Chemical Engineering Journal Advances 23 (2025) 100775

Contents lists available at ScienceDirect

Chemical Engineering Journal Advances

journal homepage: www.sciencedirect.com/journal/chemical-engineering-journal-advances

Hybridised mechanistic and machine learning digital twins for modelling and optimising chemical processes in flow: A comparative analysis of parallel and series-based hybridisation

Nur Aliya Nasruddin [a],*, Nazrul Islam [b], Sergio Vernuccio [c], John Oyekan [b,a]

[a] School of Electrical and Electronic Engineering, The University of Sheffield, S13JD, Sheffield, United Kingdom
[b] Department of Computer Science, University of York, YO10 5GH, York, United Kingdom
[c] School of Chemistry and Chemical Engineering, University of Southampton, SO17 1BJ, Southampton, United Kingdom

ARTICLE INFO

Keywords:
Digital twin
Genetic Algorithm
Hybrid modelling
Machine learning
Optimisation
Physics-Informed Neural Network
Plug flow reactor
Reaction kinetics
Smooth particle hydrodynamics

ABSTRACT

In the field of chemical engineering, accurate prediction of reaction kinetics and concentration profiles is critical for the design and optimisation of industrial processes. However, achieving accurate predictions under variable or limited data conditions remains a major challenge. Despite the growing interest in hybrid models, a systematic comparison of parallel and series-based hybridisation strategies using empirical flow reactor data for digital twin applications has not yet been established. Here we show that PINN architecture can accurately predict concentration profiles and estimate reaction rate constants under both data-rich and data-scarce conditions, while the SPH+GA framework enhances spatial simulation fidelity and enables system-level optimisation through particle-based modelling. The same PINN architecture can be effectively applied in both forward and inverse modes, accurately predicting concentration profiles and estimating reaction rate constants with errors under 2%, even in data-scarce conditions. The SPH+GA framework enables detailed particle-level simulation and global optimisation, offering insight into spatial dynamics and reactor mixing. This series hybrid model achieved an R^2 up to 0.91 and enabled flexible system tuning. These results underscore the broader value of hybrid mechanistic–machine learning frameworks, particularly for process environments with limited or noisy data. Our findings highlight that while PINNs offer high predictive accuracy and lower computational cost, SPH+GA excels in resolving spatial dynamics and supporting system characterisation. These parallel and series hybrid strategies demonstrate complementary strengths for building robust digital twins of chemical processes.

1. Introduction

Machine learning methodologies have shown a strong capability in interpreting complex data representations and modelling physical processes, offering potential solutions to challenging open problems in chemical reaction systems [1–3]. For example, machine learning (ML) models can enhance the accuracy of estimating chemical reaction rates, leading to more precise predictions and insights into reaction dynamics [4,5]. These data-driven approaches can infer underlying patterns from experimental or simulated datasets, enabling accurate estimations of reaction rates and concentration profiles. However, traditional ML models are often constrained by their dependence on large volumes of high-quality data. Their performance typically deteriorates when extrapolating beyond the training domain, which limits their reliability in data-sparse or physically complex environments. In order to address these challenges, the field has seen a growing interest in Scientific Machine Learning (SciML), an emerging paradigm that combines physical principles with data-driven learning [6–8]. These methods, trained on both experimental and numerical data, demonstrate significant potential for optimising complex industrial processes. One particular approach, which has been receiving significant attention in the scientific community, is the Physics-Informed Neural Network (PINN) method. This approach combines knowledge from first-principles models with measurement data from a physical system and has been successful in solving systems of ordinary and partial differential equations (ODEs and PDEs), such as the Schrödinger [9], Allen–Cahn [10], Navier–Stokes [11,12], coupled Navier–Stokes and heat transfer equations [13] as well as forward and inverse problems involving nonlinear PDEs [14]. Some examples of PINNs applications include modelling CO_2 capture processes in solid sorbents [15], modelling isothermal fixed-bed reactors for catalytic CO_2 methanation by integrating reaction kinetics

---

* Corresponding author.
E-mail address: nanasruddin1@sheffield.ac.uk (N.A. Nasruddin).

https://doi.org/10.1016/j.ceja.2025.100775
Received 17 March 2025; Received in revised form 14 May 2025; Accepted 16 May 2025
Available online 2 June 2025
2666-8211/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Nomenclature

Δt	Time step (s)
$\mathbf{r}_i, \mathbf{r}_j$	Position vectors of particles i and j (m)
a_i	Acceleration of particle i (m/s^2)
g	Acceleration due to gravity (m/s^2)
k	Reaction rate constants (M^{-1}s^{-1})
k_1	Reaction rate constants for Model 1 (M^{-1}s^{-1})
k_2, k_3	Reaction rate constants for Model 2 (M^{-1}s^{-1})
m_i, m_j	Mass of particles i and j (kg)
r_i	Reaction rate term for species i
v_i, v_j	Velocity of particles i and j (m/s)
w_b	Weight for boundary condition loss
w_f	Weight for ODE residual loss
w_i	Weight for initial condition loss
w_{obs}	Weight for observational data loss
x, t	Spatial and temporal coordinates

