_t(x_t|x_0)|^2\right], \tag{3.23}$$

where $\mathcal{U}[0,T]$ denotes the uniform distribution over $[0,T]$. Here, $\lambda(t)$ is a positive weighting function that controls the relative importance of the score-matching

loss at different diffusion times t. It can be chosen to balance gradient magnitudes across time steps, improve stability, or emphasize specific temporal regions during training. In practice, $\lambda(t)$ is often set to $\lambda(t) = \sigma_t^2$ or a related function, where σ_t is the noise scale at time t, as this helps normalize the loss when the score function varies significantly over time. In practice, since the conditional distribution $q_t(x_t|x_0)$ is often Gaussian, the score $\nabla_x \log q_t(x_t|x_0)$ has a closed-form expression [77]. More details can be found in [78].

4. Hybrid modelling and parameter estimation

Current generative models are primarily designed to approximate the data distribution. While they are effective at capturing short-term patterns and generating plausible outputs, they often fail to uncover the underlying causal or physical relationships between variables. As a consequence, especially in the context of time-dependent dynamical systems, these models tend to accumulate small errors at each time step, which leads to a progressive drift from the true system dynamics and eventually results in physically inconsistent or scientifically implausible predictions. To address these limitations, there is increasing interest in hybrid modeling approaches that embed known physical laws into machine learning frameworks. These approaches aim not only to fit data but also to recover interpretable and generalizable models grounded in the structure of the underlying system. In this chapter, we provide an overview of numerical schemes for time integration, introduce their extension via Neural Ordinary Differential Equations (Neural ODEs), and demonstrate how these tools can be used for modeling and parameter estimation in complex dynamical systems.

4.1 Time integration

Let the dynamical system be defined by,

$$\begin{aligned} \frac{d}{dt}x(t) &= f(t, x(t)), \quad t \in (t_0, T] \\ x(t_0) &= x_0, \end{aligned} \tag{4.1}$$

where $x(t) \in \mathbb{R}^d$ is the state variable, $f : [t_0, T] \times \mathbb{R}^d \to \mathbb{R}^d$ is a, possibly nonlinear, function governing the dynamics, and x_0 is the initial condition. In

many practical applications, especially when f is nonlinear or derived from experimental data, analytical solutions to (4.1) are not available. Numerical integration methods are therefore essential for approximating the evolution of $x(t)$ over time. The choice of numerical integrator can have a major impact on accuracy, stability, and computational cost.

The Euler method is the most basic numerical scheme for solving initial value problems. We assume the time interval $[t_0, T]$ is divided into N equally spaced points, defined by $t_k = t_0 + kh$ for $k = 0, 1, ..., N$, where the step size is given by $h = \frac{T-t_0}{N}$. Suppose that system (4.1) admits a unique solution $x(t)$ and that $x(t)$ is twice continuously differentiable on $[t_0, T]$. Then, by Taylor's theorem, for each $k = 0, 1, ..., N-1$, we have,

$$x(t_{k+1}) = x_{t_k} + (t_{k+1} - t_k)x'(t_k) + \frac{(t_{k+1} - t_k)^2}{2}x''(\xi_i), \tag{4.2}$$

for some $\xi_k \in (t_k, t_{k+1})$. Since $x(t)$ satisfies system (4.1) and $h = t_{k+1} - t_k$, this expression becomes,

$$x(t_{k+1}) = x_{t_k} + hf(t_k, x(t_k)) + \frac{h^2}{2}x''(\xi_k). \tag{4.3}$$

The explicit Euler method constructs an approximation $w_k \approx x(t_k)$ by neglecting the remainder term in the Taylor expansion. The resulting iterative scheme is,

$$\begin{aligned} w_0 &= x_0, \\ w_{k+1} &= w_i + hf(t_k, w_k), \quad k = 0, 1, ..., N-1. \end{aligned} \tag{4.4}$$

To analyze the error bound of Euler's method, we subtract Eq. (4.4) from (4.3),

$$x(t_{k+1}) - w_{k+1} = x_{t_k} - w_k + h(f(t_k, x(t_k)) - f(t_k, w_k)) + \frac{h^2}{2}x''(\xi_i). \tag{4.5}$$

Taking norms, we obtain,

$$|x(t_{k+1}) - w_{k+1}| \leq |x_{t_k} - w_k| + h|f(t_k, x(t_k)) - f(t_k, w_k)| + \frac{h^2}{2}|x''(\xi_k)|. \tag{4.6}$$

Assume that f satisfies a Lipschitz condition with constant C, and that $|x''(\xi_k)| \leq M$ for some constant M. Then,

$$|x(t_{k+1}) - w_{k+1}| \leq (1 + hC)|x_{t_i} - w_k| + \frac{h^2 M}{2}. \tag{4.7}$$

By applying Lemma 5.8 from [17], the error bound follows that,

$$|x(t_{k+1}) - w_{k+1}| \leq \frac{hM}{2C}(e^{(t_{k+1}-t_0)C} - 1), \quad k = 0, ..., N-1. \tag{4.8}$$

One way to improve the accuracy of Euler's method is to include higher-order terms from the Taylor series expansion. However, high-order Taylor methods require computing higher derivatives of $f(t, x)$, which can be computationally expensive and impractical for many applications. Runge-Kutta (RK) methods offer a practical alternative: they achieve higher-order accuracy without the need to compute higher-order derivatives explicitly. A general Runge-Kutta method updates the solution according to,

$$w_{k+1} = w_i + \phi(t_k, x_k, h)h, \tag{4.9}$$

where $\phi(t_k, x_k, h)$ is known as the increment function, typically expressed as,

$$\phi(t_k, w_k, h) = \sum_{i=1}^{s} b_i K_i, \tag{4.10}$$

with stage values,

$$K_i = f(t_i + c_i h, x_i + h \sum_{j=1}^{i-1} a_{ij} K_j), \quad i = 1, ..., s. \tag{4.11}$$

The method is fully specified by the number of stages s, and the coefficients a_{ij}, b_i and c_i which are typically presented in a Butcher tableau [62]. By appropriately choosing these coefficients, one can construct RK methods that satisfy specific order conditions and achieve the desired level of accuracy.

4.2 Neural ODE

As neural networks become deeper, they are theoretically more expressive. However, in practice, very deep networks often suffer from issues such as vanishing gradients, reduced accuracy, and slow convergence. Residual networks (ResNets), introduced by He *et al.* [31], address these problems by introducing skip connections, which allow layers to learn residual functions rather than full transformations. A typical residual block can be written as,

$$x_{k+1} = x_k + \Phi(x_k, \theta), \tag{4.12}$$

where x_k is the input to l-th block, and $\Phi(\cdot, \theta)$ is a nonlinear transformation parameterized by θ. Interestingly, the residual update resembles Euler's method for numerically integrating ODEs. This observation motivated the development of Neural ODEs [19], where the residual block is replaced by a continuous-time differential equation,

$$\frac{d}{dt}x(t) = \Phi(x(t), t, \theta), \tag{4.13}$$

where $t \in [0, T]$ indexes the network depth. With this formulation, numerical solvers such as Runge–Kutta methods can be employed to effectively model neural networks with continuous or infinitely deep layers.

