

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Scientific Machine Learning in Chemical Engineering

Analysis of Reactor Dynamics

KIAN HAJIREZA



CHALMERS

Department of Chemistry and Chemical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden, 2025

Scientific Machine Learning in Chemical Engineering
Analysis of Reactor Dynamics

KIAN HAJIREZA

© Kian Hajireza, 2025
except where otherwise stated.
All rights reserved.

Licentiatuppsatser vid Institutionen för kemi och kemiteknik
Chalmers tekniska högskola
Nr 2026:01.

Department of Chemistry and Chemical Engineering
Division of Chemical Engineering
Chalmers University of Technology
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2025.

Scientific Machine Learning in Chemical Engineering

Analysis of Reactor Dynamics

KIAN HAJIREZA

Department of Chemistry and Chemical Engineering

Chalmers University of Technology

Abstract

Traditional reactor modeling commonly relies on complete mechanistic knowledge, a requirement that in certain scenarios is deemed impractical due to incomplete kinetic models and complex non-idealities. This thesis addresses these limitations by incorporating mechanistic knowledge with data-driven modeling employing neural networks. Particularly, neural ordinary differential equations (neural ODEs) are adopted as a framework to learn the system dynamics. This framework is integrated with physics-informed constraints based on fundamental conservation laws. Two complementary strategies are developed: (i) identifying component mass balances and (ii) enforcing atom balance constraints. These principles are embedded as soft constraints during training and are shown to improve generalization capabilities and model robustness. These approaches are benchmarked against their data-driven counterparts and when relevant against a kinetic model.

Comprehensive evaluations using synthetic data demonstrate that physics-informed regularization improves generalization capabilities when data is scarce. The performance gap diminishes when the amount of available data increases, however the improved model robustness remains. When considering real-world data, employing physics-informed constraints demonstrates a decrease in epistemic uncertainty and improved predictive accuracy compared to a kinetic model. The incorporation of mechanistic knowledge in the form of soft constraints requires manual tuning of the regularization weights. This can be computationally demanding since a set of different regularization weights need to be tested. It is therefore proposed to integrate the regularization weights into the training process by reformulating the traditional optimization problem for training network weights by performing gradient ascent on the regularization weights. This automatically tunes the regularization weights and circumvents manual tuning. The proposed methods are validated and tested on both synthetic and real-world data, thus demonstrating their potential for real-world applications. By bridging domain knowledge and machine learning, this work establishes physics-informed neural ODEs as a viable tool for chemical engineers facing scarcely available data and limited system knowledge.

Keywords

Neural Ordinary Differential Equations, Scientific Machine Learning, Physics-Informed Machine Learning, Kinetic Modeling

Acknowledgments

This work was financially supported by Swedish Foundation for Strategic Research (Grant: FID20-0002), which is gratefully acknowledged. I also acknowledge the EuroHPC Joint Undertaking for awarding this project access to the EuroHPC supercomputer LEONARDO, hosted by CINECA (Italy) and the LEONARDO consortium through an EuroHPC Development Access call.

I would like to express my sincere gratitude to my supervisors, Professor Ronnie Andersson and Dr. David Eklund, for giving me the opportunity to pursue my doctoral studies and for their continuous guidance throughout this journey. Your support has been invaluable in helping me develop as a researcher, by promoting analytical depth and inspiring me to explore new methodologies and perspectives. The constructive feedback and insightful discussions we shared have greatly shaped the quality of this work.

I am grateful to my co-authors for their collaboration and valuable input in the papers included in this thesis.

I would like to also thank my colleagues at RISE Research Institutes of Sweden in the Applied Digitalization unit for the engaging discussions and for their helpfulness whenever I needed assistance. Their help has been important in navigating the research process and creating a collaborative environment.

Grateful acknowledgment is also extended to the EuroCC National Competence Center Sweden (ENCCS) for their support in helping me get started with high-performance computing.

A special thank you to my fiancée for her constant support and being by my side through every part of this process. Finally, I would also like to thank my parents and my brother for their encouragement throughout this journey.

Stockholm, December 2025

Kian Hajireza

List of Publications

Appended publications

This thesis is based on the following publications:

Paper I:

K. Hajireza, D. Eklund, R. Andersson

“Enhancing generalization and training efficiency of neural ordinary differential equations for chemical reactor modeling.” *Chemical Engineering Journal* 519 (2025): 165183.

Contributions:

- Collaborative planning and scope definition with co-authors.
- Executions of all numerical experiments.
- Joint analysis of results and manuscript preparation.
- Led manuscript drafting and coordinated revisions with co-authors.

Paper II:

K. Hajireza, D. Eklund, L. Olsson, D. Creaser, R. Andersson

“Advancing kinetic modeling of hydrodeoxygenation of stearic acid through self-adaptive physics-informed neural ODEs.” Manuscript (2025).

Contributions:

- Collaborative planning and scope definition with co-authors.
- Executions of all numerical experiments.
- Joint analysis of results and manuscript preparation.
- Led manuscript drafting and coordinated revisions with co-authors.

List of Conference Contributions

Part of this work has been presented at the following scientific conferences:

- **K. Hajireza**, D. Eklund, R. Andersson, *Strategies for robust and efficient scientific machine learning in reaction engineering*, Oral presentation at the 28th International Symposium on Chemical Reaction Engineering (ISCRE 28), Turku, Finland, June 16–19, 2024.
- **K. Hajireza**, D. Eklund, R. Andersson, *Learning the batch reactor dynamics using neural ordinary differential equations*, Poster presentation at the PhysML24 Workshop on Physics-informed Machine Learning, Oslo, Norway, May 14–16, 2024.

Contents

Abstract	i
Acknowledgement	iii
List of Publications	v
List of Conference Contributions	vii
1 Introduction	1
1.1 Motivation	1
1.2 Objectives and Scope	2
1.3 Thesis Outline	2
2 Background	3
2.1 Reactor Modeling	3
2.2 Scientific Machine Learning	5
2.3 Scientific Machine Learning Applications in Reactor modeling	8
3 Methodology	9
3.1 Model Systems	9
3.1.1 System I: Synthetic Batch Reactor Model	9
3.1.2 System II: HDO of Stearic Acid	10
3.2 Physics-Based Regularization Strategies	10
3.2.1 Automated Discovery of Conservation Laws	10
3.2.2 Atom Balance Constraints	11
3.3 Training and Mathematical Optimization Approaches	12
3.3.1 Two-Stage Training with Pre-training	12
3.3.2 Learning Regularization Weights	13
3.4 Validation Strategies	14
3.4.1 Design of Experiments Approach	14
3.4.2 Leave-One-Out Strategy	14
3.4.3 Uncertainty Estimation	15
3.5 Software	17

4	Results and Discussion	19
4.1	Incorporation of Physics-Based Regularization	19
4.2	Pre-training	25
4.3	Uncertainty Estimation	27
5	Conclusion and Future Work	33
5.1	Conclusion	33
5.2	Future Work	34
	Bibliography	35
	Appended Papers	41
	Paper I	
	Paper II	

Chapter 1

Introduction

1.1 Motivation

Traditional reactor design commonly builds upon mechanistic models derived from mass and energy balances, which manifest as coupled nonlinear ordinary differential equations (ODEs) or partial differential equations (PDEs). While these models offer interpretability and physical consistency, they present some limitations such as the inability to capture unknown side reactions, axial dispersion, dead zones or catalyst deactivation.

Concurrently, machine learning has revolutionized domains such as computer vision and natural language processing by learning complex patterns directly from data. However, purely data-driven models can produce physically implausible predictions and typically require large datasets. In turn, this limits their direct applicability in chemical engineering contexts where data may be scarce.

To leverage the strengths of traditional reactor modeling and machine learning, scientific and physics-informed machine learning seeks to combine ODEs and PDEs derived from traditional reaction engineering with neural networks constrained by physical principles. In doing so, scientific machine learning offers:

- Enhanced predictive accuracy by correcting model-form errors.
- Ability to handle scenarios where mechanistic models are impossible to formulate.
- Reduced data requirements through physics-informed regularization.
- Computational acceleration via surrogate models for inference using a set of conditions for training.

Despite these advantages, several fundamental challenges remain in the effective implementation of physics-informed modeling approaches, particularly regarding the optimal integration of physical constraints and the computational efficiency of training procedures, which this thesis aims to investigate.

1.2 Objectives and Scope

This study addresses the following research question:

- To what extent can the incorporation of physics-informed model components in a data-driven model further enhance generalization capabilities?

Decomposing the primary research question into sub-questions:

- How do physics-informed and data-driven models compare in terms of robustness when trained on corrupted data?
- How can the balance between mechanistic model components with data-driven model components be efficiently chosen?

1.3 Thesis Outline

The thesis is organized as follows. Chapter 2 introduces the challenges associated with traditional reactor modeling and discusses various scientific machine learning approaches that can complement or serve as surrogates to traditional reactor models, including examples from recent research. Chapter 3 outlines the methodologies employed, through incorporation of physics-based regularization informed by various forms of first-principles. Chapter 4 presents and discusses the results for the various case studies. Finally, Chapter 5 summarizes the findings and proposes appropriate directions for future research.

Chapter 2

Background

Differential equations can be used to model complex chemical phenomena. Extending classical approaches to differential equations, it is of interest to incorporate data-driven modeling techniques with mechanistic approaches in order to properly leverage available sensor data. This integration forms the basis of scientific machine learning [1].

2.1 Reactor Modeling

The development of accurate reactor models is imperative to understanding, controlling, optimizing and scaling up chemical processes. Reactor models serve as mathematical representations of the physical and chemical phenomena occurring within reaction systems [2]. However, the construction of these models present practical challenges that ultimately motivate alternative methods such as approaches derived from scientific machine learning [3].

For a batch reactor system, the fundamental governing equations describe the temporal evolution of species concentrations and temperature through differential mass and energy balances. The general form of the species mass and energy balance with state variables concentration denoted as c and temperature as T are expressed in Eqs. 2.1-2.2:

$$\frac{dc_s}{dt} = \sum_i \nu_{is} r_i \quad (2.1)$$

$$\frac{dT}{dt} = \sum_i \frac{r_i (-\Delta H_i)}{\rho c_p} \quad (2.2)$$

where ν_{is} is the stoichiometric coefficient matrix for species s in reaction i , r_i is the rate of reaction i , c_p is the specific heat capacity, ρ is density and ΔH_i is the reaction enthalpy. This system of ODEs must be solved simultaneously for all species in the reaction network and for temperature. The reaction rate varies exponentially with respect to temperature through an Arrhenius expression, which thus requires solving a coupled non-linear ODE system for non-isothermal reaction conditions.

The complexity increases when considering realistic reaction systems. For hydrodeoxygenation (HDO) processes, the set of reactions involve multiple parallel and sequential pathways including direct HDO, decarboxylation, decarbonylation, hydrogenation and various side

reactions [4]. Each reaction pathway requires its own kinetic expression, typically following a Langmuir-Hinshelwood formulation to account for surface adsorption phenomena as shown in Eq. 2.3:

$$r_i = \frac{k_i \prod_j c_s^{\alpha_{ijs}}}{(1 + \sum_m K_m c_m)}, \quad (2.3)$$

where k_i is the temperature-dependent rate constant, K_m is the adsorption equilibrium constants and the exponents, α_{ijs} represent reaction orders. These rate expressions introduce several model parameters that must be estimated through non-linear parameter regression [5], [6].

One of the challenges in reactor modeling is the selection of rate law model structure. Prior to parameter estimation, the functional form of each reaction rate must be chosen, whether power-law, Langmuir-Hinshelwood, Michaelis-Menten or another mechanistic form. This structural assumption constrains the mathematical relationships the model can express. Subsequently, kinetic parameters are determined through non-linear parameter regression to fit experimental data within the constraints of the chosen structure [7]–[11]. If the assumed rate law model structure is fundamentally incorrect, no amount of parameter optimization can compensate for this. An incorrect functional form limits the model’s representational capacity, introducing systematic errors that propagate through the model predictions. For example, attempting to fit saturation kinetics with a power-law expression introduces inductive bias that no parameter regression can eliminate. The functional form defines the space of behaviors the model can represent and data fitting only optimizes within that constrained space. An additional bottleneck involves computational expense in model solution. High-fidelity mechanistic models often require coupling the reaction kinetics with multiple additional physical phenomena. Examples of such phenomena include but are not limited to vapor-liquid equilibrium calculations using equations of state (e.g. PSRK) [12], mass transfer limitations at multiple scales (gas-liquid, liquid-solid), property calculations as functions of temperature, pressure and composition, additionally, catalyst deactivation, conjugate heat transfer between reactor walls and fluid contents and lastly, radiative heat transfer in high-temperature reactors. Incorporating these phenomena transforms the reactor model from a straightforward system of ODEs into a complex coupled problem requiring iterative solution of thermodynamic, transport and property correlations at each time step. For processes requiring modeling fluid flow through for instance computational fluid dynamics (CFD) [13] coupled with detailed kinetics and additional complex phenomena, a single reactor simulation may require hours to weeks of computational time on high-performance computing infrastructure [14], [15]. This limits such models for applications demanding real-time response.

The challenges outlined above create a motivation for exploring alternative modeling approaches. An ideal reactor modeling approach for modern applications would possess the following characteristics:

- **Computational efficiency:** Models should evaluate rapidly enough for real-time applications, ideally requiring seconds for inference rather than hours per simulation.
- **Robust generalization:** Models should maintain accuracy when applied to conditions beyond their training data and making predictions with confidence.

- **Scalability to complex systems:** The model structure should automatically adapt to the intrinsic complexity of the system with many species, reaction pathways and interactions without exponentially increasing development complexity.

