Automated Model Discovery & Explanation Generation for Physicochemical Systems using Artificial Intelligence

Arijit Chakraborty

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
under the Executive Committee
of the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2025

© 2025

Arijit Chakraborty

All Rights Reserved

Abstract

Automated Model Discovery & Explanation Generation for

Physicochemical Systems using Artificial Intelligence

Arijit Chakraborty

The advent of powerful computational resources coupled with substantial progress in algorithms and increased dataset size, have led to the development of machine learning (ML) models for physicochemical systems. Such models often come at the cost of interpretability and lack of explainability by a domain expert, thereby limiting its usage. Unlike applications such as game playing, recommendation systems, or even chatbots which have permeated everyday life in recent times, science and engineering systems are steeped in first-principles knowledge that must be leveraged to render meaningful explanations of the mathematical relations that attempt to model complex physicochemical systems. Accordingly, there is a need to develop models that can be used to subsequently explain salient aspects of the physicochemical phenomena. In this work, an end-to-end data-driven model discovery engine and explanation generation artificial intelligence (AI) system is developed, which is tested on two real-world case studies of varying scales.

Chapter 1 introduces the need for combining first-principles knowledge into the modeling workflow in order to obtain more meaningful results. As shall be explained, the cost of a mistake in the sciences and engineering disciplines may prove to be fatal, and thus it is imperative that the models generated be explainable. Chapter 2 outlines a data-driven symbolic model discovery engine that outputs an ensemble of best performing models when provided data. This system

relies on the a priori inclusion of first-principles knowledge of the system being modeled, resulting in meaningful functional transformations. Chapter 3 expands on the algorithm presented in the preceding chapter, to model systems of ordinary and partial differential equations, and presents the efficacy of the approach across a wide variety of case studies.

Chapter 4 applies the modeling engine developed in the preceding chapters to a bubble column aeration problem with real-world data. The resulting models obtained are compared to the analytical ground-truth, such that the improvement over the analytical model can be captured clearly—something that would not have been possible as effectively using a black-box modeling approach. Chapter 5 applies the modeling engine to a structure-to-property prediction problem concerning the isothermal adsorption capacity of three adsorbates on zeolite structures. Varying in scale compared to the bubble column aeration system, valuable insights about the most descriptive structural properties were gained by virtue of the interpretable modeling engine.

Having successfully developed interpretable ML models for the physicochemical systems in the preceding chapters, Chapter 6 outlines the development of a large knowledge model (LKM) for automatically generating explanations from ML models. This first requires the extraction and organization of domain-knowledge from textual sources, followed by a hierarchical explanation generation strategy that yields increasingly natural language explanations. This blend of combining domain-knowledge with the natural language capabilities of modern tools such as large language models allows for explanation generation of ML models describing complex physicochemical systems. Finally, the dissertation concludes by summarizing the work undertaken, and outlines potential future directions of research.

Table of Contents

Acknowledgments xiii

Dedication. xvi

Chapter 1: Introduction and Background. 1

Chapter 2: Automated Mechanistic Model Discovery for Physicochemical Systems . 5

2.1 Motivation 6

2.2 Problem Statement and Objectives 7

2.3 Automated Model Discovery Methodology 9

2.3.1 Algorithm Overview 10

2.3.2 Knowledge Representation in Mathematical Models 11

2.3.3 Mechanistic Feature Engineering 13

2.3.4 Estimation of Model Parameters 14

2.3.5 Genetic Operations 15

2.4 Data Preprocessing 16

2.4.1 Savitzky-Golay lter for data smoothing 17

2.4.2 Cubic smoothing splines for sparse and noisy data 18

2.5 Case studies 19

2.5.1 Models linear in form and parameters 20

2.5.2 Noisy data 30

2.5.3 Models nonlinear in form and parameters 32

2.6 Conclusions 45

Chapter 3: Data-driven Discovery of Systems of Differential Equations. 48

3.1 Background 50

3.2 Methodology for Discovering System of Differential Equations 51

3.2.1 Data Description 51

3.2.2 Search Space of Potential Systems of Differential Equations 52

3.2.3 Engineering the Loss Function 53

3.2.4 Parameter Estimation 54

3.2.5 Genetic Algorithm 56

3.2.6 Ef ciency 58

3.3 Results of the Data-Driven Systems of Differential Equations Discovery Engine . . 61

3.3.1 Discovering Ordinary Differential Equations 61

3.3.2 Partial Differential Equations 72

3.4 Discussion on Shortcomings and Scope of Improvement 76

3.4.1 Susceptibility to Local Minima 76

3.4.2 Adaptive Complexity 77

3.4.3 Scalability 77

3.4.4 Sensitivity to parameter changes 78

3.5 Conclusion 78

Chapter 4: Interpretable Modeling of Bubble Column Aeration 80

4.1 Problem Statement and Objectives 80
4.1.1 Background 80
4.1.2 Bubble Column Aeration System Description 81
4.1.3 Objective 85
4.2 Methods 85
4.3 Results of Building Hybrid AI models for Bubble Column Aeration 86
4.3.1 Hierarchical Models for Salt Addition 86
4.3.2 Interpretability Analysis of Obtained Symbolic Models 87
4.4 Conclusions 88

Chapter 5: Zeolite Structure-to-Property Prediction 90
5.1 Background 91
5.1.1 Zeolites 91
5.1.2 Modeling adsorption capacity 92
5.2 Problem Statement and Objectives 93
5.3 Problem Formulation 94
5.3.1 Data 94
5.3.2 Hybrid AI Interpretable Modeling 97
5.3.3 Mechanistic Constraints 98
5.4 Results and Discussion 99
5.4.1 Effects of Data Insuf ciency on Model Discovery 105
5.4.2 Mechanistic Constraint Violations in Discovered Models 106
5.4.3 Inaccurate Curve Fits by Obtained Mechanistic Models 107

5.5 Conclusions 108

Chapter 6: Domain Knowledge Extraction and Explanation Generation 110

6.1 Motivation & Background 111

6.1.1 Motivation 111

6.1.2 What is an explanation? 112

6.2 Building a Knowledge Extraction Framework 113

6.2.1 Using Domain-speci c Textual Sources for Extracting Information 113

6.2.2 Optical Character Recognition Techniques for Scanning Textual Sources . . 114

6.2.3 Extracting Mathematical Equations and Metadata 114

6.2.4 Pitfalls and Scope for Improvement 119

6.2.5 Building a domain-speci c graph database 120

6.3 Automated Explanation Generation Methodology 122

6.3.1 Hierarchical Explanation Generation Template 123

6.3.2 Need for contextual information 124

6.3.3 Large Knowledge Model Query System 124

6.4 Generating Automated Mechanistic Explanations of Machine Learning Models . . 127

6.4.1 Bubble Column Aeration Models 127

6.4.2 Zeolite Adsorption Isotherm Models 132

6.5 Discussion 138

Chapter 7: Conclusions 140

Epilogue 143

References 144

List of Figures

2.1 Overview of the AI-DARWIN model discovery algorithm 10

2.2 (a) Conventional representations of a model observed in neural networks. (b) GPSR representation, which when simplified yields the numeral 1, as the output. Such representations often hinder readability and lead to complex or even composite functions. (c) Drawback of binary tree representation—typically used in GPSR—for complex function representation. Note the increasing width and depth of the tree which leads to a huge search space of models. It is evident why hundreds to thousands of function evaluations per iteration must be carried out in GPSR in order to attain the final desired solution. 11

2.3 Representation of a hypothesized model in AI-DARWIN. 12

2.4 Representation of linear (in parameter) models hypothesized by AI-DARWIN. . . . 14

2.5 Genetic operations used in AI-DARWIN which are steeped in principles of genetic algorithms. 15

2.6 Comparison of experimental data with analytical solution by solving the underlying equation, thus, highlighting the error due to use of finite differencing for derivative estimation. 16

2.7 Second-order kinetic rate law—decrease in MSE with generations for different complexity of model 22

2.8 Determination of reaction rate constant—decrease in MSE with generations for different complexity of model without feature engineering. 24

2.9 Determination of reaction rate constant—decrease in MSE with generations for different complexity of model with feature engineering. 25

2.10 Linear activity decay rate law—decrease in MSE with generations for different complexity of model 26

2.11 Michaelis-Menten equation—decrease in MSE with generations for different complexities of model 27

2.12 Methanol-to-hydrocarbon process—mole fraction of species (C_8H_8) plotted against time (C_1) data, is obtained using cubic spline smoothing interpolation for generating densely distributed data from previously sparse data. 43

2.13 Toluene hydrogenation—concentration of species (C_8H_8) plotted against time (C_1) data, is obtained using cubic spline smoothing interpolation for generating densely distributed data from previously sparse data. 43

2.14 Hydrogenation of methyl esters—concentration of species (C_8H_8) plotted against time (C_1) data, is obtained using cubic spline smoothing interpolation for generating densely distributed data from previously sparse data. 44

3.1 Phase-space plot (left) and noisy synthetic data (right) for Lotka–Volterra system. . 52

3.2 Plots depicting the computational complexity (left) and minimization failures (right) for varying optimization algorithms. 55

3.3 Illustration of the genetic operations—elitism, crossover, and mutation—in the proposed data-driven systems of equations discovery framework. 56

3.4 Execution time for generation of systems of equations with naive redundancy checks, for different sets of functions in the function library. 58

3.5 Illustration of the hashing mechanism used to address inef ciency as a result of the duplicates generated by the genetic algorithm. 59

3.6 Execution time for parameter estimation with and without hashing, over multiple generations. 60

3.7 Best estimates of Lotka-Volterra model at generation 5 and 10 for varying function sets. The function set details are listed in Table 3.1. 62

3.8 Minimum and average loss per generation for different functional sets in the Lotka-Volterra model. 62

3.9 Comparison of the true model with ve best estimates as obtained for the Lotka-Volterra model with logistic growth systems, for noisy data. 65

3.10 Best estimates for the SIR Model using varying function sets, obtained at generations 10 and 20. The function set details are listed in Table 3.5. 66

3.11 Minimum and average loss per generation using different function sets for the SIR model. 67

3.12 Best estimates of FHN Model at generations 5 and 20. 68

3.13 Minimum and average loss per generation for different function sets in the FitzHugh–Nagumo model. 70

3.14 Plot of the noiseless and noisy data for the Brusselator model. 70

3.15 Best Systems Estimate for Brusselator in Generation 5 71

3.16 Comparison of true and predicted values for activator and recovery phases, for the Fisher-KPP system. 73

3.17 Comparison of true and predicted values for activator and recovery phases, for the FitzHugh-Nagumo model. 75

4.1 Experimental bubble column set-up with live measurements accessible through SCADA system. 82

4.2 Comparison of analytical model predictions—with and without salt addition (here, $MgSO_4$). 84

4.3 Comparison of predicted (by AI-DARWIN's models) and measured oxygen concentrations for experiments, with and without $MgSO_4$. 88

5.1 Comparison of conventional cubic spline with a PCHIP spline for two zeolites, at temperatures of 323 K and 373 K. The PCHIP spline avoids overshoot/undershoot unlike conventional cubic splines, which helps preserve the monotonically increasing nature of the curve. 96

5.2 Plots for two zeolites (GIU & AFN) at two temperatures (323 K and 373 K) from the test set for CO_2 gas. 100

5.3 Plots for two zeolites (AFG & MSE) at two temperatures (323 K and 373 K) from the test set for N_2 gas. 101

5.4 Plots for two zeolites (ZON & AFO) at two temperatures (323 K and 373 K) from the test set for CH_4 gas. 102

5.5 Plots for various zeolites at both temperatures (323 K and 373 K) across the three gases showcasing the insuf ciency of data, while the others show insigni cant variability of absolute loading across the entire range of pressures for some cases. . . . 105

5.6 Plots for some zeolites at both temperatures (323 K and 373 K) across the three gases depicting the prediction of negative absolute loading (a mechanistic constraint violation) by AI-DARWIN's models. 106

5.7 Plots for some zeolites at both temperatures (323 K and 373 K) across the three gases depicting poor ts obtained by AI-DARWIN. 107

6.1 Data extraction procedure from textual sources. 115

6.2 Identifying equation numbers within a textual source. 116

6.3 Locating an equation within a page. 117

6.4 LaTeX code generation from an equation snippet. 117

6.5 Tesseract text extraction. 118

6.6 Merged outputs from Tesseract and LaTeX OCR engines. 118

6.7 JSON Representation 119

6.8 Knowledge graph schema used to store equations and relevant context as extracted from textual sources. 120

6.9 The steps involved in MEGAN, for processing unstructured information found in relevant textual documents into structured domain-speci c knowledge for later usage. 121

6.10 LKM query system design. 125

6.11 Signed directed graph of the ML model obtained by AI-DARWIN for bubble column aeration experiments 1-4. The overall positive and negative effects of the inputs and their combinations are apparent in the SDG. 130

6.12 Signed directed graph of the ML model obtained by AI-DARWIN for bubble column aeration experiments 5 and 6. The overall positive and negative effects of the inputs and their combinations are apparent in the SDG, as well as the different features between this model and its counterpart without salt addition. 130

6.13 Signed directed graph of the ML model obtained by AI-DARWIN for the zeolite adsorption isotherm for CH_4-adsorbate at 323 K. 136

6.14 Signed directed graph of the ML model obtained by AI-DARWIN for the zeolite adsorption isotherm for N_2 adsorbate at 373 K. 136

List of Tables

2.1 Comparison of the analytical solution with AI-DARWIN algorithm output—Noiseless Data 31

2.2 Comparison of the Analytical Solution with AI-DARWIN Algorithm Output–Noisy Data (post- ltering). MSE in predictions is reported with respect to the ltered signal. 33

2.3 Comparison of the SISO analytical solution with AI-DARWIN algorithm output – dense and noisy data (post- ltering). Fitness in predictions is reported with respect to the analytical solution. 39

2.4 Comparison of the MISO process model with AI-DARWIN algorithm output – sparse and noisy (post- ltering). Fitness in predictions is reported with respect to the numerical integral of the process model. 46

3.1 Best estimate equations and R^2 scores for different function sets at generations 5 and 10 in the Lotka–Volterra model. 63

3.2 Lotka-Volterra Equations with Logistic Growth, No Noise. Best estimate equations and $'^2$ values for different function sets tted to noise-free data. 64

3.3 Lotka-Volterra equations with logistic growth, with noise. Best estimate equations and $'^2$ values for different function sets tted to noisy data. 64

3.4 Best estimate for the Lotka-Volterra model with logistic growth using noisy data. . 65

3.5 Best estimate systems of equations, and corresponding R^2 scores for different function sets at Generations 10 and 20 in the SIR Model. 67

3.6 Best estimate systems of equations, and their corresponding R^2 scores for different function sets at generations 5 and 20 in the FitzHugh–Nagumo Model. 69

3.7 Best estimate systems of equations for the Brusselator model, using noisy data. . . 71

3.8 Best estimate system of differential equations obtained for the FitzHugh-Nagumo system, and their corresponding R^2 scores. 76

4.1 Experimental conditions used for the bubble column system. 84

4.2 The two models with standardized features predicting oxygen concentration as discovered using AI-DARWIN, along with corresponding fitness scores (here, mean absolute error and R^2 metric). 86

4.3 Parameters for the two models predicting oxygen concentration discovered using AI-DARWIN. 87

4.4 Features in the deterministic model and selected features from the models discovered using AI-DARWIN 87

5.1 Descriptors for structural and porosity properties used in this study. 95

5.2 Train and test set metrics for N_2, CH_4, and CO_2 gases at 323 K. Mean Absolute Error (MAE) and R^2 metrics have been evaluated with respect to the ground truth points, not against the spline fitted curve. 102

5.3 Train and test set metrics for N_2, CH_4, and CO_2 gases at 373 K. Mean Absolute Error (MAE) and R^2 metrics have been evaluated with respect to the ground truth points, and not against the spline fitted curve. 103

5.4 2 most important features in the model obtained for 323 K. 104

5.5 2 most important features in the model obtained for 373 K. 104

Acknowledgements

“If I have seen further it is by standing on the shoulders of Giants."

— Isaac Newton

I am lled with gratitude, rst and foremost, towards my doctoral advisor, Prof. Venkat Venkatasubramanian. He has been the most supportive, kind, and helpful mentor that one could have to guide them on this journey. I shall forever remain indebted to him for having had the opportunity to learn from him the qualities that make one a world-renowned researcher and thinker. I am particularly grateful for the freedom to explore a wide array of interesting topics that expanded my horizons and allowed me to grow as a researcher, but even more as a person. I began working with Professor Venkat during a chance meeting at the National University of Singapore, with no inkling that 6 years later I would be completing my Ph.D. under his advisement—and for that, I am grateful to all events caused by the butter y effect that led me on this path.

I am thankful to my former colleague, Dr. Abhishek Sivaram, through whose guidance and support during my rst 2 years as a Ph.D. student, I was able to learn and contribute effectively to the vast array of exciting projects being undertaken in the Complex Resilient Intelligent Systems Laboratory (CRIS Lab). I am grateful to the past and present lab members that I had the opportunity to work with and learn from, as well as the numerous talented interns that have worked with us on a multitude of projects throughout my time at Columbia.