Most commonly used activation functions, such as ReLU, Leaky ReLU, SoftPlus, Tanh, Sigmoid, ArcTan, and Softsign, as well as operations like max-pooling, have Lipschitz constants equal to 1. Additionally, layers such as dropout, batch normalization, and other pooling methods have well-defined Lipschitz constants [70]. Given these properties, the neural network is typically Lipschitz continuous, and therefore, by Picard's existence theorem [18], the existence and uniqueness of solutions for Eq. (4.13) is guaranteed.

4.3 Parameter estimation

Consider the Susceptible-Infectious-Removed (SIR) model, a fundamental system in epidemiology,

$$\begin{aligned} \frac{dS}{dt} &= -\frac{\beta}{N} IS, \\ \frac{dI}{dt} &= \frac{\beta}{N} IS - \gamma I, \\ \frac{dR}{dt} &= \gamma I, \end{aligned} \tag{4.14}$$

where S, I, R represent the susceptible, infectious, and recovered populations, respectively. Estimating the basic reproduction number $R_0 = \frac{\beta}{\gamma}$ is crucial for evaluating the effectiveness of mitigation strategies and informing public health decisions.

In practice, real-world data is often noisy and complex, and many extensions of the SIR model have been proposed to better capture these complexities. To estimate parameters, classical methods such as the Extended Kalman Filter (EKF), least squares, and Bayesian inference are commonly used [48, 52, 76]. These approaches typically assume that the model structure is known. However, in many real-world scenarios, the exact governing equations may be unknown or only partially specified. In such cases, hybrid modeling approaches, which combine mechanistic models with data-driven components, offer a flexible and powerful alternative. Within the Neural ODE framework, various optimization method can be used with backpropagation via adjoint methods or automatic differentiation.

For example, in [22], a time-varying quarantine strength term was introduced to model the effects of local policy changes. This term, difficult to define explicitly,

was approximated using a neural network and trained via a Neural ODE solver. This approach, known as a Universal Differential Equation (UDE) [63], embeds neural components within a differential equation to learn unknown dynamics from data while preserving known mechanistic structure. In Paper II [45], we extend this idea by estimating a time-varying reproduction number $R_0(t)$, as well as parameters in an additional integral term, highlighting the flexibility and expressiveness of the UDE framework in real-world modeling tasks.

5. Dynamical system identification

A wide range of natural and human-driven phenomena are governed by dynamical systems, which are typically described using differential or difference equations that model the evolution of state variables over time. In Chapter 4, we explored parameter estimation techniques under the assumption that the system is fully or partially known. However, a more challenging problem arises when the underlying system is entirely unknown. In that respect we ask the question: how can we identify both the governing equations and parameter values from discrete time-series data? We tackle this question head on in this chapter by focusing on regularization-based methods for dynamical system identification.

5.1 Sparse identification of nonlinear dynamics

Consider an autonomous continuous-time dynamical system described by the following differential equation,

$$\dot{x}(t) = f(x(t)), \tag{5.1}$$

where $x \in \mathbb{R}^d$ is the observable state vector at time t, $\dot{x}$ is its time derivative, and $f : \mathbb{R}^d \to \mathbb{R}^d$ is an unknown nonlinear function that governs the system's dynamics. The goal of system identification in this context is to recover the functional form of f from a finite set of time-series measurements. Assuming that f can be approximated by a linear combination of a set of nonlinear candidate functions, system (5.1) can be rewritten as,

$$\dot{x}(t) = f(x(t)) = \Xi^\top \Theta(x(t))^\top, \tag{5.2}$$

where $\Theta(\cdot) : \mathbb{R}^{1\times d} \to \mathbb{R}^{1\times n_\theta}$ is a library of n_θ candidate nonlinear functions (e.g., polynomials, trigonometric functions, or other symbolic expressions), and

$\Xi = [\xi_1, ..., \xi_d] \in \mathbb{R}^{n_\theta \times d}$ is a matrix of coefficients. In many physical and biological systems, such as the epidemic model discussed in system (4.14), the true dynamics are typically driven by a relatively small number of dominant mechanisms. Although the function library Θ may be large and expressive, only a few basis functions are expected to be active for each component of f. This motivates the assumption that the coefficient matrix Ξ is sparse.

Let the discrete-time measurements of the state be denoted as,

$$\boldsymbol{X} = \begin{bmatrix} x^T(t_1) \\ x^T(t_2) \\ \vdots \\ x^T(t_n) \end{bmatrix} = \begin{bmatrix} x_1(t_1) & \dots & x_d(t_1) \\ x_1(t_2) & \dots & x_d(t_2) \\ \vdots & & \vdots \\ x_1(t_n) & \dots & x_d(t_n) \end{bmatrix}, \tag{5.3}$$

and the corresponding time derivatives as,

$$\dot{\boldsymbol{X}} = \begin{bmatrix} \dot{x}^T(t_1) \\ \dot{x}^T(t_2) \\ \vdots \\ \dot{x}^T(t_n) \end{bmatrix} = \begin{bmatrix} \dot{x}_1(t_1) & \dots & \dot{x}_d(t_1) \\ \dot{x}_1(t_2) & \dots & \dot{x}_d(t_2) \\ \vdots & & \vdots \\ \dot{x}_1(t_n) & \dots & \dot{x}_d(t_n) \end{bmatrix}. \tag{5.4}$$

We then define a nonlinear feature matrix $\Theta(X) \in \mathbb{R}^{n \times n_\theta}$ by applying the function library element-wise to each row of $\boldsymbol{X}$,

$$\Theta(X) = \begin{bmatrix} 1 & \boldsymbol{X} & \boldsymbol{X}^{P_2} & \boldsymbol{X}^{P_3} & \dots & \sin(\boldsymbol{X}) & \cos(\boldsymbol{X}) & \dots \end{bmatrix}. \tag{5.5}$$

Here, $\boldsymbol{X}^{P_2}$, $\boldsymbol{X}^{P_3}$, etc., are higher polynomials, where $\boldsymbol{X}^{P_2}$ denotes the quadratic nonlinearities in the state,

$$\boldsymbol{X}^{P_2} = \begin{bmatrix} x_1^2(t_1) & x_1(t_1)x_2(t_1) \dots & x_2^2(t_1) & \dots & x_d^2(t_1) \\ x_1^2(t_2) & x_1(t_2)x_2(t_2) \dots & x_2^2(t_2) & \dots & x_d^2(t_2) \\ \vdots & \vdots & \vdots & & \vdots \\ x_1^2(t_n) & x_1(t_n)x_2(t_n) \dots & x_2^2(t_n) & \dots & x_d^2(t_n) \end{bmatrix}. \tag{5.6}$$

The dynamical system (5.1) can therefore be written in matrix form as,

$$\dot{\boldsymbol{X}} = \Theta(\boldsymbol{X}^T)\Xi, \tag{5.7}$$

For multivariate systems, Eq. (5.7) can be solved column-wise by treating each component of the vector field independently. That is, for each state variable $\boldsymbol{X}_j$ we solve the least-squares regression problem,

$$\min_{\Xi_j} ||\dot{\boldsymbol{X}}_j - \Theta(\boldsymbol{X}^T)\Xi_j||_2^2, \tag{5.8}$$

where $\dot{\boldsymbol{X}}_j$ and Ξ_j are the j-th columns of $\dot{\boldsymbol{X}}$ and Ξ, respectively. The time derivative matrix $\dot{\boldsymbol{X}}$ is typically not available directly and must be approximated numerically from the discrete time-series data. Common numerical differentiation methods include central difference schemes, polynomial interpolation, and spline-based approaches [44, 82], each of which balances trade-offs between accuracy and noise sensitivity.