Scientific machine learning approaches offer promising solutions for addressing these requirements. By leveraging neural networks’ universal approximation [16]–[19] capabilities while incorporating physical laws and constraints, physics-informed machine learning models can potentially learn complex reaction kinetics directly from experimental data. The subsequent sections explore how these techniques can be applied specifically to reactor modeling challenges, presenting the physics-informed machine learning framework employed in this thesis.

The fundamental challenge in chemical reactor modeling lies in the fact that while the governing equations based on mass, energy, momentum and charge balances are well-established from first principles, the reaction kinetics often remain unknown. This knowledge gap creates an ideal scenario for physics-informed machine learning approaches that combine known physics with data-driven components. In contrast to purely mechanistic models that require complete knowledge of reaction mechanisms and kinetic parameters or purely data-driven models that may violate physical constraints, scientific and physics-informed machine learning methods offer a framework for integrating partial mechanistic knowledge with machine learning [20], [21].

2.2 Scientific Machine Learning

Recently, methods in scientific machine learning such as neural ODEs [22], universal differential equations (UDEs) [23] and physics-informed neural networks (PINNs) [24], [25] have been introduced to bridge data-driven modeling with mechanistic modeling. These approaches enable integrating models that rely on differential equations together with deep learning architectures, leveraging the flexibility of neural networks, thus permitting them to capture complex phenomena that is difficult to model from first-principles due to knowledge that is partially available or uncertain.

In chemical reaction engineering, accurately modeling the reaction rate often proves to be an onerous task due to the complexity of underlying kinetics. This motivates leveraging the universal approximation theorem, to approximate the reaction rate using a neural network. For specifically batch reactors, this would correspond to modeling the entire right-handside with a neural network, $\mathcal{NN}$ as shown in Eq. (2.4), where the model parameters are denoted as θ :

$$\frac{dc_s}{dt} = \mathcal{NN}_s(\theta, \mathbf{c}(t)). \quad (2.4)$$

The output at any time t_1 given initial condition $\mathbf{c}(t_0)$ is obtained by solving the ODE using a standard ODE solver such as the Runge-Kutta methods:

$$c_s(t_1) = c_s(t_0) + \int_{t_0}^{t_1} \mathcal{NN}_s(\theta, \mathbf{c}(t)) dt. \quad (2.5)$$

Neural ODEs are particularly relevant for systems governed by continuous-time dynamics. Unlike Long-Short-Term Memory (LSTM) networks, neural ODEs can be evaluated at arbitrary time points, making them suitable for irregularly sampled data.

UDEs extend the neural ODE framework by embedding known differential equation structures into the right-handside of the differential equation, creating hybrid models that combine mechanistic knowledge with learned components. The key insight is to replace unknown or poorly characterized terms in differential equations with universal approximators, as demonstrated in Eqs. 2.6-2.7:

$$\frac{dc_s}{dt} = f_{\text{known},s} + \mathcal{NN}_s(\boldsymbol{\theta}, \mathbf{c}), \quad (2.6)$$

$$\frac{dc_s}{dt} = f_{\text{known},s} \cdot \mathcal{NN}_s(\boldsymbol{\theta}, \mathbf{c}), \quad (2.7)$$

where $f_{\text{known},s}$ represents the well-understood mechanistic term and the neural network learns the unknown or closure term from data. This formulation incorporates partial mechanistic knowledge while allowing the model to learn corrections or missing physics.

For a chemical reaction system, this approach is particularly powerful. Consider Eq. 2.8, showing a reactor model where most reactions are well-characterized but some kinetic expressions are uncertain:

$$\frac{dc_s}{dt} = \underbrace{\sum_{i=1} \nu_{is} r_s(\mathbf{c}, T)}_{\text{known kinetics}} + \underbrace{\mathcal{NN}_s(\boldsymbol{\theta}, \mathbf{c}, T)}_{\text{unknown kinetics}}. \quad (2.8)$$

The neural network learns the residual dynamics that account for unknown effects. When mechanistic models are incomplete, UDEs are an appropriate candidate for alternative modeling approaches since they preserve known physics, while integrating learnable model components. This hybrid structure maintains some interpretability through the mechanistic model component, while the neural network component is designated to learning specific unknown terms rather than the entire system dynamics. In chemical reactor modeling, UDEs are well-suited to modeling systems such as semi-batch reactors since the flow term is typically known, while the reaction rate terms remain uncertain.

PINNs take a different approach compared to neural ODEs and UDEs by directly modeling the state variable, $\mathbf{y}$ with a neural network, $\mathbf{y}(\mathbf{x}, t) \approx \mathcal{NN}(\boldsymbol{\theta}, \mathbf{x}, t)$ thus circumventing the use of an ODE solver. PINNs define their loss function by encoding known physical governing equations into the loss function. Consider Eq. 2.9, showing a general differential equation describing the system:

$$\frac{\partial \mathbf{y}}{\partial t} = \mathcal{F}(\mathbf{y}, \nabla \mathbf{y}, \mathbf{x}, t), \quad (2.9)$$

where $\mathcal{F}$ is the right-handside containing the physics and ∇ is the gradient. The key innovation for PINNs is the physics-informed loss function that combines three objectives and balances their penalty strength with a regularization strength denoted as λ , as presented in Eq. 2.10:

$$L(\boldsymbol{\theta}) = \lambda_{\text{data}} \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}}(\boldsymbol{\theta}) + \lambda_{\text{bc}} \mathcal{L}_{\text{bc}}(\boldsymbol{\theta}) + \lambda_{\text{init}} \mathcal{L}_{\text{init}}(\boldsymbol{\theta}), \quad (2.10)$$

where the data loss, $\mathcal{L}_{\text{data}}$ measures how well the model predictions match observations, for example by computing the mean-squared error. While the physics loss penalizes the neural network for deviations of the residual from zero:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N_{\text{physics}}} \left\| \frac{\partial \mathcal{NN}}{\partial t} - \mathcal{F}(\mathcal{NN}_{\boldsymbol{\theta}}, \nabla \mathcal{NN}_{\boldsymbol{\theta}}, \mathbf{x}, \mathbf{t}) \right\|_F^2, \quad (2.11)$$

where N_{physics} is the number of collocation points and $\|\cdot\|_F$ denotes the Frobenius norm. Computing the PDE residual in Eq. 2.11 requires determining the derivatives of the neural network output with respect to inputs, which is efficiently performed by utilizing automatic differentiation [26], [27]. Lastly, the boundary (Eq. 2.12) and initial condition (Eq. 2.13) loss encourages that the solution satisfies examples of the constraints:

$$\mathcal{L}_{\text{bc}} = \frac{1}{N_{\text{bc}}} \|\mathcal{NN}(\boldsymbol{\theta}, \mathbf{x}_{\text{boundary}}, \mathbf{t}_{\text{boundary}}) - \mathbf{y}_{\text{boundary}}\|_F^2, \quad (2.12)$$

$$\mathcal{L}_{\text{ic}} = \frac{1}{N_{\text{ic}}} \|\mathcal{NN}(\boldsymbol{\theta}, \mathbf{x}_{\text{initial}}, \mathbf{0}) - \mathbf{y}_{\text{initial}}\|_F^2, \quad (2.13)$$

where N_{ic} is number of spatial collocation points and N_{bc} is number of space-time collocation points.

PINNs have found several applications for systems with sparse or incomplete data. They can solve forward problems [24], [28], [29], inverse problems by inferring both unknown parameters and unknown model terms from data [30], [31] and lastly, for ill-posed problems that are missing initial and boundary conditions. Unlike traditional numerical methods that discretize the domain and solve on a mesh, PINNs do not require a mesh.

These three approaches represent distinct methodologies within scientific machine learning, each offering unique advantages for reactor modeling applications. Comparing the three approaches, neural ODEs and UDEs require solving differential equations during both training and inference, with gradients updated during training via the adjoint method [32] with backpropagation through the ODE solver. This couples the learning process to the ODE solver's stability and accuracy. In contrast, PINNs avoid explicit time integration, by leveraging automatic differentiation to compute residuals and integrating them into the loss function. For inverse problems, which is the common theme in chemical reaction engineering, neural ODEs and UDEs couple the state derivatives to the neural network through the ODE solver. Nevertheless, PINNs incorporate an additional neural network to model the state variables and the unknown term(s) independently. Understanding these distinctions is essential before examining their computational characteristics and practical implications. In practice, the choice depends on the specific modeling context: the availability and quality of data, the extent of mechanistic knowledge, the importance of interpretability and the computational resources available.

2.3 Scientific Machine Learning Applications in Reactor modeling

One of the first adoptions of neural ODEs in chemical kinetics was the ChemNODE framework [33], which demonstrated the capability of neural networks to accurately model chemical source terms in combustion systems. For batch reactor applications, which are of particular interest for this thesis, neural ODEs offer a natural framework for modeling time-dependent concentration profiles, when the kinetic rate expressions are unknown or complex, a neural network serve as flexible surrogates for estimating the reaction rate as demonstrated in Eq. 2.14:

$$r_s(\mathbf{c}, T) \approx \mathcal{NN}_s(\boldsymbol{\theta}, \mathbf{c}, T). \quad (2.14)$$

To extend the purely data-driven model, it is of interest to couple the neural ODE with mechanistic knowledge. For instance, Sorourifar et al. [34] coupled neural ODEs with known mechanistic information such as the stoichiometric matrix, incorporating the neural network to model the reaction rates for the individual reactions that simultaneously take place rather than modeling the reaction rates that produce and consume species. This can be beneficial when number of reactions does not exceed number of species by decreasing the dimensionality of the model. Other than the stoichiometric matrix, additional studies have incorporated Arrhenius temperature dependence, equilibrium conditions and zero-rate prediction in absence of certain reactants [35], exhibiting improved generalization capabilities compared to a traditional neural ODE. Kircher and Votsmeier [36] proposed Global Reaction Neural Networks (GRNN) that incorporates hard constraints such as stoichiometric and thermodynamic knowledge into the neural network architecture. Another neural ODE framework that embeds hard constraints into the neural network is the Chemical Reaction Neural Network (CRNN) [37], [38], which autonomously discovers reaction pathways and kinetic parameters by embedding the law of mass action and Arrhenius expression directly into its structure, where network parameters are interpretable and correspond to reaction orders, stoichiometric coefficients and rate constants. A limitation of this approach is that it assumes a power-law structure for the chemical kinetics. A recent study extended the CRNN framework by incorporating atom conservation as a hard constraint within the output layer [39].

As explained in Section 2.2, UDEs provide an approach to combining mechanistic model components with neural networks, encoding this formulation as a differential equation. For chemical reactors, this framework is particularly powerful because certain model components in the reactor mass and energy balances are typically well-known, while the reaction kinetics commonly remain uncertain. UDEs have been successfully applied across various chemical engineering systems. For instance, Bangi et al. [40] proposed a UDE that integrates prior knowledge of first-principles kinetics while employing a neural network to learn the unknown components. This approach demonstrated improved accuracy compared to existing mechanistic kinetic models. UDEs have also found applications in adsorption models, employing neural networks as a surrogate while maintaining the governing advection–diffusion PDE structure [41].

Chapter 3

Methodology

This work develops a progressive approach to integrating physical constraints into neural ODEs for chemical reactor modeling. The methodology is established using synthetic data with automatically discovered conservation laws, which is thereafter extended to real-world data with constructed atom balances and learnable regularization weights. Throughout both studies, the neural ODE framework serves as the core modeling approach with contrasting strategies for physics-based regularization, optimization and validation.

3.1 Model Systems

To demonstrate the proposed methodologies, two model systems are studied. **System I** is a synthetic batch reactor system, while **System II** investigates a real-world system, specifically the HDO of stearic acid as a model compound.

3.1.1 System I: Synthetic Batch Reactor Model

The system is a simplified model system and consists of three simultaneous non-isothermal reactions involving five chemical species (A , B , C , D , E) with both parallel and consecutive reactions:



where species C is an intermediate product and D and E are end products. The reaction rates follow Arrhenius kinetics with specified activation energies and pre-exponential factors (see Appendix A of the original paper). This model system possesses known analytical conservation laws while exhibiting sufficient complexity to evaluate the proposed discovery and regularization methodology. Importantly, the methodologies presented do not assume prior knowledge of these conservation laws or the reaction stoichiometry.

3.1.2 System II: HDO of Stearic Acid

System II investigates HDO of stearic acid illustrating the reaction pathways in Fig. 3.1, where heptadecane (C17) and octadecane (C18) are the end-products with octadecanal and octadecanol as intermediates. Unlike **System I**, **System II** utilizes real-world experimental data which was collected in the work by Arora et al. [7].

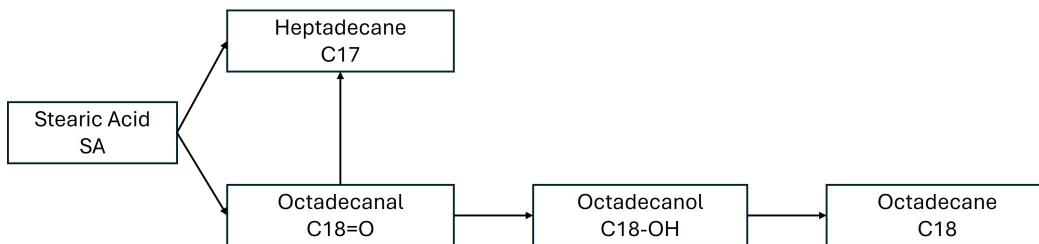


Figure 3.1: Reaction pathways for **System II** for HDO of stearic acid.

Henceforth, the results for the predictions of the end-products C17 and C18 will be lumped with their corresponding isomers and denoted as C17+ and C18+, respectively.