I am grateful to our collaborators whose contributions to the work presented in some of these chapters are invaluable. I am grateful to Dr. Peter Jul-Rasmussen, Prof. Jakob Kjøbsted Huusom, and Prof. Xiaodong Liang, from Technical University of Denmark (DTU), for the collaborative effort on the bubble column aeration project. I am grateful to Dr. Akhilesh Gandhi and Prof. Faruque Hasan from Texas A&M University for our collaboration on the zeolite structure-to-property prediction problem. Besides these interesting and challenging problems that are outlined in this dissertation, I am grateful to Dr. Forwood Wiser, Prof. Faye McNeill, and Dr. Siddhartha Sen, for our collaborative work on the chemical mechanism reduction project. I am also grateful to our collaboration with Dr. Matteo Bernasconi and Prof. Thomas Friedli from the University of St. Gallen for our collaborative work. I have learned much from our efforts, and I am thankful for the time spent working on these projects together.

I am grateful to the late Professor Arvind P. Kudchadker for his support and initial nudge into the exciting world of academia. My rst foray into academic research sprang from a conversation with him during his visit to my undergraduate college in my sophomore year. I remember the conversation fondly, in which he encouraged me to seek out interesting problems to tackle and not be afraid to fail should I hit any roadblocks along the way. Following his advice, I began working with Prof. Anirban Roy, Prof. Sampatrao D. Manjare, and Prof. Iftekhar Karimi, during my undergraduate years, and I am grateful for all their support and encouragement that further propelled me to continue in research.

I came to Columbia University in the middle of the pandemic in September of 2020, having decided that I must begin my doctoral journey. Despite several restrictions on being able to meet in-person in the earlier years, I am grateful to my friends, Çem, Farid, John, Kevin, Samay, and Wallis, for their camaraderie, support, and company. It was truly comforting to know that we were all in the same boat. I also extend my heartfelt thanks to my friends beyond Columbia who have constantly encouraged and supported me in this journey: Raunaq, Sunny, Dhrubajyoti, Maitreya, Adway, Suhrid, Shantanab, Soumyadipta, Meghna, Arpita, and Deepayan.

Finally, I would like to thank my family for supporting me in this endeavor. My rst big break

into the sciences came when I was about 4 years old, and my parents presented me with a “Big Book of Knowledge", which never left my side in those formative years. I am grateful to my mother and father for that, and the boundless support and encouragement that they have showered upon me since. I am grateful to my sister for her love and kindness. I thank my in-laws for showering me with their love and support. I sincerely thank my wife, Alisa Chakraborty, for her constant love, support, and encouragement throughout the doctoral journey and beyond. She has been my lighthouse during the storms, and I am lled with gratitude to have had her by my side.

Dedication

Dedicated to my family, and my wife, Alisa, for their unwavering love and support throughout, that made this dissertation possible.

Chapter 1: Introduction and Background

Mathematical models are abstractions of the complex physicochemical phenomena observed in the world around us. Such phenomena are typically governed by fundamental laws of physics, chemistry, and biology, as well as constitutive equations and empirical laws. Models built based on these “ rst-principles” attempt to represent these complex phenomena in succinct forms that often belie their inherent complexity while providing insight into the interactions of various components of these phenomena. Such models can be useful in a myriad of disparate applications such as recognizing trends that indicate a forthcoming safety hazard; interactions of molecules that can indicate ef cacy of a new drug compound; ef cient aerodynamic design of vehicles to reduce drag thereby reducing fuel demand; discovering new relations for previously unknown complex systems; explaining the reasoning for a particular decision that can have adverse economic impacts; among others. Chemical engineering is such a eld that has a multitude of fundamental laws and equations in its arsenal that are used to model complex systems within its purview, encompassing a wide range of length scales—from as small as molecules within a drug compound and its interactions within the human body, to large re nery complexes where several tons of chemicals are produced daily for further downstream processing into useful products. As a domain that began with the industrial production of soda ash, the increasing breadth of the domain's applications today necessitate greater depth of its constituent knowledge base.

Initially, chemical engineering relied primarily on empirical methods and heuristics, without much emphasis on quantitative, rst-principles-based models for almost a century [1]. That all changed in the 1950s, marked by what is known as the Amundson era [2], when applied mathematical methods, particularly linear algebra, ordinary differential equations (ODEs), and partial differential equations (PDEs), were introduced. These techniques enabled the creation of models for unit operations grounded in rst-principles [3, 4]. This was further augmented over the sub-

sequent decades by the advent of mathematical optimization techniques like mixed-integer linear programming (MILP) and mixed-integer non-linear programming (MINLP) [5]. This transformation was pioneered by Roger Sargent [6], and his students.

This purely mathematical modeling approach in chemical engineering was followed by the next signi cant modeling paradigm—the introduction of knowledge representation concepts and search techniques from arti cial intelligence (AI). Such an approach to developing AI systems gave rise to automated proving of mathematical theorems [7]; the earliest language model, ELIZA [8]; an AI expert that diagnosed bacterial infections, MYCIN [9], among numerous others. These systems relied on a combination of large amounts of specialized domain knowledge, often in the form of heuristics, explicit rules, symbolic logic, and other knowledge representations. Such a paradigm laid the foundations for several expert systems to be developed within chemical engineering. This started in the early 1980s under the leadership of Westerberg, Stephanopoulos, and others from that era [10, 11, 12, 13].

The expert systems era was followed by what became known as the AI Winter, when funding for AI research greatly diminished both in computer science, and in its application domains. During this period, research in data-driven techniques continued resulting in what would later become known as “numeric AI” or machine learning (ML). In ML, models learned from data without the need for human involvement. The rapid increase in the data and computational resources over the past three decades paved the way for the growth of AI's purely data-driven sub- eld of machine learning (ML). Its successes in conventional applications such as digits classi cation [14], game playing [15], protein structure prediction [16], and recently in its latest avatar as conversational chatbots [17] or large language models (LLMs), are a testament to their ef cacy in many tasks that were previously reserved only for humans. There are numerous bene ts to such systems such as increased memory and processing capacity, reduced repetitive tasks for humans, enhanced scalability, and reduction in errors due to human cognitive overload.

Machine learning's conventional applications, which include movie recommendations [18], restaurant recommendations, face recognition, autonomous driving, dialogue generation, among

others, however, do not rely on fundamental first-principles knowledge as that of the physicochemical systems omnipresent in scientific and engineering domains. They seem to be good at certain applications, such as writing drafts, coding assistance, summarizing, translating, and answering questions; but fail when asked to provide optimal flowsheet configurations for processes, structure-to-property predictions for novel materials, among numerous such technical problems. Further, ML systems generally excel at pattern matching without as much nuanced understanding of the problems that they attempt to solve as that of a human expert. Despite their seemingly advanced capabilities, such systems face several limitations. They may generate misleading or biased information which could lead to erroneous reasoning. As the consequences of an inaccurate decision in physicochemical systems can be severe, potentially fatal, there are concerns of verifiability of the predictions, generalizability of the model on unseen data, inexplicability and lack of reasoning by the system. Furthermore, these conventional applications have colossal amounts of data that have been curated from a variety of sources to train the ML models. These “big data” domains are unlike the sciences and engineering, which are termed as “precious data” domains, owing to the lack of massive amounts of data used for training many popular ML models of today. This is due to the costs of obtaining new data being capital and labor intensive. Thus, conventional ML algorithms may not be as effective for such domains since they tend to be purely data-driven.

Accordingly, a new paradigm of modeling physicochemical systems has emerged which combines the first-principles knowledge of physicochemical systems into the AI workflow, to create “hybrid AI” systems [19, 20]. Hybrid AI systems are generally more robust, accurate, and lend themselves to explanation generation. Such systems can take advantage of symbolic domain-specific information and combine them with the data-driven capabilities of ML models. Such a paradigm is prescriptive, and encourages the inclusion of domain knowledge to create more robust models that adhere to fundamental laws of the physicochemical system being studied. A valuable contribution of hybrid AI systems are their ability to yield interpretable and explainable models, which allows one to provide some reasoning for the model's output.

In accordance with the hybrid AI paradigm of modeling physicochemical systems, an end-

to-end data-to-model and model-to-explanation pipeline has been built such that one can provide data, and the AI system would yield an interpretable ML model followed by generating an explanation of the same in a natural language manner. Such a hybrid AI system required the integration of symbolic AI principles such as knowledge representation along with machine learning approaches. The resulting models are interpretable by design, as opposed to their often used highly-accurate black-box modeling counterparts, neural networks. The obtained models are then used to generate mechanistic explanations by providing relevant context of the physicochemical system being modeled. The resulting natural language explanation generation tool is an example of large knowledge model (LKM) [21], an improvement on LLMs which are augmented with domain knowledge. The subsequent chapters outline the components of this automated modeling and explanation generation pipeline along with real-world examples which highlight its ef cacy.

Chapter 2: Automated Mechanistic Model Discovery for Physicochemical Systems

Mathematical models are a means of representing complex physicochemical phenomena using various functional transformations that operate on the variables of the system. Conventionally, obtaining such models required rst-principles knowledge of the system through fundamental laws, constitutive equations, empirical relations, among others. Such techniques yielded equation(s) with multiple variables, parameters, and functional transformations which one could use to study and manipulate the real system. A key shortcoming of such an approach manifested itself in the modeling of complex systems, where accurate and complete rst-principles knowledge of the associated physicochemical phenomena was not known. Such an approach is termed “white-box" modeling since the reasoning for obtaining the models was clearly presented, and the approach is transparent overall. Such models can be used to generate cause-and-effect relationships, such as in the Bernoulli's equation which describes the interrelationship between pressure, velocity, and elevation in uid dynamics. Further, they are useful to generate explanations to end-users since these are rooted in scienti c principles.

An attempt to circumvent the need for domain-speci c knowledge that was previously critical in rst-principles-based white-box modeling, is the use of purely data-driven techniques such as machine learning (ML). Due to the rise in data for a variety of applications within chemical engineering (and beyond), ML modeling provided a feasible alternative. Here, based on the data available, an algorithm could yield functional relations linking the inputs to the outputs of a physicochemical system, without the inputs of domain knowledge into the work ow. Models obtained based on this paradigm are typically termed “black-box" models, since their internal workings of input-output mappings are not easily interpretable or explainable. This lack of transparency in

the model development is an acceptable trade-off for numerous applications when the cost of an erroneous prediction or decision is trivial. However, as brie y alluded to in the previous chapter, chemical engineering systems remain complex dynamical systems where a mistake can potentially be fatal. Thus, the need for interpretability and explainability is paramount in such applications.

In this chapter, an approach for data-driven mechanistic model discovery, which combines the bene ts accrued from incorporating rst-principles knowledge of a physicochemical system along with the data-driven algorithmic prowess of ML modeling, is presented. These models are termed “gray-box" models because they are a combination of the white-box and black-box modeling philosophy, resulting in more robust, interpretable, and generalizable models. Such models are a sub-class of the broader group of “hybrid AI" models, which attempt to incorporate domain-speci c knowledge into the AI/ML work ow.

2.1 Motivation

Mechanistic models are desirable because they are built on fundamental principles and laws that govern complex physicochemical systems and processes. They are more reliable and explainable as they explicitly account for the underlying mechanisms. In black-box models these are sacri ced in favor of obtaining a good t between the input and output variables quickly and easily [22, 23]. Now, such an approach might be acceptable, perhaps even appropriate, for certain applications such as game-playing, computer vision, and natural language processing, where there are no fundamental laws or constitutive equations to guide us. Furthermore, these domains generate tremendous amounts of data that can be mined blindly. They are truly "big data" applications.

However, chemical engineering systems and processes, indeed most engineering systems, are governed by fundamental principles and laws, which should be taken into consideration. Moreover, it is not easy to generate such copious amounts of data in many chemical engineering applications—data here is limited and precious. While there is increased access to more data now than, say, a decade ago, it is not feasible to generate terabytes of data easily, except perhaps in computer simulations. So, purely data-driven techniques borrowed from such domains are not

appropriate for "precious data" applications and hence might not work as well. On the other hand, valuable rst-principles knowledge can be leveraged and exploited to reduce the need for large amounts of data.

The lack of mechanistic or causal interpretation and explanation in conventional ML regression approaches is a serious drawback, in general, and worrisome in safety-critical applications, in particular. Machine learning results often provide no insight about how they were obtained and hence their “black-box" character [1, 24, 25, 26]. The limitations of popular model-free parametric regression strategies such as neural networks and genetic programming symbolic regression (GPSR) are two-fold—(i) they provide complicated and uninterpretable input-output relationships, and (ii) parameter sensitivity is not captured. These hinder the usability of such models, speci cally for the identi cation of underlying mechanisms. They often develop intractable functional transformations [27], further limiting explainability and aiding over tting. Neural networks, for example, might use relatively simple activation functions, but their multi-layered architectures often lead to complicated hidden representations that can be hard to interpret and explain [28, 29, 30]. GPSR often results in complicated functional forms that do not have rst-principles mechanisms to support them [31].

One way to address the interpretability issue is to develop hybrid models [24, 32, 33, 34, 35, 36, 37, 38, 39]. Hybrid models incorporate rst-principles concepts with data-driven techniques [34, 40]. Typically, rst-principles such as conservation equations are augmented with data-driven models like neural networks [40, 41, 42, 43] and fuzzy logic [44, 45] for parameter estimation. This preserves a rst-principles understanding of the phenomena, while letting much of the interpretability go into the feature-parameter relationship identi cation. This, however, comes at the cost of gaining insights into the causal relationships in the data.

2.2 Problem Statement and Objectives

In this chapter, an attempt is made to address the following challenge:

> Can a machine learning system be developed that automatically identi es the

rst-principles-based models that capture the underlying physical, chemical, and/or biological mechanisms generating the data?

In some sense, this is what science is all about—understanding the world around us in terms of a few fundamental principles and mechanisms expressed mathematically. This challenge is addressed in a manner similar to how humans go about reasoning and discovering mechanisms and models, guided by intelligent guesses, heuristics, and inferences. For example, in reaction kinetics, the reaction rates are driven by the concentration of various species in the system. It is hence advisable to obtain relationships between the species concentration and rates, despite not knowing the actual functional form. In many chemical engineering applications, phenomena driven by the underlying physics and chemistry rst-principles yield equations that are composed of elementary functional forms. Even for more complicated cases, the proposed approach could yield simpler, reduced-order, models that are easier to interpret, and gain insights from, without sacri cing accuracy too much.

The proposed approach employs a genetic algorithm (GA) to identify the simpler, elementary, functional forms of a model, guided by domain knowledge, that can be directly coupled to rst-principles-based physicochemical mechanisms. For instance, in reaction engineering, domain knowledge informs one that rate variables, concentration variables, and certain mathematical functional forms are important. The proposed framework exploits this understanding by incorporating such knowledge in the creation of the library of elementary functions. This is augmented by statistical regression using the identi ed functions [46, 47]. In conventional genetic function approximation methods, a tree representation of the structure gives rise to unnecessarily complex functions, which should be avoided. The use of statistical methods, such as ordinary least squares (OLS), partial least squares (PLS) or least absolute shrinkage and selection operator (LASSO) regression, provide the coef cients needed to determine the nal functional form, along with an estimation of errors. Thus, one can understand which inputs are most strongly predictive of an outcome based on the coef cients of the functions in the obtained function form. This approach has the bene t of restricting the complexity of the model obtained. This is often a problem with

conventional GPSR algorithms, where the resultant composite functions must be simplified to a great extent to uncover the actual function [48, 49, 50, 51, 52]. Further, confidence levels in parameter estimates are not determined either. For example, though Eureqa® [53], an evolutionary algorithm-based model identification engine, predicts functional forms from data (and often yields complex functional forms [51, 54]), it is unable to provide statistical significance of the estimated parameters.

2.3 Automated Model Discovery Methodology

The approach proposed is termed AI-DARWIN [55, 56], as a nod to the Darwinian concept of evolution and subsequent use of genetic algorithm (GA) in the workflow. Here, the simplest model form (as justified by Occam's Razor) that fits the data adequately is obtained using the genetic algorithm (GA) that searches through a space of allowed elementary functions and their combinations. These elementary functions in the GA function library were selected based on domain knowledge of chemical engineering, knowing what kinds of functional forms and their combinations often appear in the models. These, in turn, are based on first-principles understanding of how such elementary functions arise from the corresponding physicochemical mechanisms.