However, in practice, the system described by Eq. (5.7) is often affected by measurement noise, which can introduce significant challenges for accurate derivative estimation and stable identification of sparse coefficients. To mitigate these issues, sparse regression methods are employed, where a regularization term is added to the optimization problem to promote sparsity and enhance robustness to noise [8]. The general form of the regularized objective function is,

$$\min_{\Xi_j} ||\dot{\boldsymbol{X}}_j - \Theta(\boldsymbol{X}^T)\Xi_j||_\beta^\alpha + \varphi||\Xi_j||_\delta^\gamma, \tag{5.9}$$

where $||x||_p = (\sum_i |x_i|^p)^{1/p}$ denotes the L_p norm of a vector x, $\varphi \in \mathbb{R}$ is a regularization parameter controlling the trade-off between data fidelity and sparsity, and $\alpha, \beta, \gamma, \delta \in \mathbb{R}$ are constants (different choices of these constants recover well-known methods). For example, ridge regression corresponds to $\alpha, \beta, \gamma, \delta = 2$ which penalizes large coefficients but does not enforce sparsity [34]. The LASSO (Least Absolute Shrinkage and Selection Operator) method arises when $\alpha, \beta = 2$, and $\gamma, \delta = 1$, promoting sparsity through an L_1 penalty [71, 80].

Alternatively, rather than solving a single regularized optimization problem, the Sparse Identification of Nonlinear Dynamics (SINDy) framework proposed in [16] enforces sparsity through an iterative thresholding strategy known as the Sequential Thresholded Least Squares (STLS) algorithm. In this approach, the coefficients are obtained by solving a standard regression problem of the form,

$$\min_{\Xi_j} ||\dot{\boldsymbol{X}}_j - \Theta(\boldsymbol{X}^T)\Xi_j||_\beta^\alpha, \quad \text{with } |\xi_{l,j}| \geq \lambda, \tag{5.10}$$

where λ is a user-defined threshold parameter. The method begins by estimating the coefficients via least-squares regression, followed by thresholding: coefficients with magnitude below λ are set to zero. The model is then re-fitted using only the remaining active terms. This process is repeated iteratively until convergence, yielding a sparse, interpretable model that captures the essential dynamics. It was later shown that this iterative procedure converges to a solution of the form (5.7) with $\alpha = \beta = 2$ and $\gamma = 0$ [86]. An extension of the STLS algorithm includes an additional L_2 penalty term, resulting in the sequential

threshold ridge regression (STRidge) method [69], defined as,

$$\min_{\Xi_j} ||\dot{\boldsymbol{X}}_j - \Theta(\boldsymbol{X}^T)\Xi_j||_2^2 + \varphi||\Xi_j||_2^2, \quad \text{with } |\xi_{l,j}| \geq \lambda. \tag{5.11}$$

Although STLS and its variants perform well in many scenarios, their performance can degrade in the presence of significant noise and limitation of datasets. To enhance robustness, Bayesian regularization methods have been proposed [57, 60], which incorporate prior distributions to better manage uncertainty and noise in the data. Additionally, the use of resampling techniques has been shown to improve model stability. For instance, the Ensemble-SINDy approach [26] leverages bootstrap resampling to reduce sensitivity to noise. Furthermore, in Paper III [47], we introduce an alternative resampling strategy which can further enhance robustness and accuracy in sparse model recovery.

5.2 Model selection

A central challenge in the SINDy framework is model selection, which refers to the process of identifying the subset of terms in the function library $\Theta(\boldsymbol{X})$ that are most relevant for describing the system's dynamics. Different choices of sparsity threshold λ, regularization weights φ, or function library composition can lead to different candidate models, raising the key question: which model should be chosen? The objective is to identify a model that accurately captures the essential dynamics while keeping the complexity as low as possible. Since the function library is typically overcomplete, it is important to enforce sparsity in the coefficient matrix Ξ in order to avoid overfitting and improve interpretability. To aid in automatic model selection, information-theoretic criteria such as the Akaike Information Criterion (AIC), its corrected version AIC_c, and the Bayesian Information Criterion (BIC) have been widely used [4]. These criteria quantify the trade-off between model fit and complexity by penalizing the number of active terms in the model. Several studies have successfully applied these methods in the context of SINDy to systematically select the most informative and robust model from a pool of candidates [47, 50, 51, 54].

The AIC, originally proposed by Akaike [1, 2], provides a relative estimate of information loss across competing models by balancing model complexity and goodness of fit. For a given candidate model, the AIC is defined as,

$$AIC = -2\ln \mathbf{L}(\hat{\Xi}|x) + 2k, \tag{5.12}$$

where $\hat{\Xi}$ denotes the estimated coefficients of the model, k is the number of free parameters, and $\mathbf{L}(\hat{\Xi}|x)$ is the likelihood of the observed data given the model evaluated at $\hat{\Xi}$.

However, when the sample size is small relative to the number of parameters, the AIC may exhibit a tendency to overfit. To address this issue, a corrected version known as the second-order AIC (AIC_c) was introduced. It incorporates an additional penalty term to account for finite-sample bias,

$$AIC_c = AIC + \frac{2k(k+1)}{n-k+1}, \tag{5.13}$$

where n is the number of observations.

In practice, a common choice for the likelihood function in regression problems is based on the residual sum of squares (RSS), defined as:

$$RSS = \sum_{i=1}^{n} (x(t_i) - \hat{x}(t_i))^2, \tag{5.14}$$

where $x(t_i)$ are the observed values and $\hat{x}(t_i)$ are the corresponding predictions from the candidate model. Higher AIC or AIC_c scores indicate models that either include unnecessary parameters or fail to adequately fit the data. Thus, models with lower AIC values are preferred, as they strike a better balance between goodness-of-fit and model simplicity.

The AIC was later modified by G. Schwarz to yield the Bayesian Information Criterion (BIC) [73], which introduces a stronger penalty on model complexity. BIC is defined as,

$$BIC = -2\ln(\mathbf{L}) + k\ln(n). \tag{5.15}$$

In comparison to AIC, BIC imposes a heavier penalty on models with a larger number of parameters, which often leads to more conservative model choices. While some studies argue in favor of AIC or AIC_c due to their theoretical underpinnings and better performance in certain scenarios [4, 85], our work in Paper III [47] found that BIC yielded higher accuracy in model selection.