3.2 Physics-Based Regularization Strategies

The typical regularization approach for physics-informed models is commonly employed as shown in Eq. 3.4:

$$\mathcal{L}_{\text{total}}(\theta) = \mathcal{L}_{\text{data}}(\theta) + \lambda_{\text{physics}} \mathcal{L}_{\text{physics}}(\theta), \quad (3.4)$$

where $\mathcal{L}_{\text{data}}$ is computed using the mean-squared error and $\mathcal{L}_{\text{physics}}$ is a penalty for deviations from a physics-based constraint. Examples of such constraints include but are not limited to mass, atom, mole, energy, momentum or charge balances. The choice of the hyperparameter, λ_{physics} remains crucial to the training process and the loss landscape. For this reason, it is natural to incorporate it into the training process by permitting the model to learn and self-adaptively tune the regularization strength rather than manually tuning it.

3.2.1 Automated Discovery of Conservation Laws

To encourage physical consistency within the proposed model when the system dynamics is unknown and the training data is limited, it is proposed to address this challenge by automatically discovering conservation laws from temporal concentration data and embedding them as soft constraints within the total loss function. The discovery methodology exploits the fundamental property that solutions to chemical reactor systems satisfy linear conservation relations through a mass balance. For a batch reactor with five chemical species, with species concentrations $\mathbf{c} = [c_A, c_B, c_C, c_D, c_E]$, any linear combination satisfying a conservation law can be expressed as shown in Eq. 3.5:

$$\alpha c_A + \beta c_B + \gamma c_C + \delta c_D + \epsilon c_E = \phi, \quad (3.5)$$

where $\alpha, \beta, \gamma, \delta, \epsilon, \phi \in \mathbb{R}$. With $\mathbf{c} = [c_A, c_B, c_C, c_D, c_E]$ and $\boldsymbol{\xi} = (\alpha, \beta, \gamma, \delta, \epsilon)^T$, the equation is summarized in 3.6:

$$\mathbf{c}\boldsymbol{\xi} - \phi = 0. \quad (3.6)$$

where $\boldsymbol{\xi}$ is the coefficient vector. The identification of these relations proceeds by the computation of the null space of the centered data matrix, excluding the temperature dataset. The numerical rank of this matrix determines the dimension of its null space, which directly corresponds to the number of linearly independent conservation laws.

For **System I**, two independent conservation laws exist:

$$c_A - c_B + 2c_D = c_{A,0} - c_{B,0} + 2c_{D,0}, \quad (3.7)$$

$$2c_B + 2c_C + c_E = 2c_{B,0} + 2c_{C,0} + c_{E,0}. \quad (3.8)$$

Note that the discovered relations are not unique and are instead linear combinations of the true underlying conservation laws. By computing the null space of the centered data concentration matrix, a linear combination of Eqs. (3.7) and (3.8) is discovered without requiring *a priori* knowledge of reaction stoichiometry or molecular formulas. This distinguishes the approach from previous work that assumed known chemical structures. [39], [42]–[45].

Once the relations have been deduced, they are then applied as a regularization term in the loss function in similar manner as shown in Equation 3.4, with $\mathcal{L}_{\text{physics}} = \mathcal{L}_{\text{mass}}$. The discovered mass conservation regularization term, $\mathcal{L}_{\text{mass}}$ is defined as shown in Eq. 3.9:

$$\mathcal{L}_{\text{mass}}(\theta) = \frac{1}{LDK} \|\mathbf{C}\boldsymbol{\Xi} - \boldsymbol{\Phi}\|_F^2, \quad (3.9)$$

where D is the number of datasets, L is the number of discovered conservation laws, K is the number of collocation points where conservation laws are enforced. Additionally,

- $\mathbf{C} \in \mathbb{R}^{D \times K \times S}$ is a tensor containing all predicted concentrations from the neural ODE,
- $\boldsymbol{\Xi} \in \mathbb{R}^{S \times L}$ is a matrix whose columns contain the conservation law coefficients,
- $\boldsymbol{\Phi} \in \mathbb{R}^{D \times K \times L}$ is a tensor of constant terms.

The linear relations evaluated on the training data in the collocation points should theoretically be as small as possible, therefore the resulting squared residual is minimized during training.

3.2.2 Atom Balance Constraints

Unlike mass balances, which can be discovered from concentration data alone, atom balances require *a priori* knowledge of the molecular composition of each species. These constraints encourage that the predicted total number of moles of an atom remains invariant throughout the reaction. For a system with species s and atoms j , the atom balance is expressed in Eq. 3.10:

$$\sum_s \nu_s^j n_s(t) = \sum_s \nu_s^j n_{s,0}, \quad (3.10)$$

where ν_s^j is the number of atoms of type j in species s , $n_s(t)$ is the number of moles of species s at time t and $n_{s,0}$ is the initial number of moles in the starting material.

To incorporate this principle into the neural ODE, a soft physics-based regularization term is introduced that penalizes deviations from the initial number of moles of carbon by letting $\mathcal{L}_{\text{physics}} = \mathcal{L}_{\text{atom}}$. Specifically, for carbon conservation, the total loss function is:

$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \lambda_C \mathcal{L}_{\text{atom}}(\boldsymbol{\theta}), \quad (3.11)$$

where λ_C is a regularization weight and $\mathcal{L}_{\text{atom}}$ quantifies violations of carbon balance, as shown in Eq. 3.12:

$$\mathcal{L}_{\text{atom}}(\boldsymbol{\theta}) = \frac{1}{DK} \|\mathbf{C}_{\text{initial}} - \hat{\mathbf{C}}(\boldsymbol{\theta})\|_F^2, \quad (3.12)$$

where $\mathbf{C}_{\text{initial}}$ is a matrix of initial moles of carbon atoms where each row corresponds to a given dataset and the predicted carbon content at collocation point t_k is shown in Eq. 3.13:

$$\hat{\mathbf{C}}_{d,k}(\boldsymbol{\theta}) = \sum_{s=1}^S \nu_s^C \hat{n}_s(\boldsymbol{\theta}, t_k), \quad (3.13)$$

where ν_s^C is the number of carbon atoms in species s and $\hat{n}_s(\boldsymbol{\theta}, t_k)$ is the predicted number of moles for species s at time t_k . This regularization term encourages physical consistency by penalizing deviations from the initial moles of carbon atoms.

As discussed in **Paper II**, the atom balance is applied only to liquid-phase species in **System II**, assuming negligible moles of carbon atoms in gaseous components. This assumption is justified by their minimal contribution to overall carbon balance.

3.3 Training and Mathematical Optimization Approaches

This section of the thesis proposes two alternative training strategies for neural ODEs related to reducing the computational overhead and self-adaptively tuning of regularization weights for physics-informed neural ODEs.

3.3.1 Two-Stage Training with Pre-training

Training neural ODEs requires backpropagation through the ODE solver, which is computationally expensive. The adjoint sensitivity method is employed to compute gradients efficiently with constant memory cost. The adjoint state is defined as:

$$\mathbf{a}(t) = \frac{\partial \mathcal{L}}{\partial \mathbf{z}(t)}, \quad (3.14)$$

where $\mathbf{z}(t)$ represents the hidden state and $\mathcal{L}$ is the loss function. Specifically, the adjoint follows the differential equation:

$$\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^T \nabla_{\mathbf{z}} \mathcal{NN}(\boldsymbol{\theta}, \mathbf{z}(t)). \quad (3.15)$$

The gradient with respect to network parameters is computed by integrating backwards in time:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L} = \int_{t_1}^{t_0} \mathbf{a}(t)^T \nabla_{\boldsymbol{\theta}} \mathcal{NN}(\boldsymbol{\theta}, \mathbf{z}(t)) dt. \quad (3.16)$$

For scenarios with abundant data, a pre-training strategy can be employed to reduce computational cost. Pre-training trains the neural network utilizing estimates of the numerical derivatives of the state variables. This circumvents the need for repeated ODE solver calls, thus minimizing the loss function directly as a function of the system dynamics rather than of the individual state variables, as elucidated in Eq. 3.17:

$$\mathcal{L}_{\text{deriv}}(\boldsymbol{\theta}) = \frac{1}{D(S+1)\tau_{\text{obs}}} \|\mathbf{y}_{\text{deriv}} - \mathcal{NN}(\boldsymbol{\theta}, \mathbf{y}_{\text{obs}})\|_F^2, \quad (3.17)$$

where $\mathbf{y}_{\text{deriv}}$ and $\mathbf{y}_{\text{obs}}$ are third-order tensors over datasets, state variables and temporal observations. They represent the numerical derivatives estimated through spline interpolation and finite differences. $\mathcal{NN}(\boldsymbol{\theta}, \mathbf{y}_{\text{obs}})$ denotes the neural network predictions of the derivatives at the observed states and τ_{obs} is the number of temporal observations.

3.3.2 Learning Regularization Weights

To overcome the limitations of fixed scalar regularization weights, which require extensive tuning through grid search, a self-adaptive approach is introduced, which is inspired from the application on PINNs [46], [47]. Instead of a single weight, a set of learnable point-wise weights $\boldsymbol{\lambda} = (\lambda_1, \dots, \lambda_K)$ are employed, where each collocation point applies its own regularization weight. These weights adapt dynamically during training to emphasize regions where carbon conservation is hardest to enforce. Specifically, these weights are applied to the carbon balance constraint and the objective function is reformulated as:

$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \sum_k^K \frac{\lambda_k}{DK} \cdot \|\mathbf{C}_{\text{initial}} - \hat{\mathbf{C}}_k(\boldsymbol{\theta})\|_F^2. \quad (3.18)$$

Training performs gradient descent on network parameters and gradient ascent on the regularization weights. The gradients are presented in Eqs. 3.19-3.20:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{total}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{data}} + \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{atom}}, \quad (3.19)$$

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}_{\text{total}} = \frac{1}{DK} \sum_{d=1}^D \begin{bmatrix} \left(C_{\text{initial}} - \hat{C}_d(\boldsymbol{\theta}, t_1) \right)^2 \\ \vdots \\ \left(C_{\text{initial}} - \hat{C}_d(\boldsymbol{\theta}, t_K) \right)^2 \end{bmatrix}. \quad (3.20)$$

The regularization weights increase at time points where carbon balance errors are large, thus enforcing stronger penalties in challenging regions. Conversely, weights remain small where constraints are easily satisfied, allowing the model to prioritize data fitting. Note that each parameter update for θ and λ use their own corresponding learning rate, η_θ and η_λ .

This adaptive weighting strategy is particularly effective for complex reaction systems where the difficulty of enforcing conservation laws varies significantly over time. Previous studies have incorporated atom conservation laws either as soft constraints [43], [48]–[51] or hard constraints [39], [52], [53] while the proposed method self-adaptively adjusts the regularization weights rather than manually tuning it. Related methods have instead applied post-hoc corrections to the trained model, rather than enforcing the constraint throughout the training process [54]. While other self-adaptive regularization approaches for physics-informed neural ODEs have linearly increased the regularization weight [55] during the training process.

3.4 Validation Strategies

The validation approaches are adapted to match the available data in each study.

3.4.1 Design of Experiments Approach

To systematically explore the operating space with synthetic data for **System I**, Design of Experiments (DoE) is adopted using full factorial designs. Particularly, two set-ups are investigated:

- 2^3 full factorial design with a center point, varying initial concentrations of the reactant species and temperature on two levels (± 1),
- 3^3 full factorial design with three levels ($-1, 0, +1$) for each factor.

The experimental design described above is further elucidated in Fig. 3.2 and Table 3.1. This structured approach ensures comprehensive coverage of the operating space and captures interactions between factors. An independent test set of 1000 randomly sampled initial conditions is generated by performing a Monte Carlo simulation to evaluate generalization capabilities with each test trajectory sampled at 1000 evenly spaced time points.

Note that the 2^3 full factorial design incorporates five temporal data points per training dataset, whereas the 3^3 full factorial design utilizes 75 temporal data points per dataset.

3.4.2 Leave-One-Out Strategy

In contrast to the systematic factorial designs used with synthetic data in **System I**, the limited availability of data in real-world applications, necessitated a leave-one-out validation strategy for **System II**. The central point in the operating space serves as the validation dataset, while the remaining experimental conditions are used for training. This approach maximizes the use of limited data for training while providing a realistic assessment of model generalization for unseen conditions. The experimental design for this scenario was constructed by Arora et al. [7] and is presented in Fig. 3.3 with the complementary experiment numbers in Table 3.2.

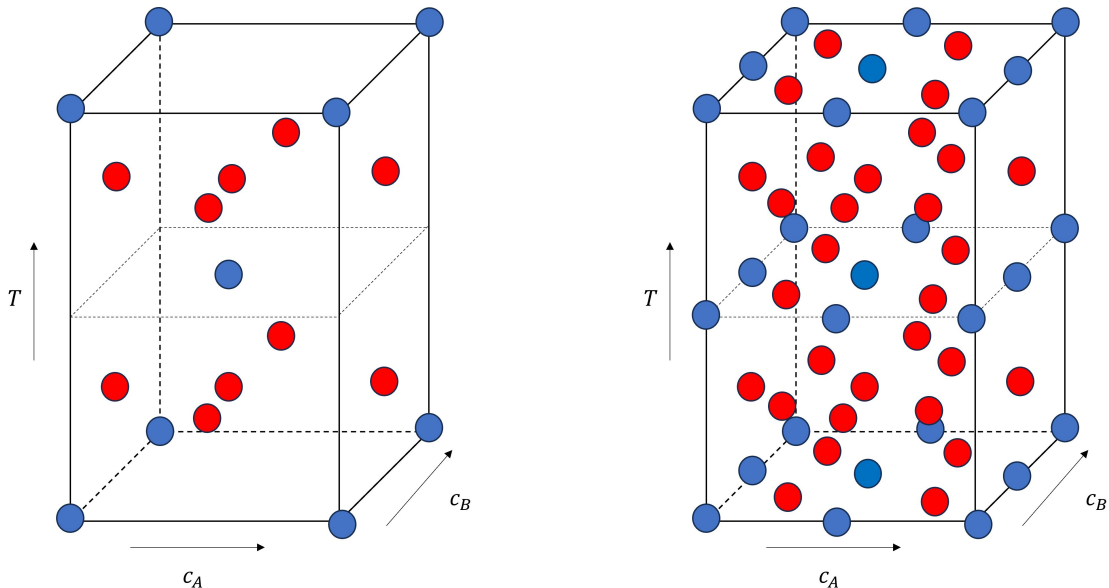
(a) 2^3 full factorial design with a center point.(b) 3^3 full factorial design with three levels.