For example, one often finds elementary functions such as G^2, 4^G, $\ln G$, $\sin G$, and $\cosh G$ in chemical engineering models, but rarely does one come across forms such as $4^{\cosh G^2}$, $\sinh^{-1}\sinh^{-1}\sinh^{-1}G$, $\ln^{-1}\sin^{-2}G$, $\ln^{-1}G$, $4^{G4^{G4^G}}$, etc. But such complicated expressions are often identified in prior work on symbolic regression (SR), such as in Eureqa [53], and by Babu and Karthik [49], using genetic algorithms. Such needlessly complex models are unhelpful in understanding the underlying physics and chemistry of the system. There is no physicochemical mechanism that would have generated such functional forms from first-principles. There is no gain in mechanistic insights from a model of such incredible complexity. Such complex functional forms are hardly different from black-box neural network models, which are also known to fit the data well but with no interpretability. The AI-DARWIN approach avoids such unnecessarily complex models, due to the strong belief that it is important to have simpler models, wherever possible, from which one can

identify causal relationships between variables and parameters. They are better than the complex functional transformations that often arise from over tting. Since the relevant functional forms are identi ed by exploiting mechanistic insights, the GA is constrained to develop only simple and meaningful combinations of these functions and their parameters for the nal model. This important aspect is missing in prior work on GA-based symbolic regression. Therefore, the kind of models they identify often are too complex and ill-suited for many applications.

2.3.1 Algorithm Overview

Figure 2.1: Overview of the AI-DARWIN model discovery algorithm

The algorithm starts with a randomly generated initial population of individuals, where each individual is an elementary functional form that attempts to model the given data. Each individual is a xed-length vector of features (i.e., functional transformations of the input variables), where each element of the vector represents a functional transformation of a speci ed set of functions, collectively termed the function pool or library, consisting of polynomials, trigonometric functions, hyperbolic functions, and exponentials. While this library is adequate for many applications in chemical engineering, it can be suitably adapted based on the understanding of the problem. The

GA examines different combinations of features from the function pool [57, 58, 59]. If some key functional form was missing in the library, the system would return a lower-order approximation of the true model. It must be noted that this function pool is not exhaustive—functions can be added or removed as per the need or knowledge of the speci c application. Figure 2.1 illustrates the overview of the AI-DARWIN algorithm.

2.3.2 Knowledge Representation in Mathematical Models

Figure 2.2: (a) Conventional representations of a model observed in neural networks. (b) GPSR representation, which when simpli ed yields the numeral 1, as the output. Such representations often hinder readability and lead to complex or even composite functions. (c) Drawback of binary tree representation—typically used in GPSR—for complex function representation. Note the increasing width and depth of the tree which leads to a huge search space of models. It is evident why hundreds to thousands of function evaluations per iteration must be carried out in GPSR in order to attain the nal desired solution.

In the domain of mechanism identi cation, if model interpretability is not the desired output,

one may choose to use neural networks, support vector machines (SVMs), and other “black-box" models. These are known for their prediction capabilities but lack in explaining their prediction(s) as well as purely symbolic models. However, the focus of the proposed system is in obtaining interpretable models which explain the system. Conventionally, the models obtained by existing model identi cation engines are represented as LISP expressions, or tree representations. While this is acceptable and useful for simple functions, the complexity associated with the model search space increases signi cantly when dealing with larger models. This further breaks apart when modeling large nonlinear multi-parameter models, or even for that matter, sloppy models [60]. Such a representation also accounts for composite functions, among other obscure transformations which are not observed in mechanistic models which describe real-life processes. As a result, the models in AI-DARWIN are represented by use of a combination of three arrays. Perils of conventional representations in neural networks and GPSR are depicted in Figure 2.2.

Figure 2.3: Representation of a hypothesized model in AI-DARWIN.

AI-DARWIN's representation is depicted in Figure 2.3. Three arrays are used for denoting the functional transformations $f()$, variables to be transformed $d()$ and operators acting upon two functional transformations $\$()$. The hyperparameters in such a representation include the complexity of the features involved in the model formulation step (p), as well as the number of terms to include in the model form (l). The hyperparameters allow the user to obtain the simplest model which describes the data while avoiding complicated functions as would be obtained from the aforementioned conventional modeling methods. Increased complexity of functions using multi-

plication and division of features from the function library to create more complex features for regression, is permitted. For instance, $\sin^{1}G_{8}^{o}$ $\cos^{1}G_{9}^{o}$ and/or $\frac{\sin^{1}G_{8}^{o}}{\cos^{1}G_{9}^{o}}$ can be a feature for the algorithm. The eventual inclusion of these in the best functional form obtained by the algorithm is contingent upon the accuracy desired. This adds complexity, which is analogous to adding more levels of depth in the conventional binary tree representation of symbolic regression.

2.3.3 Mechanistic Feature Engineering

The elementary functions used in AI-DARWIN are directly related to underlying physicochemical mechanisms, such as reactions (1 order, 2 3 order, etc) and diffusion, as well as to conservation laws, system geometries, and boundary conditions. AI-DARWIN exploits such insights and understanding in designing the GA library of elementary functions. Furthermore, by design, the GA ensures that the next best solution obtained in an iteration is at least equivalent or better than the best solution from the preceding iterations. There is no need for exorbitantly large evaluations to arrive at a reasonable approximation of the underlying parameters and mechanistic equations. All these lead to simpler and interpretable models that also t the data satisfactorily.

Feature extraction by AI-DARWIN combines elements from the function pool to create an individual (each individual being a mathematical model.) These are regressed against the outputs, to yield measures of t and uncertainty of parameter estimates, which corresponds to statistical testing. As an example, for reaction engineering, a common knowledge is that there are certain orders of reactions which are dominant. These characterize the function pool — representing different orders of the reaction. Each individual is a xed-length vector of features — $\frac{G_1}{G_2}$ $\frac{G_1^2}{G_2}$ $\frac{G_1}{G_2^2}$ $\frac{G_1^2}{G_2^2}$ — where G_1 and G_2 are different variables of the process. The GA identi es these xed-length features (feature extraction), and associated parameters (statistical testing) to identify the underlying mechanism of the system.

Model representation is a key issue because GAs manipulate the encoded representation of the functional form [61], which can eventually affect performance [62]. In AI-DARWIN, the proposed representation is solely based on using arrays. An inherent advantage of this representation is that

Figure 2.4: Representation of linear (in parameter) models hypothesized by AI-DARWIN.

the complexity of the functional form obtained can be restricted a priori. This helps in identifying the smallest set of features which need to be used to explain the input-output relationships. Figure 2.4 illustrates the representation used in the AI-DARWIN implementation.

2.3.4 Estimation of Model Parameters

Once the population is generated, the parameters of each individual in the population (i.e., each model) are estimated using ML regression methods, and the accuracy of t is estimated using a metric (typically termed tness score)—here, mean squared error (MSE) is used. The tness score is assigned to every individual and is used to rank the individuals based on the accuracy of modeling the data. This is analogous to quantifying the survivability of an individual in the next population, hence invoking the Darwinian philosophy of the survival of the ttest. The purpose of having the population ranked on the basis of the tness score, is to ensure that the best individuals of the next population will be at least equivalent or better than their counterparts in the previous population, but not worse.

It is useful to make a distinction between feature selection (i.e., elements of the function pool), and feature extraction and combination. Feature selection is done by the user based on domain

knowledge, such as deciding which inputs and functions in the function library to provide to AI-DARWIN. Feature extraction and combination is performed by the GA by extracting the appropriate elementary functions from the function library, and their combinations, guided by the statistical testing for the survival of the ttest model. Even though it might seem that the feature selection might involve a great deal of domain expertise, as will be described through the different case studies in the following sections, for many chemical engineering applications, the function library created is quite adequate. This can be further augmented by adding more functions should the need arise as per the domain-speci c problem.

2.3.5 Genetic Operations

Figure 2.5: Genetic operations used in AI-DARWIN which are steeped in principles of genetic algorithms.

The three genetic operations performed in AI-DARWIN are:

1. Selection: an individual is copied without alteration into the next population.

2. Mutation: one or more elements of an individual are altered to generate a new individual.

3. Crossover: a new individual (offspring) is created by combining randomly chosen segments of two individuals (parents).

The three genetic operations performed on every population are illustrated in Figure 2.5. This creates a new set of individuals which becomes the new population, thereby completing one iteration. Successive iterations are termed generations and the algorithm runs for a pre-speci ed number of generations ($=64$). This is usually taken to be a high value so that a high accuracy t is obtained. Alternatively, a convergence limit can be set to limit the likelihood of over tting.

2.4 Data Preprocessing

Figure 2.6: Comparison of experimental data with analytical solution by solving the underlying equation, thus, highlighting the error due to use of nite differencing for derivative estimation.

The often repeated phrase “garbage in, garbage out" holds true for the identi cation of models which are expected to shed some light on the physicochemical processes taking place. One must be wary of the pitfalls of using a model identi cation engine with improper data, as it could yield an output which does not resemble the true model—either in form or parameters. A robust model

identi cation algorithm must be able to appropriately account for noise in the data. Consider a model to be of the form $H^{1}x^{o} = 5^{1}x^{o} \, ¸ \, n^{1}x^{o}$, where the true model is given by $5^{1}{}^{o}$, and the error/deviation is given by $n^{1}{}^{o}$. This deviation from the truth can have disastrous effects when using conventional techniques for model discovery, for instance when using numerical differentiation for obtaining derivatives. The expected nonlinear trend could seem to follow a linear relation or vice-versa, or even yield a haphazard trend with no relevance to the actual process. Such examples with experimental data are shown in Figure 2.6.

It is thus imperative to remove noise from the data, or at least recover as much truth from the noisy signal as possible. There is ample discussion on a variety of methods discussed in literature [63]. These methods can be broadly classi ed as global and local methods. In the context of AI-DARWIN, two distinct denoising approaches—Savitzky-Golay lter (local method) and cubic weighted spline smoothing interpolation (global method)—were used with each having their own merits and scope of application.

2.4.1 Savitzky-Golay lter for data smoothing

The Savitzky-Golay lter [64] is one of the most prominent methods in ltering of noise from data. It relies on a convolution operation that is successively performed on a speci ed collection of adjacent data points, over the entire data set. The coef cients of the convolution function for a small range of data points are given in literature and have been expanded on by others to encompass a large number of cases. This procedure is equivalent to tting a lower-order polynomial by performing least-squares minimization, a procedure which minimizes the sum of the squared errors between the predicted and observed values. This objective function is given by Eq. 2.1.

$$<8= "(= \sum_{8=1}^{\tilde{\Theta}^{8=}} \left(-H_{8}^{1}t^{o} \sum_{9=0}^{\tilde{\Theta}^{?}} 0_{9}t^{9} \right)^{a^{2}} \qquad (2.1)$$

Here, the coef cients of the low-order polynomial are given by 0_{9}, $=_{?}$ is the order of the

polynomial considered; and $F_{8=}$ is the number of data points considered in a single convolution operation (termed window length). It must also be noted that the Savitzky-Golay lter relies on the choice of the order of polynomial that will be used for localized smoothing, and the window-length. If the window-length is high, then a larger number of points will be used to t the chosen polynomial.

The convolution function's coef cients are also provided such that estimation of higher order derivatives from noisy data is possible, and this technique has yielded acceptable results in an earlier version of AI-DARWIN [56] that focused on models which are linear in parameters but not necessarily in their features. This lter works best when the density of data available is high. By virtue of being a localized ltering technique, it does not account for information beyond the domain for which it is being used, and is thus, limited by window-length. Accordingly, the use of this lter is restricted when dealing with sparse data. This is typical of industrial processes where there is a limitation on the sampling of data and measurements that can be made. The construction of models using less data might yield a function that ts well on the training set (data points used to obtain parameters), although it might not explain the underlying mechanism due to poor generalizability by virtue of poor performance on the test set (unseen data points used to test performance of the trained). In such a case, an interpretable mechanism identi cation engine does not have any conclusive advantage over its counterpart black-box models such as neural networks or surrogate models.

2.4.2 Cubic smoothing splines for sparse and noisy data

In sparse data regimes, cubic smoothing splines are the preferred denoising mechanism. It is important to note that these are different from interpolating splines. Interpolating splines t every data point, which is detrimental when applied to noisy data, while smoothing splines t the data while simultaneously smoothing the trend by virtue of a smoothness parameter (λ). This can be tuned to the user's preference of the degree of smoothness desired [63]. The objective function for smoothing splines is given by Eq. 2.2. The rst term is the conventional least-squares minimization

metric, and the global smoothness of a function (quanti ed by its curvature) is the second term. The global smoothness parameter can be tuned through the hyperparameter (cross-validation to generate an optimal trajectory which ts the data.

$$\min \; AA>A = \tilde{O} \; {}^{1}H_{8}{}^{1}C^{\circ} \quad H_{8}{}^{1}C^{\circ o2} \; , \; _ \quad {}^{1} \quad H^{00}{}^{1}C^{\circ 2}3C \qquad (2.2)$$

$$8=1$$

The bene ts of this method are as follows:

1. It is a global smoothing method which yields a smooth function.
2. The desired degree of smoothness can be controlled by a user de ned smoothness factor.
3. Sparse data can be extended to as many points needed within the interval to ensure that nite difference and other derivative estimation methods work accurately.

In addition to the above-mentioned bene ts of spline smoothing, weights can be assigned to the noisy data set prior to the smoothing exercise such that certain data points are given higher preference in the process of smoothing and interpolation than others. This is of considerable importance in contrast to previously discussed local method of denoising because the user can in uence the output of the spline smoothing. A priori knowledge of the expected trend can be incorporated at this stage of the mechanism identi cation process by use of assigning weights to the data points. Moreover, spline smoothing interpolation allows one to adjust for gaps in the data. This is particularly useful when dealing with sparse data.

2.5 Case studies

In this section, the physicochemical mechanisms which were used to validate AI-DARWIN are presented. These include mechanisms from the fundamental realms of chemical engineering such as reaction engineering and transport phenomena. The objective was that given data, the algorithm must be able to obtain the best functional form—-accurate, yet not too complex—that

maps the inputs variable to the outputs via the discovered feature(s) and thus, assist in identifying the underlying mechanism.

At the outset, it must be clari ed what is meant by `analytical solution'. Throughout this discussion, analytical form/solution is used to describe the `ideal' function which must be obtained through AI-DARWIN. Speci cally, it is the truth on the basis of which the data has been simulated/obtained, and that AI-DARWIN seeks to discover through its algorithm. In the various case studies discussed in this section, there may be some where a closed-form solution to the underlying differential equation(s) does not exist. Here, `analytical solution' would refer to the differential equation(s), since that is what AI-DARWIN must discover. In other case studies, where a closed-form solution (typically, of differential equations) is present, `analytical solution' refers to this model since that is what AI-DARWIN attempts to estimate here.

2.5.1 Models linear in form and parameters

For the test cases described in this section, a population of 10 individuals was randomly generated. Convergence to a solution with acceptable error is contingent upon the initial random population. If this population consists of obscure features with no bearing to the true mechanistic model, then naturally the algorithm would take a larger number of generations to converge. However, if this same population is similar to the underlying equation, then convergence is faster. This is analogous to having an initial guess in the vicinity of the global optimum (resulting in faster convergence) as opposed to having the guess farther away (convergence will be slow), for the case of gradient-based methods. 30% of the population was selected into the next generation, 40% of the population was mutated upon while the remaining 30% was crossed-over. The resulting individuals after performing genetic operations became the next population. 1,000 simulated data-points are used to estimate the function approximation on the basis of which inferences can be made of the underlying mechanism of the corresponding phenomena. The algorithm is run for 1,000 generations in all case studies.

In many practical applications, the measurement data is often noisy or erroneous, posing a con-

siderable challenge for mechanism and model identi cation. Noisy outlier data, for example, can lead to incorrect functional forms that are quite different from the true model. For instance, exponentials can be approximated by higher order polynomials. As a result, this incorrect identi cation could lead to an incorrect mechanism. Tables 2.1 and 2.2 present the results of the case studies with noiseless and noisy data, respectively. Noise is added using a random number generated with the mean of the data and a standard deviation of 10% of that value. This noise is added during the trajectory of the trend, and so the system operates at different regimes with different noise characteristics. This was done to provide a realistic scenario to test AI-DARWIN's ef cacy in dealing with noisy data.

Preprocessing of this data is necessary to lter the noise. Conventional mean-centering efforts would not have been valid due to the trend in the signal and variable noise characteristics across the trends. There are a variety of lters available in the literature for the purpose of smoothing data, two of which have been described previously in section 2.4. Here, the Savitzky-Golay (SG) lter [64] was used as it tends to retain the shape of the signal [65] in addition to preserving inherent features of the signal whilst being simple to implement. Other ltering techniques could be applied in lieu of the method used in this study.

Single-Input Single-Output Systems

Single-input single-output (SISO) system models are considered of the type:

$$H = V_1 5_1 ¹G^{º}, V_2 5_2 ¹G^{º}, \quad , V_< 5_< ¹G^{º} \tag{2.3}$$

Here, $<$ is the number of features of the model. It is important to note that $5_8 ¹G^{º}$ does a functional manipulation of the singular input G alone. The variable and its features do not have any additional parameters, for instance e^{4UG} or $\sin VG$ (which would make the optimization problem nonlinear in parameters).

Second-order kinetic rate law

Consider a liquid-phase reaction A + B C, which is rst-order in reactant B. Given excess B, the order of the reaction with respect to reactant A must be obtained. The analytical solution of this case study is obtained by solving the corresponding rate law, yielding a second-order rate law which is of the form = 5¹Ⓒ. The number of features are varied in unit increments from 2 to 6, to highlight the differences in convergence with varying complexity of function forms.