6. Research

In this chapter, we summarize the main findings from Papers I–V and relate them to the theoretical concepts introduced in the previous chapters. We also provide a perspective on possible future research directions for each study.

6.1 Paper I

To model time-dependent data arising from systems governed by differential equations coupled with microscopic stochastic process, we consider the following coupled equations,

$$\begin{aligned} \tfrac{d}{dt}\boldsymbol{X} &= \tfrac{1}{\tau}F(\boldsymbol{X}, \bar{\sigma}) \\ \tfrac{d}{dt}\mathbb{E}f(\boldsymbol{X}, \sigma) &= \mathbb{E}\mathfrak{L}f(\boldsymbol{X}, \sigma), \end{aligned} \tag{6.1}$$

where $\mathbf{X}$ is the state vector, σ is the microscopic stochastic process defined on a spatial lattice $\mathcal{L}$, $\bar{\sigma} = \frac{1}{N}\sum_{x\in\mathcal{L}}\sigma(x)$ is the spatial average of σ, τ is a characteristic time, $\mathbb{E}$ represents the expected value, F is a function defining the ODE dynamics, f is a test function, and $\mathfrak{L}$ is the generator of the stochastic process σ. In practice, the available data consists of a clean time series $\boldsymbol{X}$ and a highly noisy macroscopic spatial average $\bar{\sigma}$. While frameworks like Neural ODEs or SDEs provide general modeling tools for such problems, their training becomes particularly challenging when confronted with high levels of noise.

In Paper I [46], we tackle this challenge by approximating the dynamics described in (6.1) with a separable stochastic differential system containing unknown components. We begin by learning the vector field that governs the evolution of both $\boldsymbol{X}$ and $\bar{\sigma}$ through neural networks, denoted N_{θ_1} and N_{θ_2}, respectively. The networks are trained on time derivatives approximated numerically and treated as ground truth. While this approximation could also be achieved using classical regression techniques, the neural network approach offers greater flexibility in capturing nonlinear dependencies. Based on the learned dynamics, we estim-

ate the residual noise in the microscopic stochastic process. Assuming Gaussian noise, we use an additional regression model H_ϕ to estimate its variance, yielding the update scheme,

$$\begin{aligned} \boldsymbol{X}_{i+1} &= \boldsymbol{X}_i + N_{\theta_1}(\vec{X}_i, \bar{\sigma}_i)dt, \\ \bar{\sigma}_{i+1} &= \bar{\sigma}_i + N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i)dt + H_\phi(\boldsymbol{X}_i, \bar{\sigma}_i)\sqrt{dt}\epsilon, \end{aligned} \tag{6.2}$$

where ϵ denotes standard Gaussian noise. However, the underlying stochasticity originates from a non-Gaussian jump process, making the Gaussian assumption suboptimal. To capture the true noise structure, we introduce an auxiliary neural network K_φ that approximates the cumulative distribution function (CDF) of the noise using inverse transform sampling. A standard uniform random variable ν is used as input to generate noise samples consistent with empirical distributions, resulting in the update rule,

$$\begin{aligned} \boldsymbol{X}_{i+1} &= \boldsymbol{X}_i + N_{\theta_1}(\vec{X}_i, \bar{\sigma}_i)dt, \\ \bar{\sigma}_{i+1} &= \bar{\sigma}_i + N_{\theta_2}(\boldsymbol{X}_i, \bar{\sigma}_i)dt + K_\varphi(\boldsymbol{X}_i, \bar{\sigma}_i, \nu)dt^\alpha, \end{aligned} \tag{6.3}$$

where ν is a standard uniform random variable and α is a scaling exponent estimated using multifractal detrended fluctuation analysis, a method commonly used to infer the Hurst exponent in fractional Brownian motion.

We evaluated the effectiveness of these methods for modeling microscale or subgrid stochasticity in dynamical systems evolving over long time horizons. Both formulations, given in Equations (6.2) and (6.3), outperform conventional approaches such as LSTMs, vector autoregression, and Neural SDEs in terms of accuracy and computational efficiency. Notably, the non-Gaussian version (6.3) also successfully predicts rare events such as those observed in saddle-node bifurcation scenarios.

Despite these promising results, the proposed approach has limitations. In particular, it requires finely sampled input data to maintain numerical stability and minimize errors, which may be prohibitive in high-dimensional systems due to the exponential growth in data requirements. In future work, we aim to enhance the generalization capabilities of the proposed approach by incorporating Gaussian noise during pretraining. We also plan a deeper investigation into how numerical derivative approximations, $\boldsymbol{X}'_{i+1} = \frac{\boldsymbol{X}_{i+1} - \boldsymbol{X}_i}{t_{i+1} - t_i}$, influence the stability and accuracy of the method. This will include a comprehensive analysis of error propagation from both theoretical and numerical perspectives.

6.2 Paper II

Accurate long-term forecasting is essential for understanding multi-wave pandemics such as COVID-19. However, the classical SIR model (4.14) is inadequate for this purpose. To address this limitation, [11] proposed an extended SIR model that incorporates the concept of immune memory decay,

$$\begin{aligned} \dot{S} &= -\beta SI + \gamma \int_0^t K(t-\tau) I(\tau) d\tau, \\ \dot{I} &= \beta SI - \gamma I, \\ \dot{R} &= \beta I - \gamma \int_0^t K(t-\tau) I(\tau) d\tau. \end{aligned} \tag{6.4}$$

Here, K is a probability density function (PDF) modeling immune memory decay. We define the function $z(t)$ to describe the return of individuals from the removed to the susceptible compartment over time. It is given by,

$$z(t) = \gamma \int_0^t K(t-\tau)\, I(\tau)\, d\tau.$$

While the modified model captures important biological dynamics, [11] does not provide a strategy for parameter estimation. Neural ODE methods, introduced in Chapter 4, offer a natural framework for hybrid dynamical systems, but the convolution term $z(t)$ in Eq. (6.4) poses a challenge: it involves an intractable integral over past and current values of the dependent variable, which is incompatible with standard Neural ODEs.

In Paper II [45], we extend the Neural ODE framework to address Volterra integro-differential equations such as Eq. (6.4). Unlike the delta distribution used in [11], which poses numerical challenges, we model the memory kernel K as a Gaussian distribution for improved tractability. To further enhance the efficiency of parameter estimation, we incorporate Bayesian optimization into the training process. For numerical integration, we adopt a second-order Runge-Kutta scheme (also known as the modified Euler method),

$$\begin{aligned} x_{k+1} &= x_k + hK_2 \\ K_1 &= f(t_k, x_k) \\ K_2 &= f(t_k + \frac{h}{2}, x_k + \frac{h}{2} K_1). \end{aligned} \tag{6.5}$$

This scheme improves the numerical accuracy over Euler's method while keeping the computational cost relatively low.