Figure 3.2: Experimental design for 3.2(a) and 3.2(b). The blue scatter points are training datasets and the red scatter points are validation datasets. Reprinted from **Paper I**.

Table 3.1: Initial values used to generate synthetic data and their corresponding levels in the experimental design in Fig. 3.2. Reprinted from **Paper I**.

	Level	c_A [mol/L]	c_B [mol/L]	T [K]
2^3 Full Factorial Design	+1	14	7	320
	-1	6	3	290
3^3 Full Factorial Design	+1	14	7	320
	0	10	5	305
	-1	6	3	290

3.4.3 Uncertainty Estimation

Reliable uncertainty estimation is a prerequisite for utilizing neural networks in chemical process applications, where predictions are accompanied by confidence estimates. Among the various strategies for uncertainty estimation in neural networks, this work adopts deep ensembles as the primary approach [56]. Deep ensembles comprises multiple neural networks trained independently with different random initializations, aggregating the network predictions to obtain a mean prediction together with standard deviations for the model prediction. Although Bayesian neural networks (BNNs) offer a framework which obtains the distributions over the network weights [57], [58], their practical limitations such as lack of parallelization and lack of exploration of multiple loss landscapes makes deep ensembles a more suitable

Table 3.2: Experimental conditions for the different experiment numbers.

Experiment number	Temperature [°C]	Pressure [bar]	Amount of SA [wt%]
1	325	50	5
2	300	70	5
3	300	40	5
4	275	50	5
5	300	50	2
6	300	50	8
Validation	300	50	5

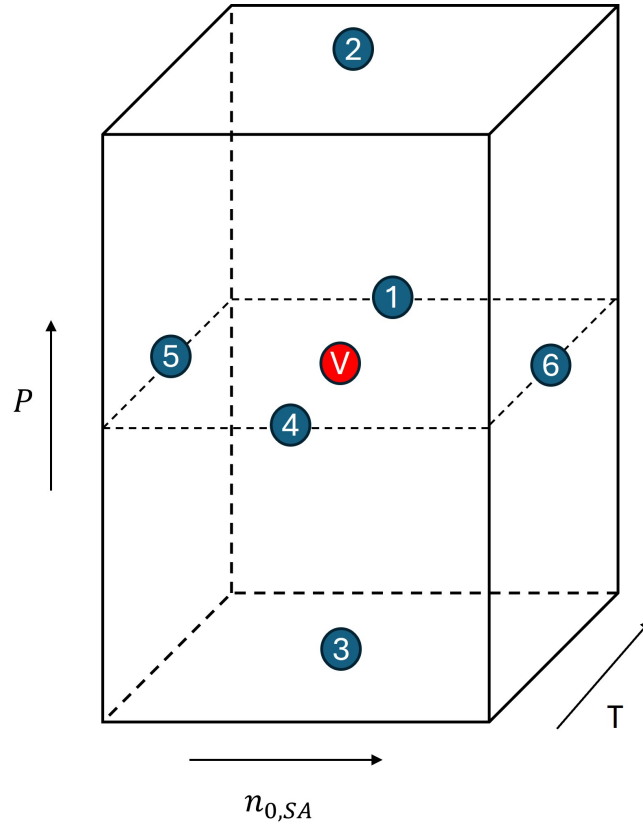


Figure 3.3: Experimental design where the blue and red scatter points represent dataset(s) used for training and validation, respectively.

choice for this application.

3.5 Software

Since neural ODEs form the core computational component of this work, requiring frameworks capable of efficient ODE solving coupled with automatic differentiation for gradient-based optimization and adjoint sensitivity methods for backpropagation remains imperative. A comparative evaluation of three prominent scientific machine learning ecosystems was conducted: PyTorch with torchdiffeq, JAX with Diffrax and Julia with Lux.jl, Zygote.jl, DiffEqFlux.jl and DifferentialEquations.jl. The comparison focused on four criteria: computational performance, automatic differentiation capabilities, ease of implementation and lastly, ecosystem maturity and community support.

From the work in **Paper I** and **II**, the experience across all three frameworks revealed distinct trade-offs. Julia demonstrated greater computational speed but exhibited some limitations such as limited automatic differentiation capabilities in certain contexts, optimizer bugs, mutation-related issues when handling multiple datasets, which required non-trivial workarounds and complex dependency management that often necessitated creating new environments when updating packages. PyTorch with torchdiffeq, conversely, proved robust and stable with no encountered bugs but suffered from considerably slower training times. JAX with Diffrax however, delivers computational performance comparable to Julia while maintaining the robustness observed in PyTorch. However, achieving peak performance in JAX requires deliberate application of just-in-time (JIT) compilation decorators and vectorization primitives (vmap), whereas Julia’s speed benefits are more readily accessible without such specialized knowledge. Additionally, JAX remains in relatively early development stages, with GPU support varying in maturity depending on available hardware infrastructure.

Table 3.3: Comparison of scientific machine learning frameworks for neural ODEs.

Framework	Speed	Robustness	Ease of use	Maturity
PyTorch + torchdiffeq	Slow	High	High	Mature
JAX + Diffrax	Fast ^a	High	Moderate ^b	Developing
Julia + Packages	Fast	Low	High	Moderate

^a Requires JIT and vmap to observe speed-up benefits.

^b Learning curve for JIT/vmap, but otherwise straightforward.

Based on these findings, Julia was selected for **Paper I** to leverage its computational advantages, while JAX was adopted for **Paper II** to benefit from both high performance and stability, accepting the additional implementation complexity as a worthwhile trade-off.

Chapter 4

Results and Discussion

4.1 Incorporation of Physics-Based Regularization

The physics-informed neural ODE approaches are investigated for the two systems presented in Section 3.1. Firstly, the component mass balances are incorporated into the loss function and evaluated for **System I**. To properly investigate the proposed model, it is benchmarked against a traditional neural ODE utilizing 20 different random initializations of the network parameters, thus resulting in 20 realizations of the trained network parameters, which are then used to obtain an estimate of the test accuracies. The incorporation of discovered formulations for mass conservation laws into the loss function demonstrates improvements in generalization capabilities. This is specifically investigated when training data is limited, since fully data-driven models commonly require large amounts of data. Employing a 2^3 full factorial design with a center point as presented in Fig. 3.2(a) and Table 3.1 with only 5 temporal observations per trajectory, the median test accuracy, R^2_{test} improved from 0.92 for the benchmark neural ODE to 0.94 for the physics-regularized model. Notably, the physics-informed neural ODE enables accurate predictions even when key dynamics are inadequately sampled through the training data. Fig. 4.1 illustrates a case where uniform temporal sampling fails to capture the rapid concentration changes that occur early in the reaction. The traditional neural ODE predicts physically infeasible negative concentrations during this phase, while the physics-informed neural ODE maintains physical consistency and provides a more accurate prediction of the concentration trajectories. This behavior can be understood through the applied constraint. By evaluating the conservation law at logarithmically-spaced collocation points, which are densely populated in the early phase of the reaction, the model is encouraged to make predictions that satisfy the mass balance relations even in regions that are sparsely populated by training data. The collocation point distribution thus serves as a form of informed sampling that guides the neural network toward physically consistent solutions.

Table 4.1 motivates the choice for the distribution of collocation points. Specifically, three different collocation sampling techniques are investigated, elucidating that the logarithmic sampling results in the best generalization capabilities, demonstrated by obtaining the highest test accuracy compared to the other collocation sampling techniques. Additionally, logarithmic sampling results in the lowest standard deviation for the model accuracy, displaying lower

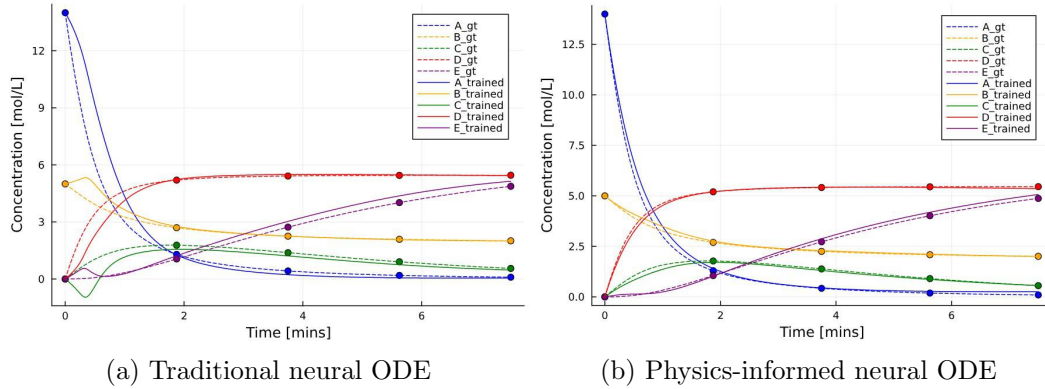


Figure 4.1: Temporal evolution of concentration for 4.1(a) traditional neural ODE and 4.1(b) physics-informed neural ODE a selected validation dataset. The dashed lines represent ground truth trajectories, the solid lines represent the prediction of the trained model and the scattered dots represent the specific time points for the observed data points. Reprinted from **Paper I**.

variance. With logarithmically-spaced collocation points, the median R_{test}^2 reached 0.94, obtaining a higher test accuracy than both uniform spacing (0.93) and Latin hypercube sampling (0.93).

Table 4.1: Test accuracy, R_{test}^2 for various collocation sampling techniques with $\lambda_{\text{mass}} = 1$ and 5 observed data points. Reprinted from **Paper I**.

Collocation sampling technique	R_{test}^2		
	Average	Median	Std Dev
Logarithmic sampling	0.93	0.94	0.04
Uniform sampling	0.91	0.93	0.06
LHS	0.92	0.93	0.06
Benchmark	0.89	0.92	0.06

Table 4.2 presents the effect of systematically increasing temporal observations from 5 to 10 points. The physics-informed neural ODE consistently demonstrates higher mean and median accuracy across all number of temporal data points, with the most pronounced improvement at the lower number of temporal observations, with the improvement fading away as number of observations increases. As data becomes more abundant, both methods converge to similar accuracy levels, indicating that the physics-informed neural ODE is most valuable for improved generalization capabilities when data availability is scarce.

Importantly, the physics-informed neural ODE exhibits consistently lower or the same standard deviation in R_{test}^2 across the various observation counts. This reduced variance indicates decreased sensitivity to parameter initialization and more stable models, requiring

Table 4.2: Test accuracy, R_{test}^2 for varying numbers of temporal observations. Adapted from **Paper I**.

Observations	Traditional neural ODE			Physics-informed neural ODE		
	Mean	Median	Std Dev	Mean	Median	Std Dev
5	0.89	0.92	0.06	0.93	0.94	0.04
6	0.94	0.95	0.03	0.95	0.95	0.03
7	0.95	0.97	0.05	0.96	0.97	0.02
8	0.95	0.97	0.07	0.97	0.97	0.01
10	0.96	0.97	0.04	0.97	0.97	0.02

less number of realizations of model parameters to obtain an accurate model. The variance reduction arises from the constraint reducing the parameter space that the network can explore, leading to more consistent convergence.

To evaluate the proposed regularization technique in a more realistic setting, absolute Gaussian noise with standard deviations of 0.01, 0.02 and 0.05 is added to the concentration measurements, with temperature noise scaled by a factor of 10 to account for typical measurement precision differences. At $\sigma = 0.01$, both methods demonstrate better generalization capabilities than with non-corrupted data as shown in Table 4.3. The physics-informed neural ODE obtains a mean R_{test}^2 of 0.94 with noisy data compared to 0.93 with non-corrupted data and the traditional neural ODE shows a median R_{test}^2 of 0.93 with noisy data compared to 0.92 with non-corrupted data. The small improvement in generalization capabilities is likely a result of implicit regularization from noise, preventing overfitting to the precise training trajectories. As noise levels increase, both methods degrade however the physics-informed neural ODE maintains better generalization capabilities as evidenced by Table 4.3. For instance, at $\sigma = 0.05$, the physics-informed neural ODE demonstrates a mean R_{test}^2 of 0.85 compared to 0.81 for the traditional neural ODE.

Table 4.3: Test accuracy, R_{test}^2 under varying absolute Gaussian noise levels. Adapted from **Paper I**.

Noise σ	Traditional neural ODE			Physics-informed neural ODE		
	Mean	Median	Std Dev	Mean	Median	Std Dev
0.01	0.89	0.93	0.09	0.94	0.94	0.02
0.02	0.89	0.92	0.07	0.93	0.92	0.04
0.05	0.81	0.84	0.11	0.85	0.87	0.09

Real-world measurements of concentration through gas chromatography and temperature measurements obtained via thermocouples are often corrupted by noise. To emulate this, two noise components are introduced to the synthetic data: (i) a relative Gaussian error (1–5% relative standard deviation) to emulate the precision of gas chromatography and (ii) absolute homoscedastic Gaussian noise (up to 0.5 °C) added to the temperature data. As

shown in Table 4.4, at $\sigma = 0.05$, the physics-informed approach obtains a median R_{test}^2 of 0.85 compared to 0.83 for the traditional neural ODE, despite both methods showing increased standard deviations at higher noise levels, the consistent performance advantage across both absolute and relative noise conditions demonstrates the robustness of the physics-informed regularization approach. Table 4.5 shows that increasing the number of temporal data points despite higher levels of noise, the improved generalization capabilities of physics-informed neural ODEs persist.