The solution obtained by the algorithm for two terms is the same as the analytical solution. The corresponding MSE is zero, indicating a perfect t. For the individuals with more features for this model, if the terms with lower weights are neglected, AI-DARWIN successfully discovers the equation form which explains that the input data pertains to second-order kinetics.

Figure 2.7: Second-order kinetic rate law—decrease in MSE with generations for different complexity of model

While the correct equation form was obtained and thus, one can make inferences about the underlying mechanism, it is interesting to note that by increasing the terms in the functional form, more complex yet less accurate models were obtained. Furthermore, the decrease of MSE with generations is not necessarily consistent with every run. This can be attributed to the initial population being generated randomly resulting in varying initialization. This is illustrated in Figure 2.7. At the end of the pre-speci ed 1,000 generations, AI-DARWIN arrives at the best functional form which is the true underlying rst-principles model of the system.

Determination of reaction rate constant

Consider a reaction: $A + B \xrightarrow{catalyst} C$, which is rst-order in A while B is a catalyst. Given initial concentration of A ($_{\$}$), concentration data of product C () with respect to time $C(t)$, the reaction rate constant $k(t)$ must be determined by obtaining a function that approximates the analytical solution.

This analytical solution consists of an exponential term with an additional parameter to be determined. The algorithm focuses on functions that match the form of Equation 2.3; as a result, it is not possible to obtain the exact functional form (consisting of the exponential term) in this case. Interestingly, a reasonable approximation is estimated in the form of Taylor series expansion of the solution. The corresponding decrease of MSE with successive generations for different number of features is depicted in Figure 2.8. This apparent limitation can be overcome by feature engineering—the output variable is rearranged to assist the model discovery in obtaining the correct functional form. The improvement over the case without this feature engineering step is evident in the decrease in MSE with generations, illustrated in Figure 2.9.

Linear Activity Decay Rate Law

A pernicious effect of a reaction taking place on a catalyst is the loss of catalyst activity ($a(t)$). There are several mechanisms proposed to explain the same. One such empirical mechanism is the linear decay rate law which is a function of time. Catalyst decay following this law is seen in the

Figure 2.8: Determination of reaction rate constant—decrease in MSE with generations for different complexity of model without feature engineering.

case of the para-hydrogen conversion on tungsten catalyst, when poisoned with oxygen.

The objective was to deduce the functional form of catalyst activity and corresponding speci c decay constant(λ), given data of the catalyst activity, and time (t). The algorithm outputs the equation form accurately—irrespective of the number of features, which were all obtained to be rst-degree monomials. The decrease in MSE with generations for different complexity is shown in Figure 2.10. The MSE after 1,000 generations is zero which exempli es perfect estimation of the correct linear model by the algorithm.

Figure 2.9: Determination of reaction rate constant—decrease in MSE with generations for different complexity of model with feature engineering.

Fractional Kinetics

Consider a reaction—A B + C—the kinetics of which are dictated by a fractional rate law. Often when reactions follow fractional-order kinetics, it hints at a more complex underlying mechanism. Concentration data of A () was used as the input, and the rst-order derivative of the same with respect to time ($\frac{3}{3C}$) as the output. Since simple functions such as integer-order monomials in the function library will not be able to estimate this mechanism, multiple monomials with order ranging from 0 to 3 in increments of 0.1 have been added. This allows for an increased functional search space for the algorithm. This example highlights that with moderate user assistance, one can help improve mechanism identi cation using the current implementation. AI-DARWIN

Figure 2.10: Linear activity decay rate law—decrease in MSE with generations for different complexity of model

was accurately able to estimate this differential equation with a MSE of zero and thus, fractional-order kinetics was correctly identi ed.

Trambouze Reactions—Parallel Reactions

Consider Trambouze Reactions—liquid-phase parallel reactions in which species X, Y and Z are formed from a single reactant (species A). The concentration data of A with respect to time is known and thus, it is regressed with its rst derivative, or the rate of consumption of A (A) in order to obtain more information about the underlying kinetics of the system. The system obtained the exact equation with MSE of zero. This is another instance where the algorithm is able to accurately estimate the differential equation that de nes the physics of the system.

Michaelis-Menten Equation

The most prominent kinetics equation for enzymatic reactions is the Michaelis-Menten equation. The rate of formation of substrate ($\frac{3}{3C}$) is proportional to the concentration of substrate ( ). The Michaelis-Menten equation is given by:

$$A = \frac{3}{3C} = \frac{+_{<0G}}{" , } \tag{2.4}$$

where $+_{<0G}$ is the maximum rate of reaction for a given total enzyme concentration. Since

Equation 2.4 is nonlinear, the algorithm is unable to estimate the exact functional form accurately. Furthermore, this function includes a parameter—Michaelis constant (K)—which cannot be estimated with the problem formulation of a generalized linear model since the true model form is nonlinear in parameters. Instead, AI-DARWIN generates an approximation of the same using other functions in the function pool (e.g. fractional orders, polynomials etc). It must be noted that for increasing complexity of the functional form there is an associated reduction in error. This is illustrated for two runs in Figure 2.11. It is evident that when the allowed model has 6 features, the error is signi cantly less than that for the simplest functional form (with only 2 features). This can be attributed to the increased combinations of functions which can approximate the nonlinearity more accurately through the more complex functional forms.

Figure 2.11: Michaelis-Menten equation—decrease in MSE with generations for different complexities of model

Multiple-Input Single-Output Systems

Multiple-input single-output (MISO) system models are of the type:

$$H = V_1 5_1^{1} G_{1:3}^{o}, \; V_2 5_2^{1} G_{1:3}^{o}, \; \ldots, \; V_< 5_<^{1} G_{1:3}^{o} \tag{2.5}$$

Here 3 is the number of input variables considered. Models of this form lead to additional

complexity in the algorithm, due to estimation of the features which could be combinations of multiple input variables and their functional transformations. If only a single input variable is used, this simpli es to the SISO system, described in Section 2.5.1.

Fractional Kinetics—Multiple Reactants

Consider a reaction system which follows fractional kinetics for multiple reactants. The equation is of the form:

$$A = \frac{3}{3C} = 5^{1} \quad - \quad - \quad ^{o} \tag{2.6}$$

The function library for mechanism discovery was augmented by adding fractional order monomials. This step depends on the user's knowledge of the problem, and thus leads to faster convergence. Otherwise, on the basis of the tness metric, the best functional form would be identi ed over a lot more generations without ever obtaining the correct model due to lack of exibility in the model identi cation stage.

Reversible Reactions

Consider a reversible reaction: ~Ä ™ 2 B. The rate law for this equation is of the form:

$$A = \frac{3}{3C} = 5^{1}C - \quad - \quad ^{o} \tag{2.7}$$

where A is the rate of reaction, is the forward reaction rate constant, is the backward reaction rate constant, and & are the concentrations of species and respectively. Given concentration and time data with no a priori knowledge of the type of reaction occurring, AI-DARWIN was used to discover its mechanistic model, and thereby classify the reaction as reversible or irreversible. Furthermore, if the reaction is reversible, the goal was to obtain the equilibrium constant. The derivative of the concentration with respect to time was obtained by using the nite difference

method. This preprocessing step is crucial to identify the mechanism. It is based on the domain knowledge of reaction rate laws. This derivative is the output variable while the concentrations of the species were the inputs. The algorithm was successfully able to identify the correct reaction rate law and concentration equilibrium constant with an overall MSE of zero.

Series Reactions in Batch Reactor

Consider an elementary liquid-phase series reaction carried out in a batch reactor: A $\xrightarrow{k_1}$ B $\xrightarrow{k_2}$ C. Given the concentrations of the species A, B, and C with respect to time, the rate laws must be discovered. The algorithm is assisted by using the time derivatives of concentration of the species A and B as the output. This crucial preprocessing step is based on the domain knowledge of rate laws. The three rate laws (consumption of A, consumption of B, and formation of C) were accurately obtained in this manner. The corresponding speci c rate constants were also estimated. This exempli es the capability of the system to estimate the functional relationship from differential equations for multiple-input systems. With a priori knowledge of the reactions taking place, the algorithm was able to infer that the data corresponded to a series reaction system.

Fick's 2nd Law of Diffusion

Fick's 2nd law of diffusion describes the rate of accumulation of concentration within a volume ($\frac{\partial C}{\partial t}$) being proportional to the local curvature of the concentration gradient ($\frac{\partial^2 C}{\partial x^2}$). (The constant of proportionality is diffusivity ( ). Concentration is of the form: = 5^{-1} μ μ Given concentration ( ), position (x) and time (t) data, the relationship between these variables must be determined. The rst and second derivatives of concentration with respect to time and position, respectively, were used to engineer new features for the model identi cation task. AI-DARWIN successfully estimated the diffusivity. Thus, higher-order differential equations were also tackled by the algorithm. The result showed that the data pertained to Fick's 2nd law. Since the form of this equation is the same as that of the heat equation, it could be asserted that the algorithm would likewise identify the correct mechanism given the corresponding data.

Nernst Equation for a Concentration Cell

Another valuable case study to discuss the capability of AI-DARWIN was the estimation of the Nernst equation for a concentration cell—in which both electrodes are made of the same metal and the solutions in the cells are of the same ion as the electrodes. Consider the reaction given as: $Ag(s) + Ag^{+}(conc) \rightleftharpoons Ag(s) + Ag^{+}(dil)$. In such a cell, the standard cell potential is zero because the half reactions are identical in both directions. The corresponding cell potential is linearly related to the logarithm of the concentrations of the ions. Allowing the appropriate functions in the function library resulted in the algorithm correctly identifying the true mechanistic equation.

Monod Equation

Among the many rate laws that explain the kinetics of cell growth, the most commonly used is the Monod equation for exponential growth. Given the concentrations of the cells (X) and the substrate (S) with respect to time, the AI-DARWIN algorithm was used to obtain a function approximation. The Monod equation's functional form is similar to that of the Michaelis-Menten equation. Both equations being nonlinear (in form and parameters), AI-DARWIN was unable to ascertain the exact functional form when strictly linear models were allowed. Despite this, the function obtained had a reasonably low MSE, which was accurate enough for making predictions.

Table 2.1 represents the analytical functions compared with the best discovered models obtained by AI-DARWIN, for models which are linear in form and parameters. It is evident that the algorithm was able to determine the corresponding functional form for the linear SISO and MISO models, which helped identify the underlying mechanism of the physicochemical phenomena. For nonlinear systems, the algorithm was able to make reasonable approximations which could be useful for predictive tasks in most engineering applications.

2.5.2 Noisy data

In many practical applications, the measurement data is often noisy or erroneous, posing a considerable challenge for mechanism and model identi cation. Noisy outlier data, for example, can

Table 2.1: Comparison of the analytical solution with AI-DARWIN algorithm output—Noiseless Data

Case study	Input(s)	Output	Analytical Solution	AI-DARWIN Output	MSE
Second-order Kinetics	C	1•	$\frac{1}{}$ = 20 ¸ 0•12 C	Same as the analytical solution	0
Determination of Reaction Rate Constant	C	ln¹ —— > —— > o	ln¹ —— > —— > o = 0•311 C	Same as the analytical solution	0
Linear Activity Decay Rate Law	C	0	0 = 1 10 ^{4}C	Same as the analytical solution	0
Fractional Kinetics		A >A$\frac{3}{3C}$	A = 0•2 1•5	Same as the analytical solution	0
Trambouze Reactions – Parallel Reactions		A >A$\frac{3}{3C}$	A = 10 ^{4} ¸ 1•5 10 ^{3} ¸ 8 10 ^{3} 2	Same as the analytical solution	0
Reversible Reactions	–	A >A$\frac{3}{3C}$	A = 0•5 5 2	Same as the analytical solution	0
Series Reactions in Batch Reactor	– –	A >A$\frac{3}{3C}$	A = 0•5	Same as the analytical solution	0
2.5.1	– –	A >A$\frac{3}{3C}$	A = 0•5 ¸ 0•2	Same as the analytical solution	0
2.5.1	– –	A >A$\frac{3}{3C}$	A = 0•2	Same as the analytical solution	0
Fick's 2nd Law of Diffusion	– G–!C $\frac{m^2}{mC}$	$\frac{m}{3C}$	$\frac{m}{3C}$ = 5 10 ^{5} $\frac{m^2}{mC}$	Same as the analytical solution	0
Nernst equation for a Concentration Cell	–		= 0•0591 $\log_{10}$ ¹ — o	Same as the analytical solution	0
Fractional Kinetics	– –	A >A$\frac{3}{3C}$ 31	A = 0•5 0•8 2•5 2	Same as the analytical solution	0

lead to incorrect functional forms that are quite different from the true model. For instance, exponentials can be approximated by higher order polynomials. As a result, this incorrect identi cation could lead to the inference of an incorrect mechanism.

Accordingly, results of the case studies with noisy data are presented for this study are presented in Table 2.2. Gaussian noise was added using a random number generated with the mean of the value of the data and a standard deviation of 10% of the value of the data. This noise was added during the trajectory of the trend, and so the system operates at different regimes with different noise characteristics resulting in heteroskedastic noise. Preprocessing of this data was therefore necessary to lter the noise. Conventional mean-centering efforts would not have been valid due to the trend in the signal and variable noise characteristics across the trends. There are a variety of lters available in the literature for the purpose of smoothing data, two of which have been described previously in Section 2.4. Here, the Savitzky-Golay (SG) lter [64] was used as it tends to retain the shape of the signal [65] in addition to preserving inherent features of the signal whilst being simple to implement. Other ltering techniques could be applied in lieu of the method used in this study.

2.5.3 Models nonlinear in form and parameters

At the outset, it is necessary to discuss the bene ts of using more complex model forms—beyond the previously discussed models that are linear in form and parameters—which have the added dif culty of nonlinear parameter estimation. Dynamical processes are typically described by differential equations which involve derivatives of one or more independent variables. Further, these are not necessarily restricted to the rst order and can involve higher order derivatives. Previously [56], physicochemical models were identi ed using strictly linear models where the dependent variable is either a numerically evaluated derivative of an independent variable, typically with respect to time or spatial coordinates (e.g., $H = \frac{3}{3C}$ would be evaluated as $H = \frac{\frac{8_{i+1}}{G_{i+1}} - \frac{8_i}{G_i}}{\ldots}$), or a ltered derivative (when using noisy data), and these variables were subsequently used to estimate the parameters of the constructed models. However, numerical estimation of derivatives is

Table 2.2: Comparison of the Analytical Solution with AI-DARWIN Algorithm Output–Noisy Data (post- ltering). MSE in predictions is reported with respect to the ltered signal.

Case study	Input(s)	Output	Analytical Solution	AI-DARWIN Output	MSE
Second-order Kinetics	C	1•	$\frac{1}{}$ = 20 ¸ 0•12 C	$\frac{1}{}$ = 18•93 ¸ 0•12 C	6•8 10 ^{2}
Determination of Reaction Rate Constant	C		= 1 4 $^{0•311C}$	= 0•27 C 0•029C ^{2}	1•092 10 ^{5}
Linear Activity Decay Rate Law	C	0	0 = 1 10 ^{4}C	0 = 0•999 9•845 10 ^{5}C	9•438 10 ^{11}
Fractional Kinetics		A >A $\frac{3}{3C}$	A = 0•2 $^{1•5}$	A = 0•222 $^{1•5}$	1•487 10 ^{3}
Trambouze Reactions—Parallel Reactions		A >A $\frac{3}{3C}$	A = 10 ^{4} ¸ 1•5 10 ^{3} ¸ 8 10 ^{3} ^{2} ¸	A = 3•205 10 ^{5} ¸ 1•728 10 ^{3} ¸ 7•427 10 ^{3} ^{2} ¸	2•26 10 ^{11}
Reversible Reactions	–	A >A $\frac{3}{3C}$	A = 0•5 5 ^{2}	A = 0•499 4•99 ^{2}	1•738 10 ^{11}
Series Reactions in Batch Reactor	– –	A >A $\frac{3}{3C}$	A = 0•5	A = 0•50 6•432 10 ^{4} ^{2}	1•809 10 ^{6}
2.5.1	– –	A >A $\frac{3}{3C}$	A = 0•5 ¸ 0•2	A = 0•219 0•98	8•974 10 ^{7}
2.5.1	– –	A >A $\frac{3}{3C}$	A = 0•2	A = 0•202 3•943 10 ^{3}	1•619 10 ^{6}
Fick's 2nd Law of Diffusion	– G–!C $\frac{m^2}{mC}$	$\frac{m}{mC}$	$\frac{m}{mC}$= 5 10 ^{5} $\frac{m^2}{mC}$	$\frac{m}{mC}$= 4•958 10 ^{5} $\frac{m^2}{mC}$	3•99 10 ^{9}
Nernst equation for a Concentration Cell	–		= 0•0591 $\log_{10}$ 1 — o	= 0•057 $\log_{10}$ ¸ 0•058 $\log_{10}$	1•703 10 ^{5}
Michaelis-Menten equation	(	A(>A $\frac{3}{3C}$ (	$\frac{1•33}{0•0266}$ ¸ (	A(= 9•278 $^{1•1}$ (¸ 7•153 $^{0•6}$ (	3•34 10 ^{4}
Monod equation	(–	A6 >A $\frac{3}{3C}$	A6 = $\frac{0•46}{33•5}$ ¸ (	A6 = 2 10 ^{4} $^{1•6}$ $^{0•9}$ (¸ 0•12 $^{0•9}$ 9	1•539

limited by the density of data points, errors in measured variable(s) (which ampli es the error in estimated derivatives), and its delity to the actual model. Further, data in real processes are sampled at a nite rate. Additionally, measurement sensors can induce noise in the measured values, miscalibration can cause errors, and human error can seep in during data collection. Eventually, this nite and noisy data is used to construct models which can be erroneous and do not accurately portray the underlying behavior of the system. Obtaining the differential equations that de ne the underlying complex physicochemical mechanism of a system is often not an attractive option when predictions must be made about the system's dynamics. This is primarily due to the estimation exercise of derivatives, which can steer one away from the actual true mechanistic model if estimated inaccurately.