More broadly, we introduce a strategy that integrates Bayesian optimization with classical Runge-Kutta solvers to extend Neural ODEs for systems with memory, such as the modified SIR model. This framework enables the direct estimation of time-varying parameters from data through a neural network.

Our results demonstrate that the proposed approach accurately captures both the mean and variance of the memory kernel K, and produces realistic estimates of the effective reproduction number. In addition, the model successfully predicts the timing and amplitude of epidemic wave peaks. We validate the framework on both synthetic datasets and real-world epidemiological data from Mexico, South Africa, and South Korea. Across all cases, the model reliably forecasts multi-wave epidemic dynamics and infers critical parameters of the underlying SIR model.

One key limitation of the proposed approach arises when the number of infections is small relative to the total population. In such cases, the susceptible (S) and removed (R) compartments remain nearly constant, making it difficult to learn dynamic parameters such as the mean of K, as they may not converge during training. Future work will explore several directions to overcome this limitation. Improved data preprocessing may help mitigate the impact of sparse infection data. Additionally, techniques like normalizing flows could enhance estimates of the underlying distribution from which the infection data is drawn. We also plan to investigate hybrid strategies that integrate classical approaches, such as Markov Chain Monte Carlo (MCMC), with data-driven methods like ours. Such combinations could provide better interpretability and robustness by leveraging both theoretical understanding and the empirical richness of real-world data.

6.3 Paper III

In this study [47], we build on the SINDy framework introduced in Chapter 5, aiming to enhance its robustness and data efficiency in the context of parametric dynamical systems. One of the key limitations of the original SINDy method is its heavy reliance on the quality and diversity of available data. To address this, Paper III proposes a generalized SINDy approach (GS-SINDy) capable of integrating information across multiple trajectories. This approach becomes especially useful when dealing with parametrically varying systems in nonlinear dynamics.

We consider a family of dynamical systems governed by parametric ordinary

differential equations (ODEs) of the form,

$$\frac{d}{dt}\boldsymbol{x}(t) = \boldsymbol{f}(\boldsymbol{x}(t); \boldsymbol{\lambda}), \quad t \in \mathbb{R}_{\geq 0}, \tag{6.6}$$

where $\boldsymbol{x}(t) = [x_1(t), \ldots, x_{n_x}(t)]^T \in \mathbb{R}^{n_x}$ is the vector of states at time t and the map $\boldsymbol{f} : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ represents the (nonlinear) dynamics of the system parameterized by $\boldsymbol{\lambda}$. While the exact form of $\boldsymbol{f}$ is not known, we assume access to observational data in the form of trajectory snapshots collected under various parametric settings. Then, these observations can be represented as $\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{X}}_j\}_{1 \leq j \leq n}$, where n is the number of trajectories. Each $\hat{\boldsymbol{X}}_j = \{\hat{\boldsymbol{x}}_j(t_i); \boldsymbol{\lambda}_j\}_{1 \leq i \leq m_j}^T \in \mathbb{R}^{m_j \times n_x}$ is a sequence of measurements of the state variables with parametric settings $\boldsymbol{\lambda}_j$, and m_j is the number of sampled timesteps. The goal is to solve a sparse regression problem of the form,

$$\begin{bmatrix} \dot{\hat{\boldsymbol{X}}}_1 \\ \vdots \\ \dot{\hat{\boldsymbol{X}}}_{n-1} \\ \dot{\hat{\boldsymbol{X}}}_n \end{bmatrix} = \begin{bmatrix} \Theta(\hat{\boldsymbol{X}}_1) & \ldots & \mathbf{0} & \mathbf{0} \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \ldots & \Theta(\hat{\boldsymbol{X}}_{n-1}) & \mathbf{0} \\ \mathbf{0} & \ldots & \mathbf{0} & \Theta(\hat{\boldsymbol{X}}_n) \end{bmatrix} \begin{bmatrix} \Xi_1 \\ \vdots \\ \Xi_{n-1} \\ \Xi_n \end{bmatrix}, \tag{6.7}$$

where $\Theta(\hat{\boldsymbol{X}}_j)$ is the feature matrix constructed from trajectory j, and Ξ_j denotes the corresponding sparse coefficient matrix that encodes the active terms in the dynamics for that trajectory.

In Paper III, we enhance the SINDy framework by partitioning the feature library Θ into two subsets: $\Theta^{(s)}$, which contains basis functions shared across all trajectories with common coefficients, and $\Theta^{(d)}$, which includes basis functions with trajectory-specific coefficients. Initially, all candidate functions are assigned to $\Theta^{(d)}$, and an iterative procedure is employed to selectively transfer basis functions from $\Theta^{(d)}$ to $\Theta^{(s)}$. This decision is guided by a combination of Earth Mover's distance (EMD) and similarity metrics derived from hierarchical clustering.

We further demonstrate that this generalized formulation converges to an L_0-regularized objective function, thus preserving the sparsity-inducing characteristics of the original SINDy algorithm.

This structure substantially enhances data efficiency by enabling the joint use of multiple trajectories, even when they correspond to different parameter regimes of the same underlying system. By adjusting the similarity threshold within the clustering procedure, the method can also effectively accommodate trajectories with identical or nearly identical parameter settings. As a result, the proposed

GS-SINDy approach consistently outperforms both standard SINDy and Ensemble SINDy across a wide range of benchmark systems, including the Lotka–Volterra, Brusselator, Van der Pol, Lorenz, Hopf normal form, and FitzHugh–Nagumo models.

Recent developments in the SINDy literature have focused on managing noisy datasets by employing integral or weak formulations to improve robustness [53, 66, 72]. Given the statistical underpinnings of GS-SINDy, we believe it holds strong potential to further advance these efforts. Future work will investigate extensions of GS-SINDy that integrate weak formulations or Bayesian techniques, with the goal of improving its resilience to noise and uncertainty in real-world data.

6.4 Paper IV

Incomplete data is a persistent challenge in spatiotemporal applications such as traffic forecasting, air quality monitoring, and environmental modeling. Sensor networks are often affected by failures, communication breakdowns, or design limitations, leading to missing values that undermine downstream analysis and decision-making. Traditional imputation techniques, including Kriging, linear interpolation, and low-rank matrix completion, are computationally attractive but insufficient for high-dimensional data with nonlinear spatial and temporal dependencies. More advanced deep generative approaches, such as variational autoencoders (VAEs), generative adversarial networks (GANs), and diffusion-based models, have improved reconstruction quality. During our investigation, we observed that existing models often perform well on certain nodes but poorly on others. This imbalance suggests that different subsets of nodes exhibit heterogeneous behaviors, and using a single decoder for the entire network may lead to interference across nodes. Motivated by multi-task learning (MTL), we propose an architecture in which a shared encoder captures global spatiotemporal representations, while multiple decoders specialize in different node clusters. This design reduces negative transfer between heterogeneous nodes while still enabling beneficial knowledge sharing through the encoder.