Table 4.4: Test accuracy under heteroscedastic noise. Temperature measurements use absolute Gaussian noise, while concentration measurements use relative noise proportional to the true concentration value. Adapted from **Paper I**.

Method	Noise Parameter (σ)	Mean	Median	Std Dev
	Temp: $\sigma_T = 10\sigma$, Conc: $\sigma_C = \sigma \times c_i$			
Physics-informed neural ODE	0.01	0.92	0.94	0.04
	0.02	0.91	0.93	0.05
	0.05	0.79	0.85	0.14
Traditional neural ODE	0.01	0.90	0.91	0.04
	0.02	0.89	0.91	0.07
	0.05	0.80	0.83	0.15

To evaluate the physics-informed approach for a real-world problem, experimental data for HDO of stearic acid [7] is employed rather than the synthetic model system in **System I**. Particularly, a different form of physics-informed regularization is adopted, which requires *a priori* information about the model system. Specifically, an atom balance is used as a soft constraint as shown in Eqs. 3.11 and 3.18 together with the experimental design in Fig. 3.3 is employed.

A fundamental question in physics-informed approaches lies in finding the balance between the data-driven and mechanistic model components. For instance, in Eq. 3.4, this is determined by the value of the regularization weight, λ_{physics} . Commonly, an appropriate value for the regularization weight is determined through trial and error via a grid search, which can be computationally demanding and potentially lead to suboptimal results. This can be circumvented by incorporating the regularization weights into the training procedure as explained in Section 3.3.2. Notably, a vector of regularization weights are incorporated into the training procedure, thus applying a unique regularization weight to each collocation point. This approach is benchmarked against the kinetic model developed for the HDO of stearic acid by Arora et al. [7]. Following the results in Table 4.1, a logarithmic distribution for the collocation points is employed. It is shown that the model accuracy is improved by employing an atom balance as a soft constraint together with a self-adaptive regularization approach as evidenced by Table 4.6, where 100 predictions are utilized, employing deep ensembles as a form of uncertainty quantification. Specifically, the improvement for the physics-informed neural ODE is observed both for each individual species and in aggregate. Note that the central point in Fig. 3.3 is adopted as a training dataset for the kinetic model and as a

Table 4.5: Test accuracy for varying numbers of observed data points under heteroscedastic noise with $\sigma = 0.05$. Temperature measurements use absolute Gaussian noise, while concentration measurements use relative noise proportional to the true concentration value. Adapted from **Paper I**.

Method	Number of observed data points	Mean	Median	Std Dev
	Temp: $\sigma_T = 10\sigma$ Conc: $\sigma_C = \sigma \times c_i$			
Physics-informed neural ODE	5	0.79	0.85	0.14
	6	0.87	0.91	0.10
	7	0.90	0.92	0.06
	8	0.91	0.93	0.05
	10	0.92	0.94	0.08
Traditional neural ODE	5	0.80	0.83	0.15
	6	0.84	0.90	0.13
	7	0.84	0.89	0.13
	8	0.87	0.92	0.13
	10	0.89	0.92	0.08

validation dataset for the physics-informed neural ODE, thus the central point is an unseen dataset for the physics-informed neural ODE. The most substantial improvement is observed for heptadecane and its isomers, where the predictive accuracy, R^2 increases from 0.67 in the kinetic model to 0.88 for the ensemble mean predictions. A similar improvement in predictive accuracy is evident for stearic acid, yielding an R^2 of 0.97. Even for species where the kinetic model attains its highest predictive performance, the physics-informed approach consistently obtains a higher predictive accuracy, as demonstrated by the improved prediction accuracy for C18+.

Table 4.6: Prediction accuracy, R^2 for the mean and median ensemble predictions for the center point in the experimental design for stearic acid. The prediction accuracy of the kinetic model is also presented as a benchmark.

Model	SA	C17+	C18+	Total
Kinetic Model	0.81	0.67	0.94	0.80
Physics-informed neural ODE (Mean)	0.97	0.88	0.97	0.94
Physics-informed neural ODE (Median)	0.97	0.86	0.98	0.94

Since the regularization weights are automatically updated during training for each given collocation point, it is of interest to study the regularization weight profile as presented in Fig. 4.2. Particularly, the magnitude of the regularization weights provide insights into the specific

temporal regions where enforcing mass balance is more challenging. The figure can be divided into four distinct phases, where in the initial phase, there is a sharp increase during the first 10 minutes of the reaction. The second phase consists of a plateau between the reaction times 20-65 mins, which is followed up by a gradual decline between the reaction times 65-155 mins and lastly, another sharp increase towards the end of the reaction. The sharp initial increase can likely be explained by the fast transients in the species concentrations.

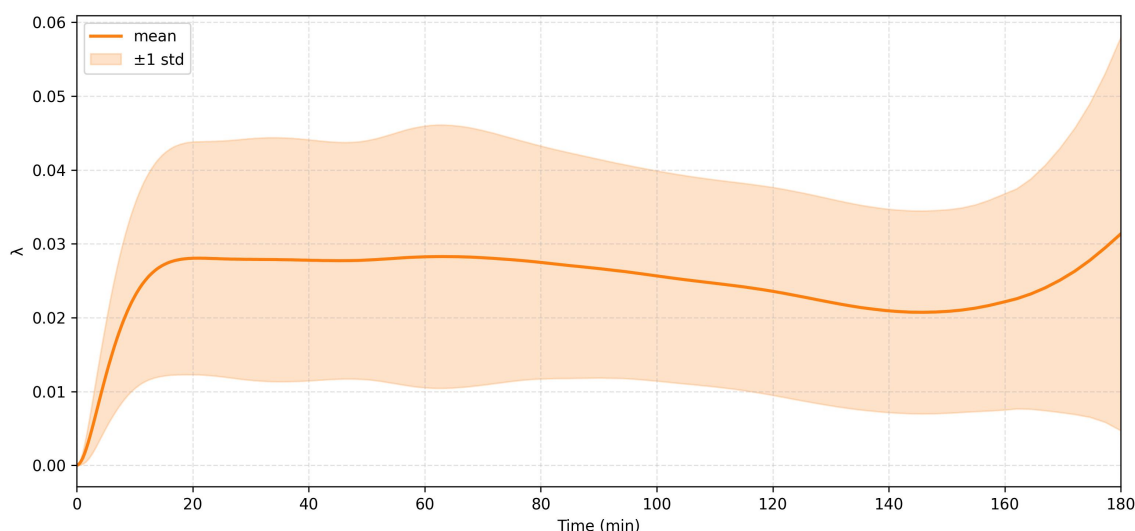
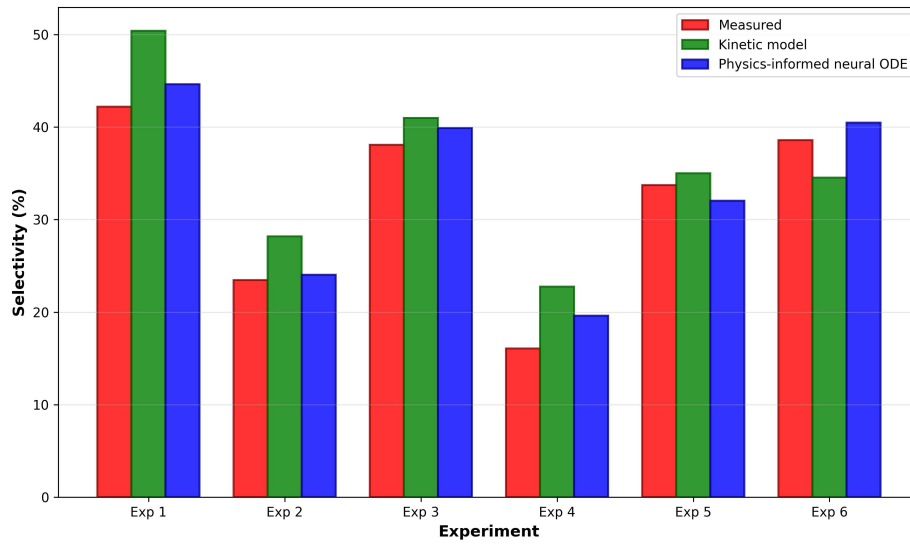
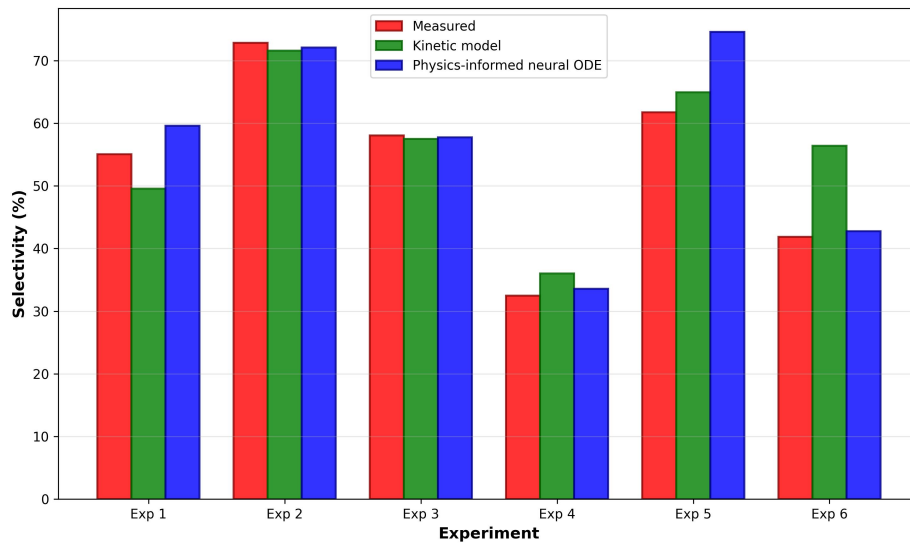


Figure 4.2: Evolution of the learnable regularization weight, λ as a function of time. The solid line represents the mean trajectory across the ensembles, while the shaded region indicates $\pm\sigma$.

Accurate selectivity prediction is essential for model prediction in chemical reactor engineering. Fig. 4.3 compares the selectivity prediction of the physics-informed neural ODE and the kinetic model to the measured selectivity. As shown in the figure, the physics-informed neural ODE typically provides more accurate selectivity predictions in aggregate than the kinetic model for experiments 1-4 and 6. Nevertheless, in experiment 5, the kinetic model demonstrates a better predictive performance. The deviation is primarily attributed to the increased epistemic uncertainty associated with the physics-informed neural ODE for that specific experiment. On average, the physics-informed model improves selectivity predictions by 2.7 percentage points for C17+ and 1.4 percentage points for C18+.



(a) C17+



(b) C18+

Figure 4.3: Selectivity predictions for 4.3(a) C17+ and 4.3(b) C18+. The red bars represent the measured selectivity from experiments, the green bar represents the selectivity prediction for the kinetic model and the blue bar is the selectivity prediction of C18+.

4.2 Pre-training

Digitalization in chemical engineering has led to the widespread adoption of online sensors such as near-infrared (NIR) spectroscopy, which provide concentration measurements with high temporal resolution. This capability enables the acquisition of large datasets suitable for data-driven modeling approaches such as techniques that are derived from deep learning.

In scenarios where abundant data is available, generalization is less challenging due to comprehensive coverage of the operating space, therefore computational efficiency becomes the primary concern. Since neural ODEs use a neural network to learn a representation of the kinetics, it is desirable to train the model on the input-output pair; $(y_i, \frac{dy_i}{dt})$. This methodology would thus circumvent the use of an ODE solver and conveniently skip utilizing the adjoint method for backpropagating through the ODE solver. Particularly, the task would be transformed to a traditional supervised learning problem. The primary challenge with this approach is that online sensors do not measure the state derivatives, instead they measure the state variables. As described in Section 3.3.1, it has been proposed to train the neural network on the state derivatives by estimating the numerical derivatives of the training data through the finite differencing scheme. This approach serves as a pre-training step, followed by standard training of neural ODEs through direct minimization of the MSE with respect to the state variables. By initializing the network with parameters obtained from pre-training, rather than using random schemes such as Glorot or He initialization, the subsequent training process benefits from more favorable starting conditions.

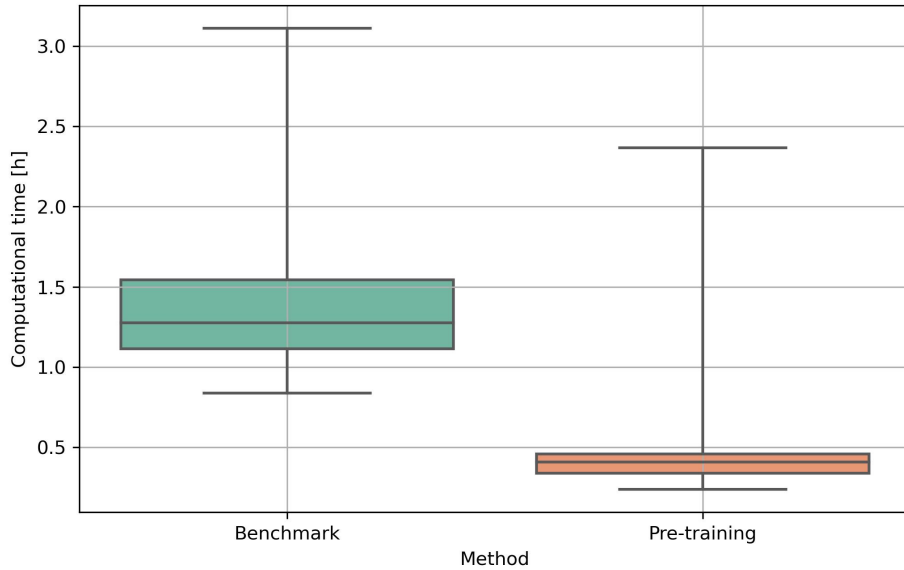


Figure 4.4: Computational times for the benchmark which considers a traditional neural ODE and when incorporating pre-training into the training procedure. Reprinted from **Paper I**.