Chemical engineering is rife with nonlinear models—a speci c example that is often encountered is the identi cation of large kinetic models, which typically have multiple parameters. Researchers have been known to erroneously linearize such models, which can adversely affect the accuracy and reliability of the results as noted by Belohlav et al. [66]. Linearizing such nonlinear models also leads to distortion in the experimental errors [67]. As a result, identifying the dynamical nonlinear equations directly are more sought after for practical purposes. As will be presented in this section, these models are solutions of the differential equations that describe the model and do not rely on estimation of derivatives from the data.

The nal step of obtaining mechanistic model(s) is to estimate the parameters. This is a considerably more dif cult problem when it comes to nonlinear parameter estimation—it is open-ended and has a multitude of solution techniques to choose from. Some of the more common approaches used are Nelder-Mead [68], Levenberg-Marquardt [69, 70], Powell's conjugate direction set methods [71], adaptive random search (ARS) [66], trust region algorithms—to name a few. Other techniques include hybrid approaches where global optimum search is performed by genetic algorithms [72], while a local optimizer is used to determine the optimal solution [73].

An interesting point to note is that most of the work in the mechanism detection and model discovery arena have not discussed the statistical reliability of the parameters of the estimated param-

eters in the model obtained. Con dence intervals are a good measure of the statistical signi cance of the parameters. It must be noted, however, that the focus of this work of model discovery is not towards estimating parameter uncertainties alone. Parameter uncertainty estimation is subject to the parameter estimation algorithm and several underlying assumptions about the plausible error models. In this work, the uncertainties are estimated based on the Hessian of the loss function (relying on a Gaussian assumption about the point of maximum likelihood estimate). The search algorithm and parameter uncertainty quanti cation can be easily adapted in AI-DARWIN. For AI-DARWIN, a large range of bounds of the parameters to be estimated has been considered. Based on a priori knowledge, if available, a better range can be used for estimation of the parameters. This will help in increasing the speed of the parameter estimation problem.

In AI-DARWIN for nonlinear model identi cation, the Trust Region algorithm is used. As a variant of the Levenberg-Marquardt algorithm, it is generally more robust for constrained optimization problems. In the neighborhood of a parameter guess, the next iterate point is obtained through an approximation of the objective function, within bounds. For a detailed discussion on the mathematical aspects of the same, the reader is directed to this article by Yuan [74]. The choice of the parameter estimation algorithm here is meant to be representative and can be substituted with other approaches as the user deems t. This work intends to be prescriptive, rather than didactic, proposing a general framework for linear/nonlinear mechanism identi cation using a genetic algorithm. For developing a non-mathematical perspective of nonlinear regression, the reader is advised to review the work by Motulsky and Ransnas [67].

The remainder of this section describes case studies used to test the ef cacy of AI-DARWIN on nonlinear model identi cation. The presented case studies are rst tested for the case of densely distributed data, followed by tests with sparse data. Both circumstances were tested on noiseless as well as noisy data, to highlight the practical applicability of AI-DARWIN.

Reaction rate constant estimation

A rst-order reaction's rate law is a rst-order linear differential equation relating the rate of change of concentration of a species to its concentration. The dif culty of identifying the reaction rate law accurately from this differential equation form is contingent upon obtaining data at a very high sampling rate coupled with estimating the rate of change of concentration with respect to time. This problem is simpli ed with the use of the nonlinear model identi cation engine, which does not rely on estimating the derivative of any input. It identi ed the exact analytical solution using this approach for the noiseless case while obtaining a reasonable estimate when using noisy data.

Series reactions in batch reactor

For the following reaction carried out in a batch reactor: $A \xrightarrow{k_1} B \xrightarrow{k_2} C$, the data to construct the system of differential equations describing reaction rate laws can be adversely affected due to sensor disturbances, malfunction, miscalibration—to name a few. These issues result in inaccurate estimation of the underlying model equation. Using AI-DARWIN, the function form along with its parameters are estimated accurately. Further, for practical cases where noisy data can lead to inaccurate estimates, data preprocessing through human guidance can yield accurate results.

Catalyst deactivation laws

Catalyst deactivation can be modeled using rate laws which are similar to reaction rate laws. They can be of rst-order, second-order, fractional-order, or other more complex functions. Here, the exponential rate law ($r(C) = 4\,e^{-\nu C}$) and reciprocal power rate law ($r(C) = C^{-\nu}$), are considered. The practical limitations of relying on the differential equations as outlined in the previous examples, hold true for these differential equations as well. $\frac{\partial r}{\partial C} = -\nu\, r$ (exponential rate law) and $\frac{\partial r}{\partial C} = -\nu\, r^{1.5}$ (reciprocal rate law). The catalyst decay constant and the corresponding function form were estimated accurately by the proposed model identi cation engine, AI-DARWIN.

Stirred-tank blending process

Consider the modeling of dynamical systems through the example of a stirred-tank blending process with two input streams. It is modeled analytically as a linear rst-order differential equation using mass and component balances. If a priori knowledge about the process is available, one can choose to include exponentials and monomials of varying orders in the function pool. Additionally, data preprocessing, which includes a denoising exercise augmented by human understanding of the trend (removal of outliers, choice of polynomial and window-size for Savitzky-Golay ltering etc) when given noisy data helps in the model identi cation exercise. Thus, one is able to steer AI-DARWIN towards interpretable solutions that explain the process. Given noiseless and noisy data, AI-DARWIN was able to identify the function with high accuracy ($R^2 = 1$ for noiseless data and $R^2 = 0{\cdot}99$ for noisy data).

Electrically heated stirred tank

Now, consider the dynamical process of an electrically heated stirred tank. This is described by the two rst-order differential equations:

$$\frac{3)}{3C} = 10 \quad 0{\cdot}2) \; ¸ \; 0{\cdot}1)_4 \tag{2.8}$$

$$\frac{3)_4}{3C} = 250 \quad)_4 ¸ \;) \tag{2.9}$$

For the analytical solution of the temperature $)_1$ (°C), the equation was identi ed with high accuracy ($R^2 = 1$ for noiseless data and $R^2 = 0{\cdot}975$ for noisy data).

Step responses of a second-order process

For dynamical systems described by second-order differential equations, when it is perturbed from its steady-state by a step input of magnitude, the response can take the following forms:

$$H(C) = \left(1 - \frac{\tau_1 e^{-C/\tau_1} - \tau_2 e^{-C/\tau_2}}{\tau_1 - \tau_2} \right), \quad \text{overdamped}, \zeta > 1 \tag{2.10a}$$

$$H(C) = \left(1 - \left[1 - \frac{C}{\tau} \right] e^{-C/\tau} \right), \quad \text{critically damped}, \zeta = 1 \tag{2.10b}$$

$$H(C) = \left(1 - e^{-\zeta C/\tau} \left[\cos\left(\frac{\sqrt{1-\zeta^2}\, C}{\tau} \right) + \frac{\zeta}{\sqrt{1-\zeta^2}} \sin\left(\frac{\sqrt{1-\zeta^2}\, C}{\tau} \right) \right] \right), \quad \text{underdamped}, 0 < \zeta < 1 \tag{2.10c}$$

, where ζ is the damping coefficient, τ – τ_1 & τ_2 are time constants.

The results for noiseless data were not tabulated since AI-DARWIN accurately discovered all the models, and noisy data is tabulated in Table 2.3 respectively. Confidence intervals for the parameter estimates are mentioned only for the case studies performed with noisy data (Table 2.3), as the exact analytical solution was identified when using noiseless data.

Nonlinear Multiple-Input Single-Output Systems

$$H = \left(\xi_1 \left(x_1, V_1 \right), \xi_2 \left(x_2, V_2 \right), \ldots, \xi_n \left(x_n, V_n \right) \right) \tag{2.11}$$

Multiple-input single-output Systems (MISO) system models are considered of the type given by Eq. 2.11. The features here are nonlinear in form as well as parameters. They can be a combination of functions from the function pool, as decided by the user at the start of the model identification exercise. Accordingly, MISO case studies pertaining to relatively complicated reaction kinetics with complex reaction pathways are presented to showcase the efficacy of AI-DARWIN's mechanistic model discovery capabilities. The examples discussed in this section use less than 20 data points for each input variable for the mechanism identification exercise. This is representative of real processes which are limited by finite sampling rate, inaccuracies, miscalibration in measurement sensors, human-error, disturbances etc.

Table 2.3: Comparison of the ODE analytical solution with AI-DARWIN algorithm output dense and noisy data (post- ltering). Fitness in predictions is reported with respect to the analytical solution.

Case study	Input	Output	Analytical Solution	AI-DARWIN Output	Fitness (MSE & ' ^{2})
Determination of Reaction Rate Constant	C		$^1C^{\circ} = 1 \quad 4^{\ 0\cdot311C}$	$^1C^{\circ} = 0\cdot994 \underset{\pm 0\cdot642}{} \quad 0\cdot989\ 4^{\ 0\cdot333C \pm 0\cdot451}$	$4\cdot90\ 10^{\ 6}$ ' ^{2} = $0\cdot999^{\circ}$
Series Reactions in Batch Reactor	C		$^1C^{\circ} = 2\ 4^{\ 0\cdot5C}$	$^1C^{\circ} = 1\cdot968\ 4^{\ 0\cdot496C \pm 0\cdot12}$	$3\cdot99\ 10^{\ 5}$ ' ^{2} = $0\cdot999^{\circ}$
2.5.3	C		$^1C^{\circ} = 3\cdot\bar{3}\ 4^{\ 0\cdot2C}$ $3\cdot\bar{3}\ 4^{\ 0\cdot5C}$	$^1C^{\circ} = 3\cdot33\ 4^{\ 0\cdot2\ C \pm 0\cdot139}$ $3\cdot6464^{\ 0\cdot502C \pm 0\cdot493}$	$5\cdot16\ 10^{\ 6}$ ' ^{2} = $0\cdot999^{\circ}$
2.5.3	C		$^1C^{\circ} = 2 \ , \ 1\cdot\bar{3}\ 4^{\ 0\cdot5C}$ $3\cdot\bar{3}\ 4^{\ 0\cdot2C}$	$^1C^{\circ} = 2\cdot05 \ , \ 0\cdot987\ 4^{\ 0\cdot553C \pm 2\cdot821}$ $3\cdot042\ 4^{\ 0\cdot182C \pm 0\cdot434}$	$4\cdot12\ 10^{\ 5}$ ' ^{2} = $0\cdot999^{\circ}$
Exponential Decay Rate Law	C	0	$0\ ^1C^{\circ} = 4^{\ 0\cdot001C}$	$0\ ^1C^{\circ} = 0\cdot996\ 4^{\ 0\cdot00097C \pm 0\cdot002}$	$5\cdot01\ 10^{\ 6}$ ' ^{2} = $0\cdot997^{\circ}$
Reciprocal Power Decay Rate Law	C	0	$0\ ^1C^{\circ} = 6 \quad C^{\ 0\cdot002}$	$0\ ^1C^{\circ} = 6 \quad C^{\ 0\cdot002 \pm 0\cdot011}$	$2\cdot39\ 10^{\ 7}$ ' ^{2} = $0\cdot998^{\circ}$
Stirred-tank Blending Process	C	G	$G\ ^1C^{\circ} = 0\cdot517 \quad 0\cdot017\ 4^{\ \frac{C}{3}}$	$G\ ^1C^{\circ} = 0\cdot517 \quad 0\cdot0168\ 4^{\ 0\cdot330\ C \pm 16\cdot104}$	$8\cdot63\ 10^{\ 10}$ ' ^{2} = $0\cdot999^{\circ}$
Electrically Heated Stirred Tank	C	)	$)\ ^1C^{\circ} = 370$ $21\cdot78 \quad 4^{\ G\ 11\cdot099}$ $1\cdot7684^{\ G\ 0\cdot901}$,	$)\ ^1C^{\circ} = 368\cdot727 \quad 20\cdot644\ 4^{\ 8\cdot831\ C \pm 0\cdot109}$ $1\cdot8754^{\ 0\cdot0221C \pm 0}$	$5\cdot123\ 10^{\ 1}$ ' ^{2} = $0\cdot9753^{\circ}$
Step response of a second-order process: overdamped	C	H	$H\ ^1C^{\circ} = 10 \quad 204^{\ 0\cdot01C}$, $104^{\ 0\cdot02C}$	$H\ ^1C^{\circ} = 18\cdot137 \quad 25\ 4^{\ 0\cdot004\ C \pm 0\cdot006}$, $14\cdot1264^{\ 0\cdot015\ C \pm 0\cdot009}$	$3\cdot466\ 10^{\ 2}$ ' ^{2} = $0\cdot983^{\circ}$
Critically damped	C	H	$H\ ^1C^{\circ} = 10 \quad 104^{\ 0\cdot02C}$, $0\cdot2C\ 4^{\ 0\cdot02C}$	$H\ ^1C^{\circ} = 8\cdot099 \quad 8\ 4^{\ 0\cdot01\ C \pm 0\cdot097}$, $0\cdot288\ C\ 4^{\ 0\cdot015\ C \pm 0\cdot01}$	$6\cdot224\ 10^{\ 2}$ ' ^{2} = $0\cdot995^{\circ}$
Underdamped	C	H	$H\ ^1C^{\circ} = 10$ $104^{\ 0\cdot01C}\cos 0\cdot017C$, $5\cdot774^{\ 0\cdot01C}\sin 0\cdot017C$	$H\ ^1C^{\circ} = 9\cdot07 \pm 0\cdot224$ $9\cdot038\ 4^{\ 0\cdot009\ C \pm 0\cdot0007}\cos 0\cdot028\ C$, $2 \quad 4^{\ 0\cdot026\ C \pm 0\cdot0006}\sin 0\cdot057\ C$	$6\cdot088\ 10^{\ 2}$ ' ^{2} = $0\cdot998^{\circ}$

As a consequence of the above-mentioned errors, use of nite difference for derivative estimation is inhibited. This is because nite difference method approximates the derivative of a function better as the limit of the independent variable approaches zero. However, for sparse data this is not satis ed and with noisy data, derivatives obtained could be vastly different from the analytical derivative. The pitfalls of using such data for derivative estimation were previously depicted in Figure 2.6. In such cases, one would like to avoid estimating the differential equation directly and minimize the error between the output variable (say, concentration of species β_8) and predicted variable (say, concentration of species as predicted by AI-DARWIN: $\hat{\beta}_8$).

Thus, a novel paradigm of nonlinear model identi cation by AI-DARWIN was proposed, when dealing with sparse data. By virtue of this approach, the user can obtain model forms with high accuracy with respect to the analytical/real model. This is illustrated through the case studies of methanol-to-hydrocarbon production, toluene hydrogenation, and methyl esters hydrogenation (in Table 2.4 for noisy data). Results for noiseless data were not tabulated since AI-DARWIN accurately discovered all the equations.

Methanol-to-hydrocarbon production

The methanol-to-hydrocarbon process is a nonlinear parameter estimation problem referenced from Floudas et al. [75]. It appears in Maria [76] and is a reduced-order kinetic model. This reaction system is given as:

$$
\begin{cases}
A \xrightarrow{k_1} B \\
A + B \xrightarrow{k_2} C \\
C + B \xrightarrow{k_3} P \\
A \xrightarrow{k_4} C \\
A \xrightarrow{k_5} P \\
A + B \xrightarrow{k_6} P
\end{cases}
\tag{2.12}
$$

where A represents oxygenates, B is H_2, C represents ole ns and P denotes paraf ns, aromatics, and other products. 16 time-stamped experimental data points are used for the mole fractions of components A, C, and P.

First, the experimental data is denoised such that it is as close as possible to the analytical solution. Further, since this method relies on the estimation of the derivative, a large number of data points with high delity to the actual trend are required. However, with no prior knowledge of the analytical solution, local smoothing methods such as nite differencing and Savitzky-Golay [64] must be relied upon. It was found that use of Savitzky-Golay lter failed to produce a smooth function with sparse data. Instead, a weighted cubic smoothing spline interpolation was used.