In this study, we introduce Graph-aUgmented Imputation with Diffusion modEls (GUIDE), a diffusion-based framework that explicitly incorporates graph clustering into the imputation process. GUIDE partitions sensor nodes into structurally coherent clusters using either a correlation-based hierarchical clustering method or a VAE-based clustering method. These clusters provide the foundation for decoder specialization, allowing the model to capture both global

dynamics and localized variations.

Empirical evaluation on benchmark datasets, including AQI-36 for air quality and METR-LA and PEMS-BAY for traffic, demonstrates that GUIDE consistently achieves competitive or superior performance compared to several baseline models. Notably, GUIDE shows its strongest improvements on AQI-36, where clusters align with meaningful structures such as rural versus urban monitoring stations. Beyond improved accuracy, GUIDE reduces predictive uncertainty and exhibits greater stability across diverse missingness scenarios, making it a promising solution for large-scale, real-world sensor network applications.

6.5 Paper V

"Live fingerprints" and "spoof fingerprints" are essential concepts in biometric security, especially within fingerprint recognition systems. Live fingerprints are authentic impressions captured directly from a person's finger, while spoof fingerprints are artificially created replicas designed to deceive recognition systems. These spoofs can be made using materials such as gelatin, silicone, or printed media. High-quality datasets containing both live and spoof fingerprints are crucial for training and evaluating biometric authentication systems. However, collecting such data is often expensive, time-consuming, and fraught with privacy and regulatory challenges.

To address these limitations, Paper V presents a generative framework for synthetically producing both live and spoof fingerprints [79]. For live fingerprints, we employ DDPMs and WGAN-GP models, as introduced in Chapter 3. Spoof fingerprint generation poses greater difficulty due to the scarcity of training samples. To overcome this, we adopt a transfer learning approach using CycleGAN, which allows the model to utilize live fingerprint data during training, enhancing the spoof generator's effectiveness despite limited spoof data.

To assess the quality and realism of the generated fingerprints, we use Fréchet Inception Distance (FID) to measure similarity to real samples, and False Acceptance Rate (FAR) to evaluate uniqueness and resistance to biometric impersonation. We also include Precision and Recall for Distributions and Coverage to provide a broader assessment of both quality and diversity. Inception Score (IS) is deliberately excluded due to its sensitivity to background noise and its emphasis on diversity over fidelity.

Our findings indicate that DDPM generates the most visually realistic and high-quality fingerprints, though it tends to produce less varied samples. WGAN-GP

offers slightly lower quality but excels in generating unique outputs, making it more suitable when diversity is prioritized. Meanwhile, the fingerprint transformation model CycleWGAN-GP proves highly effective in generating spoof fingerprints under various conditions, using fewer training samples while maintaining computational efficiency and model stability. A key insight from our study is the strong correlation between the degree of spoofiness, defined as the extent to which a spoof convincingly mimics a real fingerprint, and the effectiveness of transformation techniques such as CycleGAN. Higher spoofiness levels consistently lead to improved model performance across different generative architectures.

Given the ongoing difficulty in obtaining large, diverse, and high-quality public fingerprint datasets, our contribution includes the creation of a GDPR-compliant synthetic dataset featuring both authentic and spoof fingerprints. Looking ahead, we plan to extend this work by developing generative models capable of simulating fingerprints under diverse environmental conditions, such as wet, cold, or degraded surfaces.

References

[1] H. Akaike. A new look at the statistical model identification. *IEEE Trans. Aut. Contr.*, 19:716–723, 1974.

[2] Hirotogu Akaike. *Information Theory and an Extension of the Maximum Likelihood Principle*, pages 199–213. Springer New York, New York, NY, 1973.

[3] Brian D.O. Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

[4] D Anderson and K Burnham. Model selection and multi-model inference. *Second. NY: Springer-Verlag*, 63(2020):10, 2004.

[5] Dimitrios Angelis, Filippos Sofos, and Theodoros E. Karakasidis. Artificial intelligence in physical sciences: Symbolic regression trends and perspectives. *Archives of Computational Methods in Engineering*, 30(6):3845–3865, Jul 2023.

[6] Martín Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[7] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 214–223. JMLR.org, 2017.

[8] Richard C Aster, Brian Borchers, and Clifford H Thurber. *Parameter estimation and inverse problems*. Elsevier, 2018.

[9] Ziv Bar-Joseph, David K. Gifford, and T. Jaakkola. Fast optimal leaf ordering for hierarchical clustering. *Bioinformatics*, 17(suppl 1):S22–S29, 06 2001.

[10] Julius Berner, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. *The Modern Mathematics of Deep Learning*, page 1–111. Cambridge University Press, 2022.

[11] Michael Bestehorn, Thomas M. Michelitsch, Bernard A. Collet, Alejandro P. Riascos, and Andrzej F. Nowakowski. Simple model of epidemic dynamics with memory effects. *Phys. Rev. E*, 105:024205, 2 2022.

[12] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1 edition, 2007.

[13] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[14] Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113:3932 – 3937, 2015.

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

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

[17] R.L. Burden, J.D. Faires, and A.M. Burden. *Numerical Analysis*. Cengage Learning, 2015.

[18] John C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley and Sons, Hoboken, New Jersey, third edition, 2016.

[19] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

[20] Tianrong Chen, Guan-Horng Liu, and Evangelos A Theodorou. Likelihood training of schrödinger bridge using forward-backward sdes theory. In *International Conference on Learning Representations*, 2022.

[21] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995.

[22] Raj Dandekar, Chris Rackauckas, and George Barbastathis. A machine learning-aided global diagnostic and comparative tool to assess effect of quarantine control in covid-19 spread. *Patterns*, 1(9), 2020.

[23] Valentin De Bortoli, James Thornton, Jeremy Heng, and Arnaud Doucet. Diffusion schrödinger bridge with applications to score-based generative modeling. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2021. Curran Associates Inc.

[24] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[25] Farzan Farnia and Asuman Ozdaglar. Do gans always have nash equilibria? In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[26] Urban Fasel, J. Kutz, Bingni Brunton, and Steven Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478, 04 2022.

[27] W. Feller. On the theory of stochastic processes, with particular reference to applications. *Proceedings of the First Berkeley Symposium on Mathematical Statistics and Probability (August, 1945 and January, 1946)*, pages pages 403–432, Aug 1945. Referenced by: MathSciNet [MR0027980].

[28] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org, 2017.

[29] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 2672–2680, Cambridge, MA, USA, 2014. MIT Press.

[30] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5769–5779, Red Hook, NY, USA, 2017. Curran Associates Inc.

[31] Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2015.

[32] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[33] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[34] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[35] Guillermo Iglesias, Edgar Talavera, and Alberto Díaz-Álvarez. A survey on gans for computer vision: Recent research, analysis and taxonomy. *Computer Science Review*, 48:100553, 2023.

[36] Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

[37] MARC KENNEDY. Bayesian quadrature with non-normal approximating functions. *Statistics and Computing*, 8(4):365–375, Dec 1998.

[38] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[39] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[40] Diederik P. Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and Trends in Machine Learning*, 12(4):307–392, November 2019.