The 3^3 full factorial design presented in Fig. 3.2(b) is adopted to demonstrate the effect of pre-training, with 75 uniformly spaced temporal observations per trajectory. As Fig. 4.4 elucidates, pre-training reduces median computational time by 68%. This efficiency gain comes without compromising accuracy. All realizations in Fig. 4.4 for both pre-training and benchmark methods demonstrate strong generalization capabilities, with $R^2_{\text{test}} > 0.985$. The distribution of computational times also exhibits reduced variance with pre-training since the inter-quartile range is smaller than the inter-quartile range of the benchmark, indicating more predictable training times. The computational advantage of pre-training can be understood through the structure of the optimization landscape. By training on the densely sampled state

derivatives, the neural network is provided with a prior structural bias towards the underlying target dynamics. This effectively pre-shapes the neural network to approximate the system dynamics. When main training begins, the network parameters are already in a region where the predicted dynamics are approximately correct. The expensive ODE integration is finally incorporated to refine the model predictions. Fig 4.5 displays the prediction for a realization that utilizes pre-training, illustrating a high predictive accuracy with significantly decreased training times.

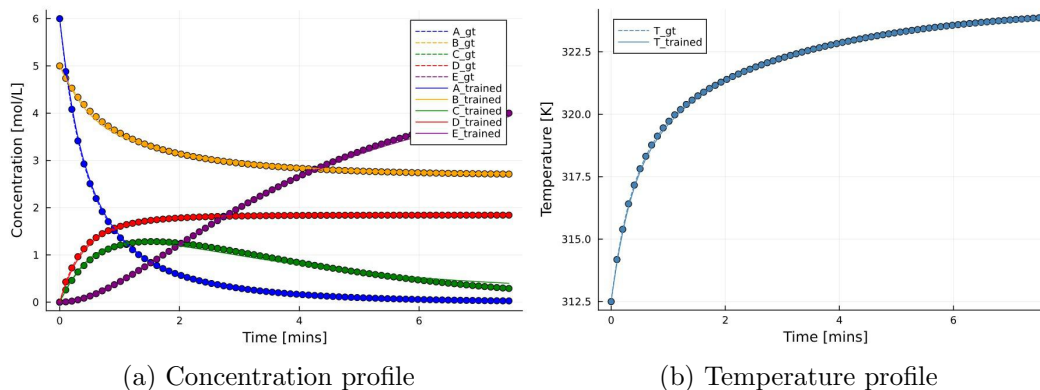


Figure 4.5: Neural ODE predictions following pre-training for 4.5(a) concentration and 4.5(b) temperature profiles. Solid lines represent model predictions, dashed lines are ground truth trajectories and scatter points are the observed data points. Reprinted from **Paper I**.

4.3 Uncertainty Estimation

Model uncertainty is quantified by epistemic uncertainty which stems from lack of data and knowledge about the underlying system. As explained in Section 3.4.3, epistemic uncertainty can be modeled using deep ensembles. This is illustrated in Figs. 4.6-4.8. Specifically, Fig. 4.6 illustrates the prediction of the traditional neural ODE, which has not incorporated any mechanistic priors. As the figure shows, there is a large amount of epistemic uncertainty which is observed by the wide uncertainty bands. Additionally, the mean ensemble prediction makes largely inaccurate predictions after $t = 70$ mins for all the chemical species shown in Fig. 4.6. Incorporation of mechanistic constraints such as an atom balance lowers the epistemic model uncertainty as demonstrated in Fig. 4.7, with tighter uncertainty bands and more accurate prediction by the mean of the ensembles.

Fig. 4.7 shows that the mean of the ensembles introduces a systematic error when predicting the steady-state value of the product distributions for stearic acid and octadecane and its corresponding isomers for the physics-informed neural ODE, while the kinetic model predicts the steady-state value for the product distributions more accurately. Nevertheless, the center point in the experimental design in Fig. 3.3 is used for validation using a leave-one-out strategy for the physics-informed neural ODE, meanwhile the kinetic model adopts the center point as a training dataset. Therefore the inability to capture the steady-state values for the product distributions is not due to a lack of capability but rather that the physics-informed

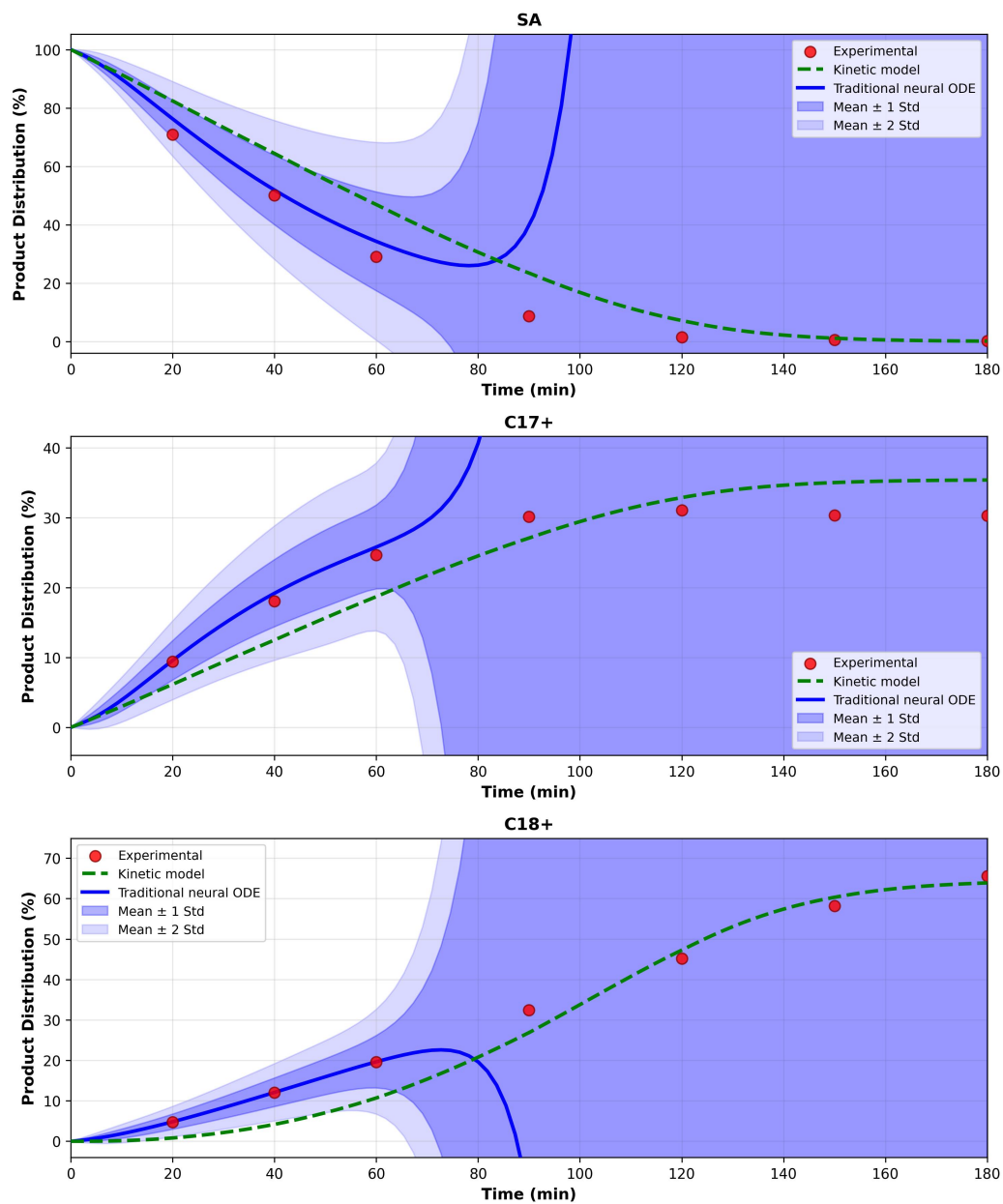


Figure 4.6: Ensemble-based uncertainty quantification for traditional neural ODE predictions without mechanistic priors predicting product distributions for the center point in the experimental design in Fig. 3.3 for temporal product distributions for SA, C17+ and C18+ species. Red circles represent experimental measurements. Solid blue lines show ensemble mean predictions from deep ensembles. Green dashed lines show predictions from the mechanistic kinetic model for comparison.

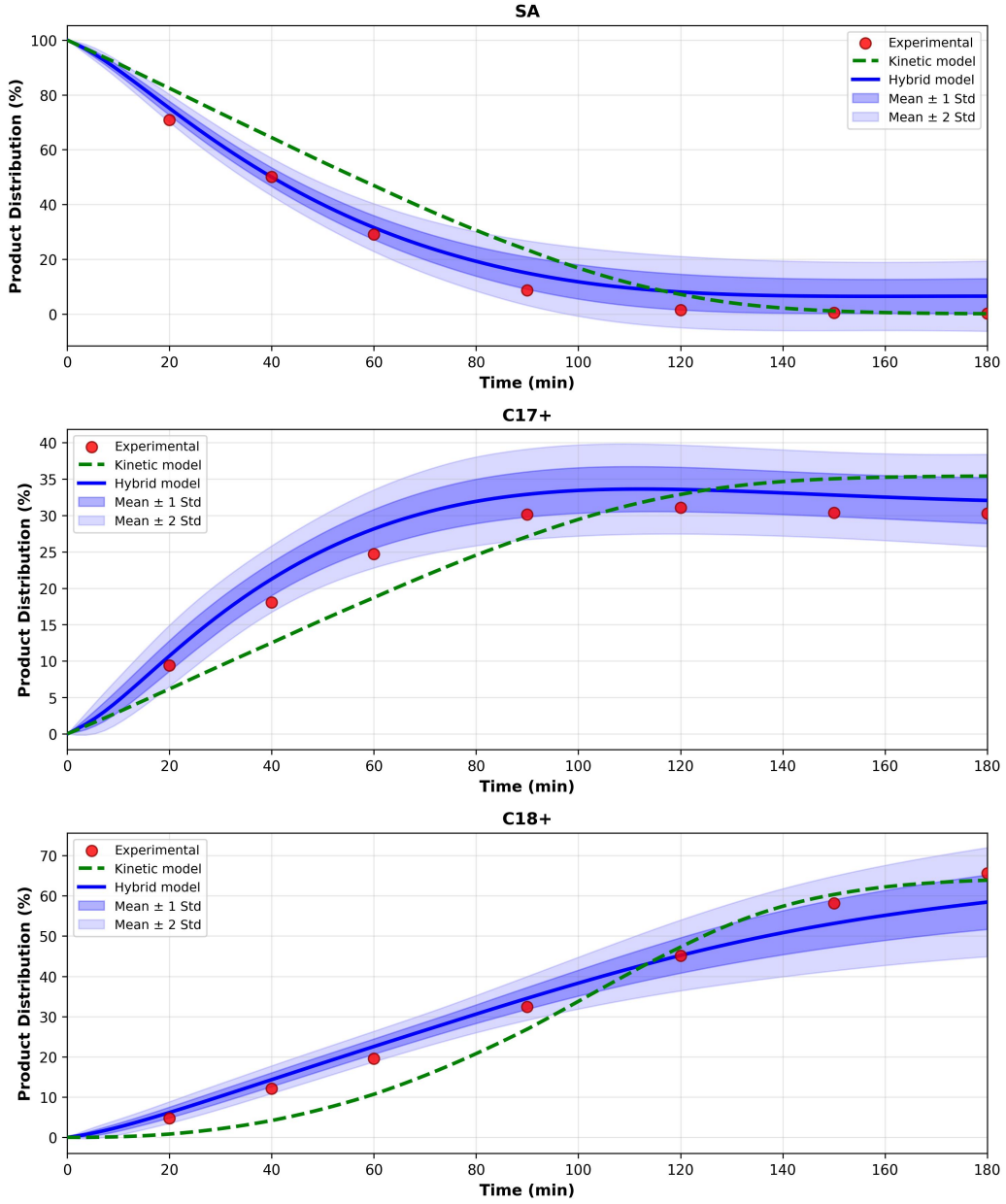


Figure 4.7: Ensemble-based uncertainty quantification for physics-informed neural ODE models predicting product distributions for the center point in the experimental design shown in Fig. 3.3 for temporal product distributions for SA, C17+ and C18+ species. Red circles represent experimental measurements. Solid blue lines show ensemble mean predictions from deep ensembles. Shaded regions indicate model uncertainty. Green dashed lines show predictions from the mechanistic kinetic model for comparison. Reprinted from **Paper I**.

model has not incorporated this dataset into its training procedure, in addition to a lack of variation in the datasets in the design space in Fig. 3.3. This is further verified by Fig. 4.8, which illustrates the prediction of the physics-informed model for a training dataset, specifically for Experiment 1 in Table 3.2. The figure shows that the physics-informed model demonstrates a more accurate prediction of the steady-state product distribution values. Unsurprisingly, the uncertainty bands of the ensemble-based predictions are also tighter for the training dataset compared to the validation dataset. Nevertheless, mechanistic kinetic models have a built-in inductive bias, which forces the kinetic model to predict a monotonically decreasing product distribution for starting products and a monotonically increasing product distribution for end-products. To inform or guarantee such behavior in the physics-informed model, this would need to be enforced through constraints or incorporation of more experimental data to induce such behavior.