Greek Symbols

α	Reaction distance in SPH modelling
ϵ	Heat transfer coefficient constant
Λ	Thermal conductivity (W/m K)
λ	Unknown physical parameters (inverse mode)
$\mathcal{B}$	Boundary condition operator
$\mathcal{F}$	Differential operator of the system
$\mathcal{I}$	Initial condition operator
μ	Dynamic viscosity (Pa s)
ν	Kinematic viscosity (m^2/s)
ρ	Fluid density (kg/m^3)
τ	Residence time in the reactor (s or min)
θ	Neural Network trainable parameters (weights and biases)
ζ	Scaling factor for system characterisation
h	Smoothing length in SPH (m)
W	SPH kernel function

Roman Symbols

A, B	Chemical species A and B
AD	Automatic differentiation
C, D	Chemical species C and D
C_A, C_B	Concentration of species A and B (mol/L)
C_C, C_D	Concentration of species C and D (mol/L)
C_{A0}	Initial concentration of reactants A (mol/L)
C_{B0}	Initial concentration of reactants B (mol/L)
$FFNN$	Feed-Forward Neural Network
L_{BC}	Loss terms for boundary conditions
L_{IC}	Loss terms for initial conditions
L_{obs}	Loss terms for data observations
L_{ODE}	Loss terms for ODEs
L_{total}	Total loss terms
N_b	Training points for boundary conditions
N_f	Collocation points for the differential equation
N_i	Training points for initial conditions
N_{obs}	Observational data points
ODE	Ordinary Differential Equations
P	Pressure (Pa)
PDE	Partial Differential Equations
$PINN$	Physics-Informed Neural Networks
Q	Volumetric flow rate (mL/s)
R^2	Coefficient of determination
$RMSE$	Root Mean Square Error
SPH	Smoothed Particle Hydrodynamics
T	Temperature (°C)
V	Reactor volume (mL)

and physical laws governing the methanation process [16], and solving stiff chemical kinetic problems by incorporating the quasi-steady-state assumption [17]. Additionally, PINNs have been used to predict functional parameters in sulfur-driven autotrophic denitrification processes leading to improved predictions under varying conditions [18]. Similarly, PINN have been applied in hydrodynamic voltammetry to analyse transport-limited currents and kinetically controlled processes with analytical expressions while adhering to physical laws [19]. PINNs also have been applied to Population Balance Model (PBM) equation for particle aggregation and breakage [20] and convection–diffusion–reaction equations for reacting flows [21]. By overcoming the limitations of traditional numerical methods, particularly their reliance on mesh generation, PINNs offer a mesh-free framework and broad applicability across various governing equations enhancing both simulation accuracy and computational efficiency. In the context of digital twin applications for chemical processes, PINNs have been increasingly used to enhance simulation and prediction capabilities of chemical engineering. For example, Tang et al. highlighted the potential of PINNs for real-time digital twin simulations of chromatography processes [22]. In their study, the authors optimised the structure, training data, and complexity of PINNs tailored to the lumped kinetic model for chromatography, demonstrating their ability to simulate breakthrough curves quickly and accurately under various conditions. This approach offers a faster and more efficient alternative to traditional numerical methods. Bibeau et al. used PINNs to predict reaction kinetics for biodiesel production in microwave reactors, effectively creating a digital twin of the microwave-assisted reaction [23]. By performing regression on experimental data, the authors aligned the physics of the process with rate constants and temperature dependencies, enabling extrapolation to different power inputs. This digital twin approach aids in the design, optimisation, and control of microwave reactors for biodiesel production, providing a compact and efficient solution for kinetic modelling. Lastly, PINNs were applied to create a digital twin for monitoring a nuclear reactor by solving point kinetic equations (PKEs). The study demonstrated the feasibility of using PINNs for nuclear reactor monitoring, showing their effectiveness in predicting reactor behaviour both within and outside the range of available data [24]. Since a digital twin is a virtual model that replicates and simulates the behaviour of a physical system, there are many ways of achieving this (See Fig. 1). For example, some digital twin leverage mathematical models to describe the dynamic behaviour of a system thereby enabling precise predictions of how the system will respond to varying conditions over time [25]. This dynamic modelling is particularly valuable for complex chemical systems where understanding and predicting interactions are essential for optimising performance. In addition to dynamic modelling, digital twin can also provide vivid visual representation of the processes occurring in a physical system under study [26,27]. Smoothed Particle Hydrodynamics (SPH) and Computational Fluid Dynamics (CFD) stand out from among the various modelling techniques that provide digital twin with such visual representation. CFD has traditionally played a