This data preprocessing step is crucial for obtaining the correct parameters. Human intervention in this step involves choice of the smoothing factor for the smoothing splines, as well as the weights to be assigned to each of the points. Additionally, removal of outliers in the sparse experimental data helps the spline smoothing exercise obtain a better t. It must be emphasized that spline interpolation and smoothing spline interpolation are different from each other. A smoothing spline aims to balance tting of data with smoothing the data, while interpolation estimates data points within an interval with no regard for the smoothness of the resulting function. This not only acts as a denoising routine, but also helps increase the number of data points within the given interval.

Weighting of the smoothing spline is used to account for points which one would like to bring the overall curve closer towards or further away from. This is particularly useful when the magnitude of the independent variables is low and there is less noise associated with the data—in such a case, the smoothing spline can be made to follow this trend more, rather than the data points associated with higher noise. Once the denoising exercise is complete, 10,000 data points were obtained within the interval from approximately 13 noisy experimental points.

Based on this larger dataset, AI-DARWIN obtains an estimate of the derivative of the dependent variable (here, derivative of the mole fraction of species $\frac{3I_8}{3C}$) with respect to time. Using this derivative estimate, the mole fraction of the species at the next time-step is predicted using numerical integration (here, Euler's method). Since the density of the data points used is high and the difference between successive data points is very low, it approximates well. The parameters of the derivative are then estimated such that the square of the sum of differences between the predicted mole fraction ( ) and the denoised mole fraction ( ) is minimized.

As opposed to traditional approach of minimizing the error between the derivative estimated from data ($\frac{3I_8}{3C}$) and the value of derivative obtained from data ($\frac{3\hat{I}_8}{3C}$), the proposed approach does not rely on prior estimation of the derivative whatsoever. Instead, the form and parameters of this derivative are explicitly obtained by AI-DARWIN, and are used to predict the output ( ), which is subsequently compared with the known measured values. Results of this are presented in Figure 2.12. It must be noted that the coef cient of determination ( ) in the plots are with respect to the analytical solution.

Toluene hydrogenation

Consider a toluene hydrogenation reaction scheme given by: A B ! C, involving toluene (A), 1-methylcyclohexene (B), and methylcyclohexane (C). This model [66] is described by three ordinary differential equations representing the kinetic reaction rate laws. 14 experimental data points corresponding to the time evolution of concentrations for each of the species are known.

Finite differencing and Savitzky-Golay ltering techniques do not yield acceptable results as

Figure 2.12: Methanol-to-hydrocarbon process—mole fraction of species (C_3H_6) plotted against time (C_1) data, is obtained using cubic spline smoothing interpolation for generating densely distributed data from previously sparse data.

shown in Figure 2.6. Hence, cubic weighted smoothing spline interpolation technique was used to convert the data from a sparse to a dense data regime. Weights were assigned to the data points for the construction of the spline such that the spline passes through the first data point as it is assumed that the initial conditions are known accurately, without any error in measurement. Assigning weights is not a necessary step but it assists in obtaining a trend with sufficient data points and as close to expected trend as possible. These data preprocessing steps are subjective and will depend on the user's expectation of what the trend could look like based on the sparse data available and priori knowledge of the process.

Figure 2.13: Toluene hydrogenation—concentration of species (C_6H_6) plotted against time (C_1) data, is obtained using cubic spline smoothing interpolation for generating densely distributed data from previously sparse data.

From Figure 2.13, it is clear that this technique helps get the predicted curves reasonably close to the analytical model. In addition to predicting well, AI-DARWIN allows one to obtain the

differential equation of the underlying mechanism. The process of denoising and data generation has a signi cant impact on the eventual model form and corresponding parameters. As per AI-DARWIN, the explicit symbolic model of the derivative $\left(\frac{3}{3C}\right)$ is obtained rst, which is then used to estimate the concentration $(\hat{C})$ of the species involved by numerical integration. The parameters and model form t perfectly when given noiseless data, where AI-DARWIN was tested with 14 simulated data points from the analytical solution given by Belohlav et al. [66], and subsequently used the cubic spline smoothing to obtain 10,000 data points for the mechanism identi cation exercise. For the noisy case, 14 experimental data points were used to generate 10,000 data points which are then provided to AI-DARWIN.

Hydrogenation of Methyl Esters

Figure 2.14: Hydrogenation of methyl esters—concentration of species (C) plotted against time (t), data, is obtained using cubic spline smoothing interpolation for generating densely distributed data from previously sparse data.

Consider the reaction scheme pertaining to the hydrogenation of methyl esters. This is given by:

$A \xrightarrow{k_1} B \xrightarrow{k_2} C \xrightarrow{k_3} D$. The underlying mechanism of this case study and toluene hydrogenation were modeled as following the Langmuir-Hinshelwood surface kinetics and corresponding models with parameters were obtained [66]. For noiseless data, these equations were simulated to obtain 8 data points and then cubic spline smoothing interpolation technique was used to generate 10,000 data points. For testing AI-DARWIN with noisy data, the 8 data points given in Belohlav et al. [66] were denoised using the same denoising mechanism as before. Results for this case study are depicted in Figure 2.14.

2.6 Conclusions

AI-DARWIN successfully identi ed the underlying model structure correctly for mechanisms when given noiseless data while also being provided domain-speci c guidance through the function library. Nonlinear mechanisms such as the Michaelis-Menten equation and the Monod equation were identi ed with some error. The model form obtained for these cases are lower-order representations of the complex nonlinear functions.

All example pathways such as common cause (Trambouze), common effect (fractional kinetics), and causal chains (series reactions) were identi ed with great precision. It must be noted that one does not need a complex tree-like architecture to de ne such relationships, but simple linear models (linear in parameters) suf ce to obtain good input-output relationships, despite the complexity in data transformations. Further, the algorithm ef ciently determines the relationship between the variables of interest for this transformation. For instance, on giving the time, position, concentration and divergence of the gradient in the Fick's law example, the algorithm predicts relationship between just the divergence of the gradient and the rate, yielding the exact value of the diffusivity. Note here that the divergence was estimated from the data as an add-on feature extractor.

In comparison, conventional GPSR methods perform a random search across all possibilities of feature and variables, in addition to estimating the parameters in the same manner. This has resulted in evaluating over 10^9 possible functions [77]. AI-DARWIN on the other hand, performed

Table 2.4: Comparison of the MISO process model with AI-DARWIN algorithm outputs - sparse and noisy (post- ltering). Fitness in predictions is reported with respect to the numerical integral of the process model.

Case study	Inputs	Output	Process Model	AI-DARWIN Output	Fitness (MSE & r^2)
Series Reactions in Batch Reactor 2.5.3	, ,	C– ! $\frac{3}{3C}$	$\frac{3}{3C} = \quad 0{\bullet}5$	$\frac{3}{3C} = \underset{\pm\, 0{\bullet}0042}{0{\bullet}437}$	$5{\bullet}112 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9974$
2.5.3	, ,	C– ! $\frac{3}{3C}$	$\frac{3}{3C} = 0{\bullet}5 \qquad 0{\bullet}2$	$\frac{3}{3C} = \underset{\pm\, 0{\bullet}0051}{0{\bullet}420} \qquad \underset{\pm\, 0{\bullet}0038}{0{\bullet}172}$	$2{\bullet}743 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9972$
2.5.3	, ,	C– ! $\frac{3}{3C}$	$\frac{3}{3C} = 0{\bullet}2$	$\frac{3}{3C} = \underset{\pm\, 0{\bullet}0044}{0{\bullet}201}$	$1{\bullet}653 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9995$
Electrically Heated Stirred Tank $\frac{3)}{3C}$ 2.5.3	)–)4	C–)! $\frac{3)}{3C}$	$\frac{3)}{3C} = 10 \quad 0{\bullet}2) \ , \ 0{\bullet}1)_4$	$\frac{3)}{3C} = \underset{\pm\, 0{\bullet}3295}{6{\bullet}563} \quad \underset{\pm\, 0{\bullet}0017}{0{\bullet}046}) \ , \ \underset{\pm\, 0{\bullet}0015}{0{\bullet}0393})_4$	$4{\bullet}618$ $10^{\ 6}$ $r^2 = 0{\bullet}9982$
$\frac{3)_4}{3C}$ 2.5.3	)–)4	C–4)! $\frac{3)_4}{3C}$	$\frac{3)_4}{3C} = 250 \quad)_4 \ , \ )$	$\frac{3)_4}{3C} = \underset{\pm\, 3{\bullet}7475}{233{\bullet}577} \quad \underset{\pm\, 0{\bullet}0154}{0{\bullet}886})_4 \ , \ \underset{\pm\, 0{\bullet}0154}{0{\bullet}872})$	$7{\bullet}907$ $10^{\ 5}$ $r^2 = 0{\bullet}9977$
Methanol-to-hydrocarbon process $\frac{3I_1}{3C}$ 2.5.3	$I_1 - I_2 - I_3$	$\frac{3I_1}{3C}$	$\frac{3I_1}{3C} = \quad I_1 \frac{12{\bullet}762 I_1 \ , \ 5{\bullet}2407 I_2}{1{\bullet}2176 I_1 \ , \ I_2}$	$\frac{3I_1}{3C} = \quad I_1 \frac{\underset{\pm\, 0{\bullet}0033}{0{\bullet}56} I_1 \ , \ \underset{\pm\, 0{\bullet}0033}{9{\bullet}729} I_2}{\underset{\pm\, 0{\bullet}0047}{0{\bullet}0909} I_1 \ , \ I_2}$	$2{\bullet}24 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9943$
$\frac{3I_2}{3C}$ 2.5.3	$I_1 - I_2 - I_3$	$\frac{3I_2}{3C}$	$\frac{3I_2}{3C} = \frac{6{\bullet}381 I_1^2 \quad 5{\bullet}2407 I_1 I_2}{1{\bullet}2176 I_1 \ , \ I_2}$	$\frac{3I_2}{3C} = \frac{\underset{\pm\, 0{\bullet}0688}{3{\bullet}588} I_1^2 \quad \underset{\pm\, 0{\bullet}0077}{3{\bullet}025} I_1 I_2}{\underset{\pm\, 0{\bullet}0145}{0{\bullet}668} I_1 \ , \ I_2}$	$3{\bullet}64 \quad 10^{\ 5}$ $r^2 = 0{\bullet}9683$
$\frac{3I_3}{3C}$ 2.5.3	$I_1 - I_2 - I_3$	$\frac{3I_3}{3C}$	$\frac{3I_3}{3C} = \frac{5{\bullet}2407 I_1 I_2}{1{\bullet}2176 I_1 \ , \ I_2}$	$\frac{3I_3}{3C} = \frac{\underset{\pm\, 0{\bullet}0362}{6{\bullet}806} I_1 I_2}{\underset{\pm\, 0{\bullet}0156}{1{\bullet}946} I_1 \ , \ I_2}$	$1{\bullet}334 \quad 10^{\ 5}$ $r^2 = 0{\bullet}9962$
Toluene hydrogenation $\frac{3}{3C}$ 2.5.3	– –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{0{\bullet}0437 \quad , \ 0{\bullet}005}{1{\bullet}9 \quad , \quad , \ 1{\bullet}8}$	$\frac{3}{3C} = \frac{\underset{\pm\, 0{\bullet}0021}{0{\bullet}093} \ , \ \underset{\pm\, 0{\bullet}0001}{0{\bullet}0071}}{\underset{\pm\, 0{\bullet}1088}{4{\bullet}608} \ , \quad , \ \underset{\pm\, 0{\bullet}3344}{8{\bullet}612}}$	$5{\bullet}675 \quad 10^{\ 5}$ $r^2 = 0{\bullet}9985$
$\frac{3}{3C}$ 2.5.3	– –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{0{\bullet}0437 \quad 0{\bullet}016}{1{\bullet}9 \quad , \quad , \ 1{\bullet}8}$	$\frac{3}{3C} = \frac{\underset{\pm\, 0{\bullet}0048}{0{\bullet}0837} \quad \underset{\pm\, 0{\bullet}0017}{0{\bullet}0287}}{\underset{\pm\, 0{\bullet}2704}{4{\bullet}434} \ , \quad , \ \underset{\pm\, 0{\bullet}3144}{3{\bullet}951}}$	$3{\bullet}888 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9855$
$\frac{3}{3C}$ 2.5.3	– –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{0{\bullet}011}{1{\bullet}9 \quad , \quad , \ 1{\bullet}8}$	$\frac{3}{3C} = \frac{\underset{\pm\, 0{\bullet}0003}{0{\bullet}0057}}{\underset{\pm\, 0{\bullet}0111}{0{\bullet}456} \ , \quad , \ \underset{\pm\, 0{\bullet}0088}{0{\bullet}654}}$	$2{\bullet}484 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9968$
Methyl esters Hydrogenation $\frac{3}{3C}$ 2.5.3	– – –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{1{\bullet}44}{\ , 28 \ , \ 1{\bullet}8 \ , \ 2{\bullet}9}$	$\frac{3}{3C} = \frac{\underset{\pm\, 0{\bullet}0002}{0{\bullet}013}}{\ , \ \underset{\pm\, 0{\bullet}0206}{0{\bullet}141} \quad \underset{\pm\, 0{\bullet}0049}{0{\bullet}01796} \ , \ \underset{\pm\, 0{\bullet}0087}{0{\bullet}121}}$	$1{\bullet}205 \quad 10^{\ 5}$ $r^2 = 0{\bullet}9354$
$\frac{3}{3C}$ 2.5.3	– – –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{1{\bullet}44 \quad 0{\bullet}84}{\ , 28 \ , \ 1{\bullet}8 \ , \ 2{\bullet}9}$	$\frac{3}{3C} = \frac{1{\bullet}44 \quad \underset{\pm\, 0{\bullet}0271}{0{\bullet}957}}{\underset{\pm\, 7{\bullet}1911}{14{\bullet}997} \ , \ \underset{\pm\, 1{\bullet}3601}{29{\bullet}999} \ , \ \underset{\pm\, 0{\bullet}2731}{1{\bullet}369} \ , \ \underset{\pm\, 0{\bullet}2481}{4{\bullet}994}}$	$1{\bullet}021 \quad 10^{\ 5}$ $r^2 = 0{\bullet}9953$
$\frac{3}{3C}$ 2.5.3	– – –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{0{\bullet}84 \quad 0{\bullet}162}{\ , 28 \ , \ 1{\bullet}8 \ , \ 2{\bullet}9}$	$\frac{3}{3C} = \frac{0{\bullet}84 \quad \underset{\pm\, 0{\bullet}0005}{0{\bullet}174}}{\underset{\pm\, 2{\bullet}0781}{3{\bullet}117} \ , \ \underset{\pm\, 0{\bullet}6861}{20} \ , \ \underset{\pm\, 0{\bullet}0293}{3{\bullet}452} \ , \ \underset{\pm\, 0{\bullet}0156}{2{\bullet}427}}$	$1{\bullet}084 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9980$
$\frac{3}{3C}$ 2.5.3	– – –	$\frac{3}{3C}$	$\frac{3}{3C} = \frac{0{\bullet}162}{\ , 28 \ , \ 1{\bullet}8 \ , \ 2{\bullet}9}$	$\frac{3}{3C} = \frac{0{\bullet}162}{\underset{\pm\, 0{\bullet}7891}{1} \ , \ \underset{\pm\, 0{\bullet}2841}{24{\bullet}703} \ , \ \underset{\pm\, 0{\bullet}0197}{2{\bullet}692} \ , \ \underset{\pm\, 0{\bullet}0102}{2{\bullet}384}}$	$1{\bullet}651 \quad 10^{\ 4}$ $r^2 = 0{\bullet}9979$

feature and variable selection from a prede ned set of choices which could be adapted as needed. It was then guided further towards a better solution by the use of statistical techniques—the use of which ensures that the next best solution will either be equivalent or better than the previous best solution. In essence, AI-DARWIN is a human-assisted mechanism discovery and model identi - cation system which exploits the domain knowledge of the user—it performs directed evolution of mechanistic models. Conventional GPSR methods, on the other hand, typically rely heavily on purely random search, hoping that the program will converge to the desired solution.

Through AI-DARWIN, it was shown how genetic programming coupled with statistical inferencing could be used for automated mechanism discovery and model identi cation. A key aspect of this work is the inclusion of a library of elementary functions, guided by rst-principles-based insights and understanding of the application domain that incorporate fundamental physicochemical mechanisms. By exploiting this knowledge base, the genetic algorithm automatically found the optimal combination of elementary functions that resulted in as simple a model as possible, as allowed by the data. For many applications in chemical engineering, such as typical reaction engineering problems, the use of models that are linear in parameters seems to suf ce. This strategy helps with regularization and prevents over tting.