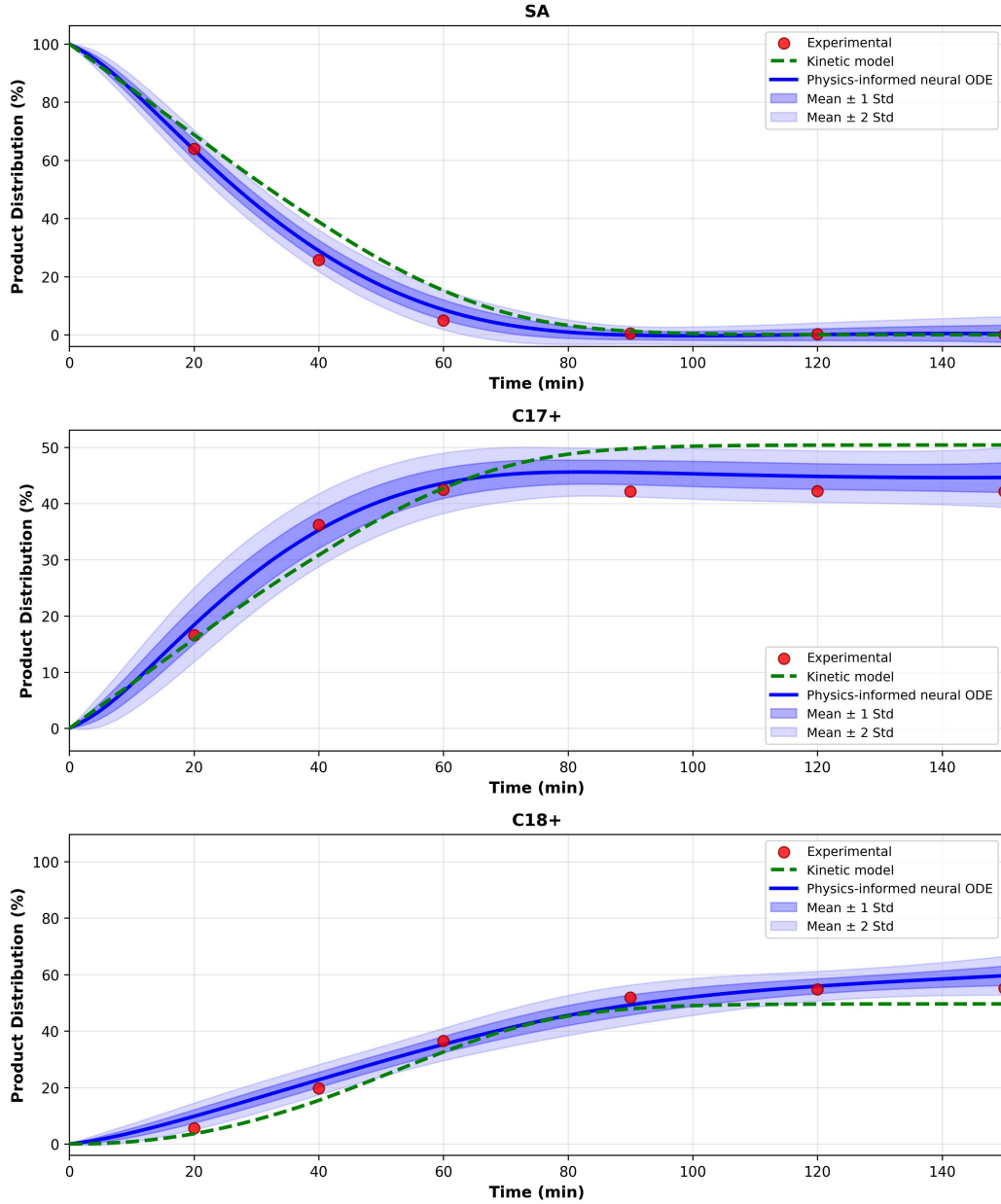


Figure 4.8: Ensemble-based uncertainty quantification for physics-informed neural ODE models predicting product distributions for the high-temperature corner point in the experimental design shown in Fig. 3.3 for temporal product distributions for SA, C17+ and C18+ species. Red circles represent experimental measurements. Solid blue lines show ensemble mean predictions from deep ensembles. Shaded regions indicate model uncertainty. Green dashed lines show predictions from the mechanistic kinetic model for comparison.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

This thesis has demonstrated that integrating mechanistic knowledge in the form of soft constraints into the neural ODE framework enhances their generalization capabilities and reduces their epistemic uncertainties. Particularly, the findings reveal that by constructing a physics-informed model through the incorporation of conservation laws in the form of component mass balances, the model not only improves the generalization capabilities but also shows improved robustness compared to a traditional neural ODE. It has been shown that scarce training data does not hinder the physics-informed model from displaying strong generalization capabilities. Nevertheless, as the amount of training data increases, the performance gap between the physics-informed model and the traditional neural ODE narrows. However, the enhanced robustness of the physics-informed model remains consistently evident. This has been further evidenced by other types of conservation laws as soft constraints, specifically the incorporation of an atom balance into the model has been shown to lower the epistemic uncertainty. In addition to the performance gap between the physics-informed neural ODE and the traditional neural ODE, the physics-informed model demonstrates higher model accuracy compared to the mechanistic kinetic model. Such improvement highlights the benefit of embedding domain knowledge into data-driven models, particularly when mechanistic models struggle with parameter uncertainty or incomplete kinetics. While the proposed methodologies show promise, certain limitations remain such as the adoption of soft constraints solely *encourage* the model to explore a specific region in the parameter space. Thus, this does not guarantee that the model strictly adheres to the applied constraints since they are not enforced as hard constraints. Enforcing hard constraints ensures and guarantees adherence to the applied constraints. Hard constraints can incorporate certain known model structures regarding the rate law such as the stoichiometric matrix, Arrhenius temperature dependence or the monotonicity of the concentration of end-products.

Furthermore, incorporation of pre-training by utilizing the estimated numerical derivatives of the state variables has shown to be an efficient training strategy. Specifically, this approach reduces total training time by 68%, effectively integrating scientific machine learning techniques to lower computational costs. This improvement is particularly valuable in scenarios where large amounts of temporal data is acquired through sensors with high

temporal resolution, enabling the practical use of scientific machine learning algorithms with large datasets collected by online sensors.

In conclusion, this work establishes a foundation for physics-informed neural ODEs in chemical engineering applications within chemical reaction engineering.

5.2 Future Work

The work presented in this thesis consists of paths for future investigations, building upon the foundations already established. For example, this thesis has explored the discovery of linear relations, subsequently incorporated as soft constraints within the optimization framework. In particular, discovered formulations of component mass balances have been embedded within the objective function. Given that these formulations are merely linear combinations of underlying component mass balances, an important extension is to identify the exact linear relations themselves. This can be achieved through integer relation algorithms such as PSLQ [59] or the HJLS algorithm [60]. Building on the contributions of **Paper I**, an additional direction, discovering non-linear relations between the state variables is also of interest. Sparse identification of non-linear dynamics (SINDy) [61] can be employed by constructing a library of suitable candidate basis functions and discovering which sparse combination remain invariant. While Kaiser et al. [62] have investigated this approach, its integration as a physics-informed regularization for neural ODEs remains unexplored.

Further future studies include integrating a neural network to the already implemented kinetic model by Arora et al. [7], where the objective of the neural network is to approximate the missing term in the mechanistic model. Additional extensions could treat the kinetic parameters as trainable parameters, enabling their estimation through stochastic gradient descent rather than traditional non-linear parameter regression. This approach thus preserves model interpretability by retaining the structural form of Langmuir-Hinshelwood kinetics.

Model interpretability remains critical for the widespread adoption of scientific machine learning models. Without proper model interpretability, even the most advanced algorithms and methods in scientific machine learning risk remaining theoretical rather than practical. Recent developments have demonstrated the promise of interpretable model architectures such as the Chemical Reaction Neural Network (CRNN) [37], [38], Hamiltonian Neural Network (HNN) [63] and most recently, Kolmogorov-Arnold Network (KAN) [64], [65]. These advances support the necessary trajectory of scientific machine learning, integrating predictive power with trustworthiness and explainable models.

Bibliography

- [1] N. Baker, F. Alexander, T. Bremer *et al.*, “Workshop report on basic research needs for scientific machine learning: Core technologies for artificial intelligence,” 2019. DOI: 10.2172/1478744 (cit. on p. 3).
- [2] H. S. Fogler, *Elements of Chemical Reaction Engineering*, 5th. Prentice Hall, 2016, ISBN: 978-0-13-3887 (cit. on p. 3).
- [3] V. Venkatasubramanian, “The promise of artificial intelligence in chemical engineering: Is it here, finally?” *AIChE Journal*, vol. 65, no. 1, 2019 (cit. on p. 3).
- [4] B. Donnis, R. G. Egeberg, P. Blom and K. G. Knudsen, “Hydroprocessing of bio-oils and oxygenates to hydrocarbons. understanding the reaction routes,” *Topics in Catalysis*, vol. 52, no. 3, pp. 229–240, 2009. DOI: <https://doi.org/10.1007/s11244-008-9159-z> (cit. on p. 4).
- [5] D. M. Bates and D. G. Watts, *Nonlinear regression analysis and its applications*. Wiley New York, 1988, vol. 2 (cit. on p. 4).
- [6] P. Englezos and N. Kalogerakis, *Applied parameter estimation for chemical engineers*. CRC Press, 2000 (cit. on p. 4).
- [7] P. Arora, E. L. Grennfelt, L. Olsson and D. Creaser, “Kinetic study of hydrodeoxygenation of stearic acid as model compound for renewable oils,” *Chemical Engineering Journal*, vol. 364, pp. 376–389, 2019, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2019.01.134> (cit. on pp. 4, 10, 14, 22, 34).
- [8] M. Snåre, I. Kubičková, P. Mäki-Arvela, K. Eränen, J. Wärnå and D. Murzin, “Production of diesel fuel from renewable feeds: Kinetics of ethyl stearate decarboxylation,” *Chemical Engineering Journal*, vol. 134, no. 1, pp. 29–34, 2007, Proceedings of the XVII International Conference on Chemical Reactors CHEMREACTOR-17 and Post-Symposium ”Catalytic Processing of Renewable Sources: Fuel, Energy, Chemicals”, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2007.03.064> (cit. on p. 4).
- [9] K. Jenišťová, I. Hachemi, P. Mäki-Arvela *et al.*, “Hydrodeoxygenation of stearic acid and tall oil fatty acids over ni-alumina catalysts: Influence of reaction parameters and kinetic modelling,” *Chemical Engineering Journal*, vol. 316, pp. 401–409, 2017, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2017.01.117> (cit. on p. 4).
- [10] I. Hachemi and D. Y. Murzin, “Kinetic modeling of fatty acid methyl esters and triglycerides hydrodeoxygenation over nickel and palladium catalysts,” *Chemical Engineering Journal*, vol. 334, pp. 2201–2207, 2018, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2017.11.153> (cit. on p. 4).
- [11] D. Kubička and L. Kaluža, “Deoxygenation of vegetable oils over sulfided ni, mo and nimo catalysts,” *Applied Catalysis A: General*, vol. 372, no. 2, pp. 199–208, 2010, ISSN: 0926-860X. DOI: <https://doi.org/10.1016/j.apcata.2009.10.034> (cit. on p. 4).
- [12] T. Holderbaum and J. Gmehling, “Psrk: A group contribution equation of state based on unifac,” *Fluid Phase Equilibria*, vol. 70, no. 2, pp. 251–265, 1991, ISSN: 0378-3812. DOI: [https://doi.org/10.1016/0378-3812\(91\)85038-V](https://doi.org/10.1016/0378-3812(91)85038-V) (cit. on p. 4).