[41] Thomas N. Kipf and Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR '17, 2017.

[42] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[43] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[44] Fernando Lejarza and Michael Baldea. Data-driven discovery of the governing equations of dynamical systems via moving horizon optimization. *Scientific Reports*, 12(1):11836, Jul 2022.

[45] Donglin Liu and Alexandros Sopasakis. A combined neural ode-bayesian optimization approach to resolve dynamics and estimate parameters for a modified sir model with immune memory. *Heliyon*, 10(19):e38276, 2024.

[46] Donglin Liu and Alexandros Sopasakis. A data driven approach for resolving time-dependent differential equations with noise. *IFAC-PapersOnLine*, 59(6):379–384, 2025. 14th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems DYCOPS 2025.

[47] Donglin Liu and Alexandros Sopasakis. Enhancing sparse identification of nonlinear dynamics with earth-mover distance and group similarity. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 35(3):033139, 03 2025.

[48] Mohamed Lounis and Dilip Kumar Bagal. Estimation of sir modelâ€™s parameters of covid-19 in algeria. *Bulletin of the National Research Centre*, 44(1):180, Oct 2020.

[49] J. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability - Vol. 1*, pages 281–297. University of California Press, Berkeley, CA, USA, 1967.

[50] Niall M Mangan, Travis Askham, Steven L Brunton, J Nathan Kutz, and Joshua L Proctor. Model selection for hybrid dynamical systems via sparse regression. *Proceedings of the Royal Society A*, 475(2223):20180534, 2019.

[51] Niall M Mangan, J Nathan Kutz, Steven L Brunton, and Joshua L Proctor. Model selection for dynamical systems via sparse regression and information

criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009, 2017.

[52] Rendani Mbuvha and Tshilidzi Marwala. Bayesian inference of COVID-19 spreading rates in south africa. *PLoS One*, 15(8):e0237126, August 2020.

[53] Daniel A Messenger and David M Bortz. Weak sindy: Galerkin-based data-driven model selection. *Multiscale Modeling & Simulation*, 19(3):1474–1497, 2021.

[54] Gustavo T Naozuka, Heber L Rocha, Renato S Silva, and Regina C Almeida. Sindy-sa framework: enhancing nonlinear system identification with sensitivity analysis. *Nonlinear Dynamics*, 110(3):2589–2609, 2022.

[55] Jiquan Ngiam, Zhenghao Chen, Pang Wei Koh, and Andrew Y. Ng. Learning deep energy models. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 1105–1112, Madison, WI, USA, 2011. Omnipress.

[56] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 2021.

[57] Robert K. Niven, Laurent Cordier, Ali Mohammad-Djafari, Markus Abel, and Markus Quade. Dynamical system identification, model selection, and model uncertainty quantification by bayesian inference. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 34(8):083140, 08 2024.

[58] Erik Norlander and Alexandros Sopasakis. Latent space conditioning for improved classification and anomaly detection. *ArXiv*, abs/1911.10599, 2019.

[59] N. E. Owen, P. Challenor, P. P. Menon, and S. Bennani. Comparison of surrogate-based uncertainty quantification methods for computationally expensive simulators. *SIAM/ASA Journal on Uncertainty Quantification*, 5(1):403–435, 2017.

[60] Wei Pan, Ye Yuan, Jorge Gonçalves, and Guy-Bart Stan. A sparse bayesian approach to the identification of nonlinear state-space systems. *IEEE Transactions on Automatic Control*, 61(1):182–187, 2015.

[61] Christos E. Papadopoulos and Hoi Yeung. Uncertainty estimation and monte carlo simulation method. *Flow Measurement and Instrumentation*, 12(4):291–298, 2001.

[62] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical Mathematics*. Texts in Applied Mathematics. Springer Berlin Heidelberg, 2006.

[63] Christopher Rackauckas, Yingbo Ma, Julius Martensen, Collin Warner, Kirill Zubov, Rohit Supekar, Dominic Skinner, Ali Ramadhan, and Alan Edelman. Universal differential equations for scientific machine learning, 2021.

[64] 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. *Journal of Computational Physics*, 378:686–707, 2019.

[65] Carl Edward Rasmussen and Zoubin Ghahramani. Bayesian monte carlo. In *Proceedings of the 16th International Conference on Neural Information Processing Systems*, NIPS'02, page 505–512, Cambridge, MA, USA, 2002. MIT Press.

[66] Patrick AK Reinbold, Daniel R Gurevich, and Roman O Grigoriev. Using noisy or incomplete data to discover models of spatiotemporal dynamics. *Physical Review E*, 101(1):010203, 2020.

[67] Raul Rojas. *Neural Networks - A Systematic Introduction*. Springer-Verlag, Berlin, 1996.

[68] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, 1958.

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

[70] Kevin Scaman and Aladin Virmaux. Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 3839–3848, Red Hook, NY, USA, 2018. Curran Associates Inc.

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

[72] Hayden Schaeffer and Scott G McCalla. Sparse model selection via integral terms. *Physical Review E*, 96(2):023302, 2017.

[73] Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464, March 1978.

[74] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.

[75] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 2256–2265. JMLR.org, 2015.

[76] Jialu Song, Hujin Xie, Bingbing Gao, Yongmin Zhong, Chengfan Gu, and Kup-Sze Choi. Maximum likelihood-based extended kalman filter for covid-19 prediction. *Chaos, Solitons & Fractals*, 146:110922, 2021.

[77] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché Buc, Emily B. Fox, and Roman Garnett, editors, *NeurIPS*, pages 11895–11907, 2019.

[78] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

[79] Weizhong Tang, Diego Andre Figueroa Llamosas, Donglin Liu, Kerstin Johnsson, and Alexandros Sopasakis. Fingerprint synthesis from diffusion models and generative adversarial networks. In Kohei Arai, editor, *Advances in Information and Communication*, pages 289–312, Cham, 2025. Springer Nature Switzerland.

[80] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[81] T. Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, 2012. COURSERA: Neural Networks for Machine Learning, 4, 26-31.

[82] Floris Van Breugel, J Nathan Kutz, and Bingni W Brunton. Numerical differentiation of noisy data: A unifying multi-objective optimization framework. *IEEE Access*, 8:196865–196877, 2020.

[83] V. Vapnik. Principles of risk minimization for learning theory. In J. Moody, S. Hanson, and R.P. Lippmann, editors, *Advances in Neural Information Processing Systems*, volume 4. Morgan-Kaufmann, 1991.

[84] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[85] Yuhong Yang. Can the strengths of aic and bic be shared? a conflict between model indentification and regression estimation. *Biometrika*, 92(4):937–950, 2005.

[86] Linan Zhang and Hayden Schaeffer. On the convergence of the sindy algorithm. *Multiscale Modeling & Simulation*, 17(3):948–972, 2019.

[87] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 116, New York, NY, USA, 2004. Association for Computing Machinery.

[88] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

Scientific publications

Author contributions

In Paper I, I designed and implemented the experimental framework and refined the approach. I also co-wrote and polished the manuscript.