For many problems in science and engineering, exploiting the domain knowledge is key to mechanism and model identi cation. Often, in model identi cation, one has some ideas and insights, based on past experience and intuition. This ought to be leveraged, instead of blindly resorting to some “black-box" machine learning technique that provides no insights. Incorporating such knowledge makes the model more reliable and easier to interpret. It also reduces the amount of data needed to develop the model. This is an important point as many problems in chemical engineering are not “big data" problems. Such elds have only limited data that needs to be leveraged with domain knowledge. This study demonstrates how one might accomplish that with automated mechanism discovery and model identi cation using machine learning.

Chapter 3: Data-driven Discovery of Systems of Differential Equations

Physicochemical systems in scienti c and engineering disciplines are governed by rst-principles knowledge in the form of constitutive relations, equations, among other functional transformations of the variables involved. As presented in the preceding chapter, such rst-principles knowledge is captured through differential equations which showcase the interdependency between the variable(s) involved and the change of variable(s) with respect to others. In reaction kinetics, this is exempli ed by a functional relationship between the rate of change of concentration and, typically, polynomial transformations of the concentrations of the species involved in a reaction. Similarly, in heat transfer, differential equations are used to present a mathematical snapshot of a dynamical system that is governed by a temperature difference, resulting in transfer of thermal energy. Likewise, in biological/biochemical systems, systems of differential equations are used to represent the complex signaling pathways—often resulting in tens to hundreds of such coupled differential equations.

Traditional approaches to modeling these systems often rely on rst-principles reasoning, deriving equations grounded in fundamental laws and empirical observations. While these methods yield interpretable and theoretically sound models, their formulation becomes increasingly challenging in the presence of high system complexity, sparse observations, or noisy data. Recent advances in data-driven methodologies, particularly in machine learning, have introduced powerful tools for discovering governing equations directly from observational data. While these approaches often excel in predictive accuracy, they frequently rely on opaque black-box models that obscure the underlying dynamics. This lack of transparency restricts their utility in applications where understanding the underlying mechanisms is as important as achieving accurate predictions.

Hybrid methods that combine rst-principles reasoning with data-driven techniques offer a promising alternative, enabling robust modeling approaches that balance interpretability with pre-

dictive capability. Such methods align with the growing demand for explainable and transparent systems in scienti c and engineering applications. Accordingly, it would be valuable to design a system that, given data, could yield systems of differential equations which attempt to model the underlying physicochemical phenomena, thereby providing insights into the same. Furthermore, if these differential equations could be obtained explicitly, and not as a surrogate via a black-box modeling approach, it could be used to probe further into the fundamentals driving the physicochemical phenomena.

In this chapter, a genetic algorithm-based symbolic model discovery engine is presented by extending the capabilities of the previously discussed AI-DARWIN, to uncover systems of differential equations—for both ordinary and partial differential equations. AI-DARWIN has demonstrated its ef cacy in identifying the underlying symbolic models by leveraging genetic feature extraction to explore unknown functional transformations, and produces mechanistically plausible and interpretable models. However, its application was previously limited to single equations. This work adapts and expands AI-DARWIN's functionality to model dynamical systems governed by both ordinary differential equations (ODEs) and partial differential equations (PDEs), introducing a methodology that reconstructs both the functional forms and parameters of such systems directly from data. The proposed framework employs genetic algorithms to explore a solution space de ned by user-speci ed functional transformations informed by domain knowledge. By incorporating these transformations, the method ensures the discovered models remain interpretable, grounded in rst-principles reasoning, and aligned with the physical, biological, or nancial underpinnings of the system. To further improve robustness, the framework integrates regularization techniques to mitigate over tting and incorporates hashing mechanisms to enhance computational ef ciency during redundancy checks and parameter estimation.

The effectiveness of this approach is demonstrated using synthetic datasets, which simulate complex dynamic behaviors under realistic conditions of noise and sparsity. The results validate the framework's ability to identify governing equations that are both accurate and interpretable, offering a scalable and reliable approach for advancing the modeling of dynamical systems in

scienti c and engineering applications.

3.1 Background

AI-DARWIN [55] is a genetic algorithm-based data-driven framework developed to identify symbolic models. The algorithm demonstrated strong performance in discovering interpretable algebraic models for physicochemical systems. This framework introduced an expressive model structure as follows:

$$H = 5_1(x; \#_1), \; 5_2(x; \#_2), \; \ldots, \; 5_m(x; \#_m) \tag{3.1}$$

where the parameters $\#_i$ could appear within the functional transformations $5_i(x; \#_i)$, enabling greater exibility in modeling complex nonlinear relationships. However, despite its broader applicability, AI-DARWIN could not address the challenges associated with dynamical systems governed by ODEs and PDEs. To expand the utility of symbolic regression in modeling dynamical systems, the present work adapts these techniques for discovering systems of ODEs and PDEs. In this setting, the goal is to recover the structure and parameters of coupled differential equations of the form:

$$\frac{3G_i}{3C} = 5_i(x; \#) \quad i = 1, 2, \ldots, " \tag{3.2}$$

where $G_i(C)$ are state variables that evolve over time, and 5_i de nes the governing dynamics using candidate expressions constructed from the user-de ned function library. This formulation introduces additional challenges compared to static systems, including handling coupled interactions between variables, accurately capturing temporal evolution, and dealing with sensitivity to initial conditions and noise.

In addition to ODEs, many real-world systems are more accurately modeled by PDEs, which describe the evolution of variables over both time and space. Extending symbolic regression frameworks to PDEs discovery enables modeling of spatiotemporal processes such as uid ow, chemi-

cal diffusion, and wave propagation. Prior work in this direction includes the discovery of governing equations for the Navier–Stokes system [78], the viscous Burgers' equation [79], and the heat and wave equations [80].

To support the discovery of partial differential equations, the framework incorporates partial derivatives into the candidate expressions. These terms are computed using second-order central nite differences. For a discretized space-time grid where 5_{8}^{9} denotes the value at spatial index 8 and time index 9, the approximations are:

$$
\begin{aligned}
\frac{m5}{mG} &\approx \frac{5_{8+1}^{9} - 5_{8-1}^{9}}{2\,G} \\
\frac{m5}{mC} &\approx \frac{5_{8}^{9+1} - 5_{8}^{9-1}}{2\,C} \\
\frac{m^2 5}{mG^2} &\approx \frac{5_{8+1}^{9} - 2\,5_{8}^{9} + 5_{8-1}^{9}}{G^2} \\
\frac{m^2 5}{mC^2} &\approx \frac{5_{8}^{9+1} - 2\,5_{8}^{9} + 5_{8}^{9-1}}{C^2}.
\end{aligned}
\tag{3.3}
$$

These enhancements allow the genetic algorithm to operate on systems involving both spatial and temporal derivatives, making the framework capable of discovering symbolic representations of both ODEs and PDEs from observed data. By integrating these capabilities, the extended methodology facilitates interpretable modeling of a wider class of dynamical systems, including those with multiscale, nonlinear, and spatially heterogeneous behavior.

3.2 Methodology for Discovering System of Differential Equations

3.2.1 Data Description

Dynamical systems often exhibit rich and complex behaviors that are highly sensitive to their initial conditions, as illustrated in the left panel of Figure 3.1, which depicts the phase-space plot of the Lotka-Volterra system for various initial conditions. This plot highlights the oscillatory dynamics of state variables over time C, and demonstrates how distinct initial conditions yield unique trajectories.

To evaluate the proposed methodology, synthetic datasets were generated by solving the governing ODEs and PDEs for the system under randomly selected initial conditions. Gaussian noise with a small variance was then added to the resulting trajectories to simulate noisy real-world observational data, as shown in the right panel of Figure 3.1. Incorporating noise into the synthetic data replicates measurement errors, environmental variability, among other stochastic in uences commonly encountered in practice. By incorporating such imperfections, the study aims to assess the robustness of the proposed framework in discovering accurate ODEs and PDEs, that are both interpretable and reliable. For each case study, the data was randomly split into training and test sets using an 80-20 split (80% of the total data was used as a training set, and the remaining 20% was used as the unseen test set).

Figure 3.1: Phase-space plot (left) and noisy synthetic data (right) for Lotka–Volterra system.

3.2.2 Search Space of Potential Systems of Differential Equations

The framework operates within a prede ned search space of candidate functional transformations that are designed to balance computational feasibility with the expressiveness needed to model complex dynamical phenomena. This search space is characterized by a set of con gurable parameters:

- : Maximum number of generations
- # : Maximum number of individuals per generation
- " : Maximum number of equations in a system
- : Maximum number of terms per equation
- : Maximum number of functions per term
- Function library: Set of allowed base functions, drawn from {const, linear, exp, sin, cos, tan, log, square, cube, rst- and second-order partial derivatives}

The selection of elementary and differential operators is guided by domain knowledge, enabling the framework to search within a space of expressions that are not only expressive but also meaningful from rst-principles. This improves both convergence and interpretability while mitigating the risk of over tting to unrealistic or overly complex functional forms, a common limitation of unconstrained symbolic regression [56]. By de ning a constrained yet exible search space, the framework supports the discovery of governing equations for a broad range of systems, including those described by ordinary and partial differential equations.

3.2.3 Engineering the Loss Function

The proposed framework employs a systematic and robust approach for estimating the parameters of governing ODEs and PDEs. By integrating a well-de ned loss function, regularization, and an effective numerical optimization scheme, the method ensures the discovery of models that are both accurate and interpretable. During model tting on the training set, the framework minimizes the discrepancy between simulated system trajectories and observed data using the mean squared error (MSE), a widely adopted metric that quanti es the average squared difference between predicted and observed values:

$$"(\quad {}^{1}\#^{o} = \frac{1}{=}\sum_{8=1}^{\tilde{O}} k y_{pred}{}^{1}G_{8}; \#^{o} \quad y_{true}{}^{1}G_{8}^{o} k^{2} - \tag{3.4}$$

where $y_{pred}(t_8; \theta)$ denotes the predicted system output at time t_8 given parameters θ, $y_{true}(t_8)$ is the corresponding observed data, and N is the total number of time points.

To discourage over tting and promote interpretability aligned with rst-principles reasoning, the framework incorporates an L_2-norm regularization term:

$$\Omega(\theta) = \lambda \sum_{g=1}^{K} \theta_g^2 \tag{3.5}$$

where λ controls the regularization strength, θ_g is the g-th model parameter, and K is the total number of parameters. Larger values of λ impose a stronger penalty, encouraging sparsity and simplicity at the potential cost of exibility.

The total objective minimized during training is a composite loss:

$$L(\theta) = \alpha \cdot MSE(\theta) + \Omega(\theta) \tag{3.6}$$

where $L(\theta)$ balances the trade-off between model accuracy captured by the MSE and simplicity enforced by regularization.

Following training, model performance is evaluated on a separate test set using the coef cient of determination (R^2), which quanti es the proportion of variance in the observed data explained by the model:

$$R^2 = 1 - \frac{\sum_{t=1}^{N} \| y_{pred}(t_t; \theta) - y_{true}(t_t) \|^2}{\sum_{t=1}^{N} \| y_{true}(t_t) - \bar{y}_{true} \|^2} \tag{3.7}$$

where $\bar{y}_{true}$ is the mean of the true data. A higher R^2 score indicates closer alignment between predicted and true trajectories, and serves as the primary evaluation metric across the case studies.

3.2.4 Parameter Estimation

The parameter estimation process is designed to identify the optimal set of parameters θ that minimizes the objective function $L(\theta)$. This is achieved through the following steps:

Initialization: The parameters θ are initialized with an initial guess, typically a vector of zeros.

This choice ensures an unbiased starting point for the optimization process, allowing for thorough exploration of the parameter space without predisposing the model to speci c parameter values.

Simulation: Numerical simulation is employed to compute the model's predicted trajectories y_{sim} ¹Ğ, #º for a given set of parameters. The simulation is processed by integrating the systems of differential equations over a speci ed time range. Among the various integration methods available, the Radau method was selected for its ability to handle stiff and differential-algebraic problems effectively. The Radau method, being an implicit Runge–Kutta scheme, offers enhanced stability and robustness compared to explicit methods, particularly in systems characterized by tight coupling or simultaneous slow and fast dynamics. This ensures reliable simulation results even for challenging dynamical systems.

Figure 3.2: Plots depicting the computational complexity (left) and minimization failures (right) for varying optimization algorithms.

Optimization: The objective function ¹#º is minimized by employing the Broyden–Fletcher–Goldfarb–S (BFGS) algorithm. BFGS is a quasi-Newton method that uses the full Hessian approximation to provide fast convergence for systems with fewer parameters. Its computational ef ciency, as illustrated in Figure 3.2, makes it a practical choice for smaller-scale problems commonly encountered in symbolic regression tasks.

For more complex systems, especially those involving stiff dynamics or highly non-smooth

objective landscapes, the framework switches to the Nelder-Mead algorithm. This gradient-free optimization method excels in robustness but comes with higher computational costs. By employing Nelder-Mead selectively, the framework ensures reliable performance for challenging cases where the BFGS algorithm may fail to converge or become trapped in local minima. The ex-ibility to alternate between BFGS for computational ef ciency and Nelder-Mead for robustness allows the framework to adapt to a diverse range of dynamical systems, thereby balancing speed and accuracy.

3.2.5 Genetic Algorithm

Genetic algorithm (GA) forms the core of the proposed framework for discovering governing equations of dynamical systems. Inspired by the principles of natural selection, GA iteratively evolves a population of candidate individuals (here, each individual is a candidate system of differential equations) to identify optimal ODEs and PDEs that best describe observed data. This evolutionary process balances exploration of the solution space with exploitation of high-performing candidates, ensuring ef cient convergence to interpretable models. An overview of the key genetic operations—elitism, mutation, and crossover—is illustrated in Figure 3.3.

Figure 3.3: Illustration of the genetic operations—elitism, crossover, and mutation—in the proposed data-driven systems of equations discovery framework.

Algorithm Initialization: The algorithm begins by randomly generating an initial population of candidate systems, each structured according to the constraints and parameters speci ed in the search space. This initial population serves as the foundation for subsequent evolutionary steps.

Selection: The selection process determines which individuals from the current population will continue unchanged into the next generation. This step is guided by a tness score, which is inversely proportional to the objective function combining the MSE and a regularization term. Candidates with lower values are preferred during selection. The probability of selecting an individual is computed using a rank-based weighting scheme:

$$P_i = \frac{W^{rank_i - 1}}{\sum_{j=1}^{N} W^{rank_j - 1}} \tag{3.8}$$

where is the probability of selecting individual , is the rank of individual in the population with rank 1 being the best, and is a tunable parameter that controls selection pressure. When = 1, all individuals have uniform selection probabilities (= 1 for all ranks). As decreases the weighting becomes more biased toward individuals with higher ranks, with selection probabilities increasing exponentially for the top performers. In this study, is set between 0.6 and 0.8 to reach a balance between exploration (diversity through discovery in the population) and exploitation (by favoring top-performing individuals) to increase the likelihood of discovering globally optimal systems. Lower accelerates convergence but may risk stagnation, while a higher maintains diversity but slows convergence.

Genetic Operators: The evolutionary process is driven by a combination of genetic operators. Elitism ensures the retention of the top-performing individuals by directly carrying them over to the next generation without modi cation, preventing high-quality solutions from being lost due to stochastic processes. Crossover promotes the exchange of traits between two parent systems by randomly splitting equations and combining their segments, creating new individuals that inherit features from both parents. To maintain diversity and prevent premature convergence, mutation introduces random alterations by replacing terms within equations with newly generated terms

from the prede ned library of allowable functions. Additionally, a new operator injects entirely novel candidates into the population to explore previously unseen regions of the search space. Together, these operators maintain a dynamic balance between exploitation of strong candidates and exploration of the broader model landscape.

3.2.6 Ef ciency

Ef ciency is a critical factor in the performance of the genetic algorithm, particularly as it scales to larger population sizes and explores increasingly complex candidate models. Allowing a wider range of functional forms and parameters enhances the algorithm's ability to identify robust solutions but introduces signi cant computational challenges. Redundancy checks and parameter estimation are computationally intensive tasks and can become prohibitive without proper optimization. To address these challenges, a hashing mechanism has been implemented to signi cantly improve computational ef ciency while preserving accuracy.

Figure 3.4: Execution time for generation of systems of equations with naive redundancy checks, for different sets of functions in the function library.

Redundancy checks are an essential step in the genetic algorithm work ow, ensuring diversity within the population and avoiding unnecessary evaluations of duplicate systems. However,

naive redundancy checks, which rely on symbolic simpli cation and direct equation-by-equation comparisons, are computationally expensive. As illustrated in Figure 3.4, the computational complexity of system generation increases exponentially with larger population sizes and more diverse functional sets. This growth arises from the increasing likelihood of redundancy as the search space expands, necessitating comprehensive comparisons to detect duplicates.

Figure 3.5: Illustration of the hashing mechanism used to address inef ciency as a result of the duplicates generated by the genetic algorithm.