-
- [13] B. Andersson, R. Andersson, L. Håkansson, M. Mortensen, R. Sudiyo and B. Van Wachem, *Computational fluid dynamics for engineers*. Cambridge university press, 2011 (cit. on p. 4).
 - [14] V. V. Ranade, *Computational flow modeling for chemical reactor engineering*. Elsevier, 2001, vol. 5 (cit. on p. 4).
 - [15] H. A. Jakobsen, “Chemical reactor modeling,” *Multiphase Reactive Flows*, 2008 (cit. on p. 4).
 - [16] G. Cybenko, “Approximation by superpositions of a sigmoidal function. math cont sig syst (mcss) 2:303-314,” *Mathematics of Control, Signals, and Systems*, vol. 2, pp. 303–314, Dec. 1989. DOI: [10.1007/BF02551274](https://doi.org/10.1007/BF02551274) (cit. on p. 5).
 - [17] K. Hornik, M. Stinchcombe and H. White, “Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks,” *Neural Networks*, vol. 3, no. 5, pp. 551–560, 1990, ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(90\)90005-6](https://doi.org/10.1016/0893-6080(90)90005-6) (cit. on p. 5).
 - [18] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991, ISSN: 0893-6080. DOI: [https://doi.org/10.1016/0893-6080\(91\)90009-T](https://doi.org/10.1016/0893-6080(91)90009-T) (cit. on p. 5).
 - [19] M. Leshno, V. Y. Lin, A. Pinkus and S. Schocken, “Multilayer feedforward networks with a nonpolynomial activation function can approximate any function,” *Neural Networks*, vol. 6, no. 6, pp. 861–867, 1993, ISSN: 0893-6080. DOI: [https://doi.org/10.1016/S0893-6080\(05\)80131-5](https://doi.org/10.1016/S0893-6080(05)80131-5) (cit. on p. 5).
 - [20] M. L. Thompson and M. A. Kramer, “Modeling chemical processes using prior knowledge and neural networks,” en, *AIChE Journal*, vol. 40, no. 8, pp. 1328–1340, 1994, ISSN: 1547-5905. DOI: [10.1002/aic.690400806](https://doi.org/10.1002/aic.690400806) (cit. on p. 5).
 - [21] D. C. Psychogios and L. H. Ungar, “A hybrid neural network-first principles approach to process modeling,” en, *AIChE Journal*, vol. 38, no. 10, pp. 1499–1511, 1992, ISSN: 1547-5905. DOI: [10.1002/aic.690381003](https://doi.org/10.1002/aic.690381003) (cit. on p. 5).
 - [22] R. T. Q. Chen, Y. Rubanova, J. Bettencourt and D. Duvenaud, “Neural ordinary differential equations,” in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS’18, Montréal, Canada: Curran Associates Inc., 2018, 6572–6583 (cit. on p. 5).
 - [23] C. Rackauckas, Y. Ma, J. Martensen *et al.*, *Universal differential equations for scientific machine learning*, 2021. DOI: <https://doi.org/10.48550/arXiv.2001.04385> (cit. on p. 5).
 - [24] M. Raissi, P. Perdikaris and G. Karniadakis, “Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations,” *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019, ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2018.10.045> (cit. on pp. 5, 7).

-
- [25] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang and L. Yang, “Physics-informed machine learning,” *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021. DOI: <https://doi.org/10.1038/s42254-021-00314-5> (cit. on p. 5).
- [26] A. G. Baydin, B. A. Pearlmutter, A. A. Radul and J. M. Siskind, “Automatic differentiation in machine learning: A survey,” *Journal of Machine Learning Research*, vol. 18, no. 153, pp. 1–43, 2018 (cit. on p. 7).
- [27] Y. Ma, V. Dixit, M. J. Innes, X. Guo and C. Rackauckas, “A comparison of automatic differentiation and continuous sensitivity analysis for derivatives of differential equation solutions,” in *2021 IEEE High Performance Extreme Computing Conference (HPEC)*, IEEE, 2021, pp. 1–9. DOI: 10.1109/HPEC49654.2021.9622796 (cit. on p. 7).
- [28] M. Raissi, A. Yazdani and G. E. Karniadakis, “Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations,” *Science*, vol. 367, no. 6481, pp. 1026–1030, 2020. DOI: 10.1126/science.aaw4741 (cit. on p. 7).
- [29] W. Ji, W. Qiu, Z. Shi, S. Pan and S. Deng, “Stiff-pinn: Physics-informed neural network for stiff chemical kinetics,” *The Journal of Physical Chemistry A*, vol. 125, no. 36, pp. 8098–8106, 2021. DOI: 10.1021/acs.jpca.1c05102 (cit. on p. 7).
- [30] M. Raissi, “Deep hidden physics models: Deep learning of nonlinear partial differential equations,” *Journal of Machine Learning Research*, vol. 19, no. 25, pp. 1–24, 2018 (cit. on p. 7).
- [31] A. Yazdani, L. Lu, M. Raissi and G. E. Karniadakis, “Systems biology informed deep learning for inferring parameters and hidden dynamics,” *PLoS computational biology*, vol. 16, no. 11, e1007575, 2020. DOI: <https://doi.org/10.1371/journal.pcbi.1007575> (cit. on p. 7).
- [32] L. S. Pontryagin, E. Mishchenko, V. Boltyanskii and R. Gamkrelidze, *The Mathematical Theory of Optimal Processes*. Wiley-Interscience, 1962 (cit. on p. 7).
- [33] O. Owoyele and P. Pal, “Chemnode: A neural ordinary differential equations framework for efficient chemical kinetic solvers,” *Energy and AI*, vol. 7, p. 100118, 2022, ISSN: 2666-5468. DOI: <https://doi.org/10.1016/j.egyai.2021.100118> (cit. on p. 8).
- [34] F. Sorourifar, Y. Peng, I. Castillo, L. Bui, J. Venegas and J. Paulson, “Physics-enhanced neural ordinary differential equations: Application to industrial chemical reaction systems,” *Industrial & Engineering Chemistry Research*, vol. 62, 2023. DOI: 10.1021/acs.iecr.3c01471 (cit. on p. 8).
- [35] A. Fedorov, A. Perechodjuk and D. Linke, “Kinetics-constrained neural ordinary differential equations: Artificial neural network models tailored for small data to boost kinetic model development,” *Chemical Engineering Journal*, vol. 477, p. 146869, 2023, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2023.146869> (cit. on p. 8).
- [36] T. Kircher, F. A. Döppel and M. Votsmeier, “Global reaction neural networks with embedded stoichiometry and thermodynamics for learning kinetics from reactor data,” *Chemical Engineering Journal*, vol. 485, p. 149863, 2024, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2024.149863> (cit. on p. 8).

-
- [37] W. Ji and S. Deng, "Autonomous discovery of unknown reaction pathways from data by chemical reaction neural network," *The Journal of Physical Chemistry A*, vol. 125, no. 4, pp. 1082–1092, 2021, PMID: 33471526. DOI: [10.1021/acs.jpca.0c09316](https://doi.org/10.1021/acs.jpca.0c09316) (cit. on pp. 8, 34).
- [38] W. Ji, F. Richter, M. J. Gollner and S. Deng, "Autonomous kinetic modeling of biomass pyrolysis using chemical reaction neural networks," *Combustion and Flame*, vol. 240, p. 111992, 2022, ISSN: 0010-2180. DOI: <https://doi.org/10.1016/j.combustflame.2022.111992> (cit. on pp. 8, 34).
- [39] F. A. Döppel and M. Votsmeier, "Robust mechanism discovery with atom conserving chemical reaction neural networks," *Proceedings of the Combustion Institute*, vol. 40, no. 1-4, p. 105507, 2024. DOI: <https://doi.org/10.1016/j.proci.2024.105507> (cit. on pp. 8, 11, 14).
- [40] M. Bangi, K. Kao and J. Kwon, "Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for β -carotene production using *saccharomyces cerevisiae*," *Chemical Engineering Research and Design*, vol. 179, Feb. 2022. DOI: [10.1016/j.cherd.2022.01.041](https://doi.org/10.1016/j.cherd.2022.01.041) (cit. on p. 8).
- [41] V. V. Santana, E. Costa, C. M. Rebello, A. M. Ribeiro, C. Rackauckas and I. B. Nogueira, "Efficient hybrid modeling and sorption model discovery for non-linear advection-diffusion-sorption systems: A systematic scientific machine learning approach," *Chemical Engineering Science*, vol. 282, p. 119223, 2023, ISSN: 0009-2509. DOI: <https://doi.org/10.1016/j.ces.2023.119223> (cit. on p. 8).
- [42] B. Klumpers, T. Luijten, S. Gerritse, E. Hensen and I. Filot, "Direct coupling of microkinetic and reactor models using neural networks," *Chemical Engineering Journal*, vol. 475, p. 145538, 2023, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2023.145538> (cit. on p. 11).
- [43] T. Kumar, A. Kumar and P. Pal, "A physics-constrained neuralode approach for robust learning of stiff chemical kinetics," in *NeurIPS Machine Learning and the Physical Sciences Workshop*, 2023 (cit. on pp. 11, 14).
- [44] D. Rehman and J. H. Lienhard, "Physics-informed deep learning for multi-species membrane separations," *Chemical Engineering Journal*, vol. 485, p. 149806, 2024, ISSN: 1385-8947. DOI: <https://doi.org/10.1016/j.cej.2024.149806> (cit. on p. 11).
- [45] T. Beucler, M. Pritchard, S. Rasp, J. Ott, P. Baldi and P. Gentine, "Enforcing analytic constraints in neural networks emulating physical systems," *Physical Review Letters*, vol. 126, no. 9, p. 098302, 2021. DOI: <https://doi.org/10.1103/PhysRevLett.126.098302> (cit. on p. 11).
- [46] L. D. McClenney and U. M. Braga-Neto, "Self-adaptive physics-informed neural networks," *Journal of Computational Physics*, vol. 474, p. 111722, 2023, ISSN: 0021-9991. DOI: <https://doi.org/10.1016/j.jcp.2022.111722> (cit. on p. 13).
- [47] D. Liu and Y. Wang, "A dual-dimer method for training physics-constrained neural networks with minimax architecture," *Neural Networks*, vol. 136, pp. 112–125, 2021. DOI: <https://doi.org/10.1016/j.neunet.2020.12.028> (cit. on p. 13).

-
- [48] C. Chi, X. Xu and D. Thévenin, “Efficient premixed turbulent combustion simulations using flamelet manifold neural networks: A priori and a posteriori assessment,” *Combustion and Flame*, vol. 245, p. 112 325, 2022. DOI: <https://doi.org/10.1016/j.combustflame.2022.112325> (cit. on p. 14).
- [49] A. Almeldein and N. Van Dam, “Accelerating chemical kinetics calculations with physics informed neural networks,” *Journal of Engineering for Gas Turbines and Power*, vol. 145, no. 9, p. 091 008, 2023. DOI: <https://doi.org/10.1115/1.4062654> (cit. on p. 14).
- [50] T. Kumar, A. Kumar and P. Pal, “Phy-chemnode: An end-to-end physics-constrained autoencoder-neuralode framework for learning stiff chemical kinetics of hydrocarbon fuels,” *Frontiers in Thermal Engineering*, vol. 5, p. 1 594 443, 2025. DOI: <https://doi.org/10.3389/fther.2025.1594443> (cit. on p. 14).
- [51] L. Luo, Q. Liu, J. Sun and Y. Huang, “Exploring surface reaction mechanism using a surface reaction neural network framework,” *Chemical Engineering Science*, vol. 306, p. 121 307, 2025. DOI: <https://doi.org/10.1016/j.ces.2025.121307> (cit. on p. 14).
- [52] T. Kircher and M. Votsmeier, “Machine learning surrogate models for mechanistic kinetics: Embedding atom balance and positivity,” *The Journal of Physical Chemistry Letters*, vol. 16, no. 19, pp. 4715–4723, 2025 (cit. on p. 14).
- [53] F. Biermann, R. Uglietti, F. A. Döppel *et al.*, “Enabling micro-kinetics based simulation of industrial packed-bed reactors by physics-enhanced neural networks,” *Chemical Engineering Journal*, p. 163 598, 2025. DOI: <https://doi.org/10.1016/j.cej.2025.163598> (cit. on p. 14).
- [54] P. O. Sturm and S. J. Silva, “A nudge to the truth: Atom conservation as a hard constraint in models of atmospheric composition using a species-weighted correction,” *ACS Es&t Air*, vol. 2, no. 1, pp. 99–108, 2024 (cit. on p. 14).
- [55] C Coelho, M. F. P. Costa and L. L. Ferrás, “A self-adaptive penalty method for integrating prior knowledge constraints into neural odes,” *arXiv preprint arXiv:2307.14940*, 2023. DOI: <https://doi.org/10.48550/arXiv.2307.14940> (cit. on p. 14).
- [56] B. Lakshminarayanan, A. Pritzel and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017 (cit. on p. 15).
- [57] D. J. MacKay, “A practical bayesian framework for backpropagation networks,” *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992. DOI: <https://doi.org/10.1162/neco.1992.4.3.448> (cit. on p. 15).
- [58] R. M. Neal, *Bayesian learning for neural networks*. Springer Science & Business Media, 2012, vol. 118 (cit. on p. 15).
- [59] H. R. Ferguson, D. H. Bailey and P. Kutler, “A polynomial time, numerically stable integer relation algorithm,” Tech. Rep., 1998 (cit. on p. 34).

- [60] J. Hastad, B. Just, J. C. Lagarias and C.-P. Schnorr, “Polynomial time algorithms for finding integer relations among real numbers,” *SIAM Journal on Computing*, vol. 18, no. 5, pp. 859–881, 1989. DOI: <https://doi.org/10.1137/0218059> (cit. on p. 34).
- [61] S. L. Brunton, J. L. Proctor and J. N. Kutz, “Discovering governing equations from data by sparse identification of nonlinear dynamical systems,” *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016. DOI: <https://doi.org/10.1073/pnas.1517384113> (cit. on p. 34).
- [62] E. Kaiser, J. N. Kutz and S. L. Brunton, “Discovering conservation laws from data for control,” in *2018 IEEE Conference on Decision and Control (CDC)*, IEEE, 2018, pp. 6415–6421. DOI: [10.1109/CDC.2018.8618963](https://doi.org/10.1109/CDC.2018.8618963) (cit. on p. 34).
- [63] S. Greydanus, M. Dzamba and J. Yosinski, “Hamiltonian neural networks,” in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox and R. Garnett, Eds., vol. 32, Curran Associates, Inc., 2019 (cit. on p. 34).
- [64] Z. Liu, Y. Wang, S. Vaidya *et al.*, “Kan: Kolmogorov-arnold networks,” *arXiv preprint arXiv:2404.19756*, 2024. DOI: <https://doi.org/10.48550/arXiv.2404.19756> (cit. on p. 34).
- [65] B. C. Koenig, S. Kim and S. Deng, “Kan-odes: Kolmogorov–arnold network ordinary differential equations for learning dynamical systems and hidden physics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 432, p. 117 397, 2024. DOI: <https://doi.org/10.1016/j.cma.2024.117397> (cit. on p. 34).