In Paper II, I developed the core algorithm and carried out experiments. I also co-wrote and polished the manuscript.

In Paper III, I proved the main theoretical results and implemented the validation experiments. I also co-wrote and polished the manuscript.

In Paper IV, I led the development and execution of the experiments. I also co-wrote and polished the manuscript.

In Paper V, I contributed to the analysis of the problem. I participated in proofreading and revising the article.

Appendix: Conference posters

Poster 1: A data driven approach for resolving time-dependent differential equations with noise

Presented at the *14th IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems (DYCOPS 2025)* in Bratislava, Slovakia, 6 2025. For further details refer to Paper I.

A Data Driven Approach for Resolving Time-dependent Differential Equations with Noise

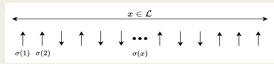
DONGLIN LIU AND ALEXANDROS SOPASAKIS
DEPARTMENT OF MATHEMATICS, LUND UNIVERSITY

Main problem

The data is generated by a coupled system of the form:

$$\begin{aligned} \frac{d}{dt}X &= \frac{1}{\tau}F(X,\bar{\sigma}) \\ \frac{d}{dt}\mathbb{E}f(X,\sigma) &= \mathbb{E}Lf(X,\sigma) \end{aligned}$$

where F is the deterministic ODE function, $\overline{\sigma} = \frac{1}{N}\sum_x \sigma(x)$ is the spatial average of σ defined on a lattice.



The dataset consists of two types of time-series data:

- **Clean data** generated from the deterministic ODE.
- **Noisy data** influenced by the stochastic process.

Our aim is to model the underlying dynamics using these datasets. We focus on two examples:

- Complex Ginzburg-Landau (CGL)

$$F(\vec{X},\sigma) = \left[\begin{pmatrix} a(\bar{\sigma}) + \gamma & -\omega \\ \omega & a(\bar{\sigma}) - \gamma \end{pmatrix} - \tilde{\gamma}|\vec{X}|^2\right]\vec{X}.$$

- Saddle-node bifurcation

$$F(X,\sigma) = a(\bar{\sigma}) + \tilde{\gamma}X^2.$$

Method

We assume that the ODE system can be represented as a set of separable SDEs:

$$\begin{aligned} dX &= N_{\theta_1}(X,\bar{\sigma})dt, \\ d\bar{\sigma} &= N_{\theta_2}(X,\bar{\sigma})dt + noise \end{aligned}$$

Using a neural network N_{θ_1} to appoximate the ODE F, where X and $\overline{\sigma}$ be input and the numerical derivative of X can be as ground truth.

Similarly, the drift of $\overline{\sigma}$ can be approximated by N_{θ_2} with ground truth be the numerical derivative with higher step size to reduce the noise. The noise can be represented as follows

$$\frac{\bar{\sigma}_{i+1} - \bar{\sigma}_i - N_{\theta_2}(X_i,\bar{\sigma}_i)(t_{i+1}-t_i)}{(t_{i+1}-t_i)^\alpha} = \Lambda(X_i,\bar{\sigma}_i).$$

Case 1. The noise is Gaussian

Then we can simply estimate the standard deviation of $\Lambda(X_i,\overline{\sigma}_i)$ and let $\alpha = 0.5$. The coupled system can be simply estimated by Euler method as follows,

$$\begin{aligned} X_{i+1} &= X_i + N_{\theta_1}(\vec{X}_i,\bar{\sigma}_i)dt, \\ \bar{\sigma}_{i+1} &= \bar{\sigma}_i + N_{\theta_2}(X_i,\bar{\sigma}_i)dt + H_\phi(X_i,\bar{\sigma}_i)\sqrt{dt}\epsilon. \end{aligned}$$

Case 2. The noise is non-Gaussian

We train $K_\varphi(X_i,\overline{\sigma}_i,\nu)$ to approximate the empirical distribution from $\Lambda(X_i,\overline{\sigma}_i)$ by inversion sampling, where ν is a standard uniform random variable.

$$\begin{aligned} \vec{X}_{i+1} &= X_i + N_{\theta_1}(\vec{X}_i,\bar{\sigma}_i)dt, \\ \bar{\sigma}_{i+1} &= \bar{\sigma}_i + N_{\theta_2}(\vec{X}_i,\bar{\sigma}_i)dt + K_\varphi(X_i,\bar{\sigma}_i,\nu)dt^\alpha. \end{aligned}$$

The Hurst exponent α can be estimated by multifractal detrended fluctuation analysis.

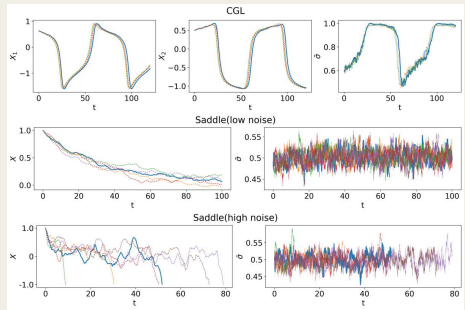


Figure 1. Synthetic data $X, \overline{\sigma}$ (blue line) versus several E-MLP simulations with empirical noise (other colors).

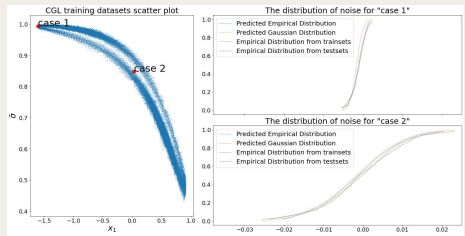


Figure 2. ethod learning the distribution of noise for CGL system. Left) Scatter plot of all training data for the CGL case. (Right) Distributions of noise at two points ("case 1" and "case 2") based on the data from the left figure.

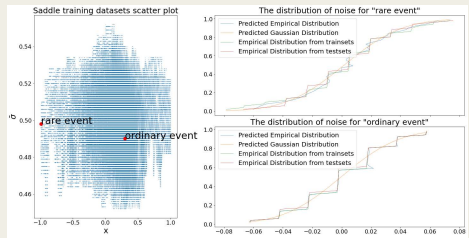


Figure 3. E-MLP method learning the distribution of noise for Saddle system. (Left) Scatter plot of all training data for the Saddle case under high noise. (Right) Comparisons of the distributions of noise at a point near a "rare event" and a point near an "ordinary event" based on the left figure.

Conclusions

The proposed method successfully learns the dynamics of both the CGL and the saddle-node systems. The version based on a non-Gaussian assumption performs better at predicting rare events in the saddle-node case, which aligns with the nature of the data generator. In both examples, the proposed approach outperforms other baseline methods—such as LSTM, vector autoregression, and Neural SDEs—in terms of accuracy and/or computational efficiency.

Printed by Media-Tryck, Lund 2025 NORDIC SWAN ECOLABEL 3041 0903

Doctoral Theses in Mathematical Sciences 2025:5
ISBN 978-91-8104-667-0
ISSN 1404-0034