To address this inef ciency, a hashing mechanism depicted in Figure 3.5, is employed to ef ciently identify and eliminate duplicate systems. Each mathematical term, such as $\frac{G_2}{G_1}$ or $\frac{G_2^2}{G_4}$, is assigned a unique integer ID through a term-to-ID mapping. This mapping ensures consistent representation of terms across all systems, dynamically adding new terms to the map as they are encountered. A system, represented as a collection of equations, is converted into a set of these unique term IDs. To normalize the representation and ensure mathematical equivalence, the term IDs in each equation are sorted. This normalized representation allows the algorithm to treat systems with identical terms but differing orders as duplicates. The sorted term IDs for all equations in a system are then hashed to generate a unique identi er for the entire system. By storing and maintaining these hashes, the algorithm can identify duplicate systems in constant time, even in large populations.

This ef ciency extends to parameter estimation, where previously solved systems can directly retrieve stored results, bypassing the need for redundant evaluations. Furthermore, candidate systems that exceed a prede ned error threshold are excluded from genetic operations, streamlining the population update process and further improving computational ef ciency. The computational

bene ts of these optimizations are evident in Figure 3.6. As generations progress, the execution time for systems without hashing grows steeply, re ecting the increasing cost of handling redundancy during genetic operations and reevaluations during parameter estimation in more complex populations. In contrast, systems employing hashing maintain relatively stable and signi cantly lower execution times, even for larger populations and complex functional sets.

Figure 3.6: Execution time for parameter estimation with and without hashing, over multiple generations.

The reduction in execution time per generation enables the algorithm to scale to larger populations and more complex function sets, making it well-suited for discovering governing equations in dynamical systems with real-world challenges such as noise and sparsity. The hashing mechanism, combined with the genetic algorithm's inherent exibility, ensures that the framework is both computationally ef cient and robust in identifying interpretable models for complex systems.

3.3 Results of the Data-Driven Systems of Differential Equations Discovery Engine

3.3.1 Discovering Ordinary Differential Equations

Lotka-Volterra Predator–Prey Model

The Lotka–Volterra model is a widely studied system of rst-order ordinary differential equations (ODEs) that describes predator–prey interactions in biological ecosystems [81]. Due to its mathematical simplicity and well-understood dynamics, it serves as a standard benchmark for evaluating dynamical model discovery. The system is given by

$$\begin{aligned} \frac{3G}{3C} &= UG \quad VGH \\ \frac{3H}{3C} &= \quad WH, XGH \end{aligned} \tag{3.9}$$

where G and H denote the prey and predator populations, U governs prey growth, V and X encode predation and predator reproduction rates respectively, and W is the natural predator decay rate.

To evaluate the proposed model discovery framework's ability to recover governing dynamics from observed data, simulations were conducted using parameters $U = \frac{2}{3}$, $V = \frac{4}{3}$, $W = X = 1$, with initial conditions $(G_0, H_0) = (1.8, 1.3)$, over the time interval $C \in [0, 10]$. Gaussian noise was added to the simulated trajectories to emulate measurement perturbations. The genetic algorithm was con gured to search for systems composed of two equations ($n = 2$), with candidate expressions constructed from progressively richer function sets drawn from the prede ned set of functions in the function library: {Linear, Exp, Sin, Cos, Tan, Log, Square}.

Figure 3.7 and Table 3.1 represent the best symbolic expressions discovered at Generations 5 and 10 with population size of 100. In earlier generations, models often incorporate extraneous nonlinearities or trigonometric functions, which deviate from the true governing structure. However, as evolution proceeds, selection pressure eliminates many of these super uous terms, producing models that closely approximate the true Lotka–Volterra equations, even from within overly expressive function spaces.

Figure 3.8 illustrates the minimum and average loss over generations for each function set.

Figure 3.7: Best estimates of Lotka-Volterra model at generation 5 and 10 for varying function sets. The function set details are listed in Table 3.1.

Figure 3.8: Minimum and average loss per generation for different functional sets in the Lotka-Volterra model.

Table 3.1: Best estimate equations and R^2 scores for different function sets at generations 5 and 10 in the Lotka–Volterra model.

Generation	#	Function Set	Best Estimate	R^2
—	–	TRUE	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	True
5	1	Linear	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	2	Exp, linear	$\frac{3G}{3C} = \quad 2{\bullet}56H, \quad 0{\bullet}63 4^{H} - \frac{3H}{3C} = \quad 1{\bullet}35H, \quad 0{\bullet}37G_{2} 4^{H}$	0.94
	3	Sin, exp, linear	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	4	Sin, cos, exp, linear	$\frac{3G}{3C} = 0{\bullet}63G \quad 1{\bullet}30GH \quad \frac{3H}{3C} = 1{\bullet}17G\sin^{1}H^{0} \quad 1{\bullet}17\sin^{1}H^{0}$	0.75
	5	Sin, cos, exp, linear, log	$\frac{3G}{3C} = \quad 0{\bullet}93H, \quad 0{\bullet}59\cos^{1}H^{0} - \frac{3H}{3C} = 0{\bullet}29H 4^{G} \quad 1{\bullet}01H$	0.95
	6	Sin, cos, exp, linear, square, log	$\frac{3G}{3C} = \quad 1{\bullet}04GH^{2}, \quad 0{\bullet}48G_{1} - \frac{3H}{3C} = \quad 0{\bullet}80H, \quad 0{\bullet}80GH$	0.98
	7	Sin, cos, tan, exp, linear, square, log	$\frac{3G}{3C} = \quad 1{\bullet}04GH, \quad 0{\bullet}60G - \frac{3H}{3C} = \quad 0{\bullet}80H, \quad 0{\bullet}80GH$	0.85
10	1	Linear	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	2	Exp, linear	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	3	Sin, exp, linear	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	4	Sin, cos, exp, linear	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	5	Sin, cos, exp, linear, log	$\frac{3G}{3C} = 0{\bullet}67G \quad 1{\bullet}33GH \quad \frac{3H}{3C} = \quad 1{\bullet}0H, \quad 1{\bullet}0GH$	0.99
	6	Sin, cos, exp, linear, square, log	$\frac{3G}{3C} = \quad 1{\bullet}04GH^{2}, \quad 0{\bullet}48G - \frac{3H}{3C} = \quad 0{\bullet}80H, \quad 0{\bullet}80GH$	0.98
	7	Sin, cos, tan, exp, linear, square, log	$\frac{3G}{3C} = \quad 1{\bullet}80GH, \quad 0{\bullet}88G - \frac{3H}{3C} = \quad 0{\bullet}71H, \quad 0{\bullet}75G\sin^{1}H^{0}$	0.94

Simpler sets, such as [Linear] and [Exp, Linear], lead to rapid convergence to the true dynamics, with near-zero loss within a few generations. In contrast, broader sets like [Sin, Cos, Exp, Linear, Log] exhibit slower convergence due to the higher dimensionality of the search space.

The experiment highlights the algorithm's ability to recover sparse, interpretable equations from noisy data, even when initialized with redundant or misleading function sets. It also underscores the trade-off between functional expressivity and convergence speed. While richer function sets can, in principle, better capture nonlinear dynamics, they are also more prone to over tting or premature convergence to local optima. This suggests that augmenting the search with regularization techniques and redundancy elimination can be critical to maintaining generalizability.

Lotka-Volterra Model with Logistic Growth

To assess the framework's adaptability to nonlinear growth constraints, the classical Lotka–Volterra model is extended by incorporating a logistic growth term in the prey dynamics [81]. This modi- cation re ects ecological scenarios where population expansion is limited by a carrying capacity

, yielding the following form:

$$\begin{aligned} \frac{3G}{3C} &= UG\,1 \quad \frac{G}{\ } \quad VGH \\ \frac{3H}{3C} &= \quad WH_{¸}\,XGH \end{aligned} \tag{3.10}$$

The initial values were set to $G_0 = 1$, $H_0 = 1.3$. The parameters were set to $U = \frac{2}{3}$, $V = \frac{4}{3}$, $X = 1$, $W = 1$, $= 50$ over time interval $C2 \; 0–20$. The search space was con gured to two equations (= 2) over 15 generations with a population size of 50. The allowed functions were drawn from {Linear, Quad, Sin, Cos, Tan, Exp, Log}. Tables 3.2 and 3.3 present the best- tting equations discovered from noise-free and noisy observations across various function sets. In both cases, the framework consistently recovers structurally accurate systems, with performance metrics exceeding ^{2} ¡ 0.96, validating its robustness under perturbations.

Table 3.2: Lotka-Volterra Equations with Logistic Growth, No Noise. Best estimate equations and ^{2} values for different function sets tted to noise-free data.

Function Set	Best Estimate	^{2}
Linear, quad	$\frac{3G}{3C} = 0.67G \quad 0.013G^2 \quad 1.33GH, \frac{3H}{3C} = \quad 1.00H_{¸}\,1.00GH$	True
Linear	$\frac{3G}{3C} = 0.019H_{¸}\,0.65G \quad 1.36GH, \frac{3H}{3C} = \quad 0.99H_{¸}\,0.99GH$	0.99
Linear, quad	$\frac{3G}{3C} = 0.67G \quad 0.013G^2 \quad 1.33GH, \frac{3H}{3C} = \quad 1.00H_{¸}\,1.00GH$	1.00

Table 3.3: Lotka-Volterra equations with logistic growth, with noise. Best estimate equations and ^{2} values for different function sets tted to noisy data.

Function Set	Best Estimate	^{2}
Linear, quad	$\frac{3G}{3C} = 0.67G \quad 0.013G^2 \quad 1.33GH, \frac{3H}{3C} = \quad 1.00H_{¸}\,1.00GH$	True
Linear	$\frac{3G}{3C} = 0.018G_2{}_{¸}\,0.65G \quad 1.34GH, \frac{3H}{3C} = \quad 0.99H_{¸}\,0.99GH$	0.97
Linear, quad	$\frac{3G}{3C} = 0.678G \quad 0.018G^2 \quad 1.35GH, \frac{3H}{3C} = \quad 0.99H_{¸}\,0.99GH_{¸}\,0.008H^2$	0.97
Linear, quad, sin, cos, tan	$\frac{3G}{3C} = 0.62G_{¸}\,0.005H^2 \quad 1.28GH, \frac{3H}{3C} = \quad 0.63H_{¸}\,0.41G^2H$	0.96
Linear, quad, sin, cos, tan, exp, log	$\frac{3G}{3C} = 0.03H_{¸}\,0.64G \quad 1.39GH, \frac{3H}{3C} = \quad 1.01H_{¸}\,0.99GH_{¸}\,0.022H^2$	0.97

To further illustrate consistency with the original system structure, the top ve estimates ob-

tained using only linear and quadratic terms are shown in Table 3.4. Their simulated trajectories exhibit close alignment with the true dynamics, as visualized in Figure 3.9.

Table 3.4: Best estimate for the Lotka-Volterra model with logistic growth using noisy data.

System	φ^2
$\frac{dG}{dt} = 0.67G - 0.013G^2 - 1.33GH$, $\frac{dH}{dt} = -1.00H + 1.00GH$	True
$\frac{dG}{dt} = 0.678G - 0.018G^2 - 1.35GH$, $\frac{dH}{dt} = -0.99H + 0.99GH + 0.008H^2$	0.97
$\frac{dG}{dt} = 0.678G - 0.0165G^2 - 1.35GH$, $\frac{dH}{dt} = -0.99H + 0.99GH - 0.0006G^2$	0.95
$\frac{dG}{dt} = 0.674G - 0.0138G^2 - 1.35GH$, $\frac{dH}{dt} = -0.99H + 0.99GH + 0.002G^2H$	0.96
$\frac{dG}{dt} = 0.66G - 0.02H^2 - 1.38GH$, $\frac{dH}{dt} = -0.99H + 0.98GH + 0.0011G$	0.97
$\frac{dG}{dt} = 0.65G - 0.004G^2 - 1.33GH$, $\frac{dH}{dt} = -0.6H + 0.39G^2H + 0.002H^2$	0.96

Figure 3.9: Comparison of the true model with five best estimates as obtained for the Lotka-Volterra model with logistic growth systems, for noisy data.

Susceptible-Infected-Recovered Model

The Susceptible–Infected–Recovered (SIR) model is a foundational framework in mathematical epidemiology that describes the temporal dynamics of infectious disease transmission in a

closed population [82]. It captures interactions among three compartments—susceptible (S), infected (I), and recovered (R)—and is governed by the following set of coupled nonlinear ODEs:

$$\begin{aligned} \frac{3(}{3C} &= \quad V\ (\\ \frac{3}{3C} &= V\ (\quad W \\ \frac{3'}{3C} &= W \end{aligned} \tag{3.11}$$

where V and W denote the disease transmission and recovery rates, respectively. The model assumes homogeneous mixing and constant total population.

To evaluate the proposed framework on a higher-dimensional system, synthetic trajectories were generated using parameters $V = 0{\bullet}0002$, $W = 0{\bullet}04$, and initial conditions $^{1}(_{0} - _{0} - '_{0}{}^{o} = {}^{1}997 - 3 - 0^{o}$ over the time interval $C2 \ \text{»} 0 - 100\text{¼}$. Gaussian noise was added to emulate measurement perturbations. The search space was con gured for three equations ($" = 3$), and the population size was increased to 200 to support broader exploration across multiple functional sets.

Figure 3.10: Best estimates for the SIR Model using varying function sets, obtained at generations 10 and 20. The function set details are listed in Table 3.5.

Figure 3.10 illustrates that function sets containing only basic terms, such as [Linear] and [Linear, Exp], converge rapidly to accurate representations of the SIR dynamics, even in the presence of noise. While richer function sets initially produce redundant or irrelevant expressions, these extraneous terms are pruned over successive generations. By generation 20, most con gurations

recover equations that closely align with the ground-truth system.

Table 3.5 provides the best discovered models and corresponding R^2 scores, showing that the framework reliably identi es sparse and interpretable forms across functional sets. The loss trajectories in Figure 3.11 further support the framework's capacity to balance exploration and convergence, validating its robustness when applied to higher-dimensional dynamical systems such as SIR.

Table 3.5: Best estimate systems of equations, and corresponding R^2 scores for different function sets at Generations 10 and 20 in the SIR Model.

Generation	#	Function Set	Best Estimate	R^2
—	–	True	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	True
10	1	Linear	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	2	Exp, linear	$\frac{3G_1}{3C} = 0{\cdot}0007G_1G_3, 0{\cdot}00007G_14^{G_2} - \frac{3G_2}{3C} = 0{\cdot}03G_2, 0{\cdot}0005G_1G_3 - \frac{3G_3}{3C} = 0{\cdot}05G_2, 0{\cdot}0002G_1$	0.92
	3	Sin, exp, linear	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	4	Sin, cos, exp, linear	$\frac{3G_1}{3C} = 0{\cdot}001\cos^4 G_2{}^{o} \quad 0{\cdot}0002G_1G_3 - \frac{3G_2}{3C} = 0{\cdot}01G_1 \quad 0{\cdot}008G_3 - \frac{3G_3}{3C} = 0{\cdot}005G_1, 0{\cdot}02G_3$	0.86
	5	Sin, cos, exp, linear, log	$\frac{3G_1}{3C} = 0{\cdot}04G_2 \quad 0{\cdot}007G_1 - \frac{3G_2}{3C} = 0{\cdot}02G_2, 0{\cdot}02G_1 - \frac{3G_3}{3C} = 0{\cdot}05G_2 \quad 0{\cdot}006G_1$	0.84
	6	Sin, cos, exp, linear, square, log	$\frac{3G_1}{3C} = 0{\cdot}08G_2, 0{\cdot}02G_3 - \frac{3G_2}{3C} = 0{\cdot}01G_1 \quad 0{\cdot}01G_3 - \frac{3G_3}{3C} = 0{\cdot}03G_2, 0{\cdot}007G_3$	0.89
	7	Sin, cos, tan, exp, linear, square, log	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2, 0{\cdot}000004G_1G_3$	0.92
20	1	Linear	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	2	Exp, linear	$\frac{3G_1}{3C} = 0{\cdot}08G_2 \quad 0{\cdot}001G_1G_3 - \frac{3G_2}{3C} = 0{\cdot}124G_2, 0{\cdot}001G_1G_3 - \frac{3G_3}{3C} = 0{\cdot}04G_2, 0{\cdot}016G_14^{G_2}$	0.95
	3	Sin, exp, linear	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	4	Sin, cos, exp, linear	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	5	Sin, cos, exp, linear, log	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	6	Sin, cos, exp, linear, square, log	$\frac{3G_1}{3C} = 0{\cdot}0002G_1G_2 - \frac{3G_2}{3C} = 0{\cdot}0002G_1G_2 \quad 0{\cdot}04G_2 - \frac{3G_3}{3C} = 0{\cdot}04G_2$	0.99
	7	Sin, cos, tan, exp, linear, square, log	$\frac{3G_1}{3C} = 0{\cdot}00004G_2G_3 \quad 0{\cdot}0005G_1G_3 - \frac{3G_2}{3C} = 0{\cdot}07G_2, 0{\cdot}0006G_1G_3 - \frac{3G_3}{3C} = 0{\cdot}04G_2, 4{\cdot}91G_1$	0.96

Figure 3.11: Minimum and average loss per generation using different function sets for the SIR model.