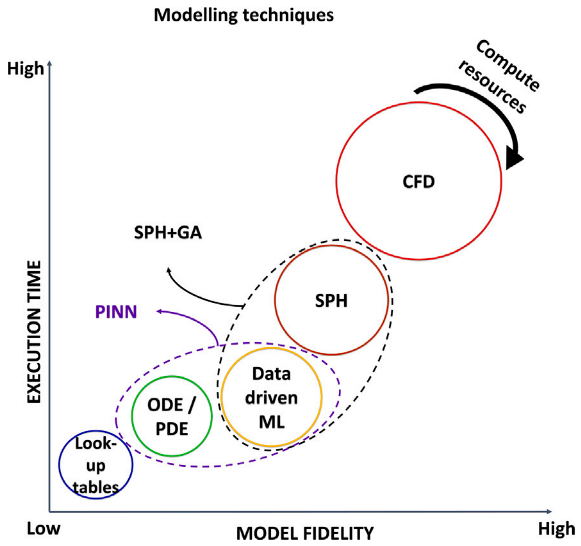


Fig. 1. Various modelling techniques with their level of fidelity, execution time and compute resources required. The size of the ovals indicate the level of compute resources required.

central role in representing the behaviour of physical systems, particularly in fluid dynamics [28,29]. Through the use of meshes, CFD divides the simulated physical system into discrete grids within which the behaviour of the physical system is calculated and predicted while taking into consideration the global behaviour of the system and the links between the discrete grids. As a result, mesh-based CFD methods offer powerful capabilities but become computationally expensive and complex when dealing with highly dynamic physical systems or irregular geometries. In this regard, Smoothed Particle Hydrodynamics (SPH) contributes significantly by simulating fluid dynamics at a granular level while providing visually the complex interactions within the system, such as mixing, reaction kinetics as well as flow behaviour at a high fidelity and accuracy as CFD [30]. This simulation capability is crucial for understanding phenomena that are difficult to capture with traditional modelling techniques alone. As a first-principles dynamic modelling technique, SPH has been used for simulating the impact of fluid dynamics on various physical systems. This includes the modelling of turbulent diffusion in anaerobic digestion [31,32], chemical reactions [33], star formations [34] and slurry mixing [35]. In fluid–structure interaction simulations, SPH has been applied to capture the behaviour of reacting flows, such as combustion processes [36]. However, its use in simulating chemical reactions is not as wide spread when compared to dedicated techniques like molecular dynamics or quantum chemistry simulations. In [31], Yan et al. applied SPH to investigate the impact of turbulent diffusion on mixing processes within biochemical reaction models while focusing on anaerobic digestion. That study extended the SPH derived Coupled Hydrodynamics and Anaerobic Digestion (CHAD) codebase to incorporate turbulent diffusion and validated it against OpenFOAM (CFD derived software) simulations. Their approach adapted SPH to overcome challenges in multi-dimensional flows and heat release in reacting flows. The technique underscored the advantages of Lagrangian methods in convection-dominated flows, offering a novel perspective on direct simulation of chemical reactions within the SPH framework. However, according to our current knowledge, no approach in literature has integrated or hybridised SPH with ML techniques for flow chemistry.

On the other hand, in line with recent developments, Zhu et al. integrated ML with CFD in [37] for ML-assisted sub-grid models for gas–solid flows. In particular, they used artificial neural networks and gradient boosting techniques trained on high-fidelity CFD–DEM datasets [37]. Their approach significantly improved the accuracy of coarse-grid simulations in fluidised beds by embedding data-driven corrections into traditional two-fluid models. Building on this foundation, Zhu et al. provided a comprehensive review of ML applications in hydrodynamics, transport phenomena, and reaction modelling in multiphase systems [38]. Their review highlighted data scarcity as one of the key challenges when applying machine learning (ML) to multiphase flow and reactor systems. This is because ML models typically require large, consistent, and high-quality datasets to perform effectively. As a result, Physics-Informed Neural Networks (PINNs) represents a promising solution to address this issue. This is because unlike conventional ML models that rely purely on data, PINNs incorporate physical laws directly into the learning process. This integration enables them to deliver accurate predictions even with limited data, making them particularly well-suited for addressing data-scarce challenges in chemical engineering. The review by Zhu et al. also outlined several promising future directions including advancing physics-informed ML methods, developing automated toolkits and robotic platforms for flow optimisation, using ML to accelerate computational simulations, as well as creating dynamic digital twin platforms that integrate physical experiments with virtual simulations for more efficient and intelligent system design. As a result, the hybridisation of first-principles or mechanistic modelling with ML offers a lot of unique advantages.

In this study, our aim is to investigate two different types of hybrid modelling strategies for simulating and optimising chemical reactions under flow conditions, using empirical reactor data. These strategies are designed within the framework of digital twin for chemical processes and combine physical modelling with data-driven learning. Our first approach makes use of Physics-Informed Neural Networks (PINNs) to address both forward and inverse kinetic problems using full and reduced real flow-ramp experimental datasets from a flow reactor system. A consistent PINN architecture and optimised network parameters are applied across different operational regimes, including forward modelling at 65 °C and inverse modelling at 75 °C, to assess the model's generalisation capability and potential for transfer learning. Our second approach introduces a hybrid framework that coupled physics-based simulation with data-driven optimisation by integrating Smoothed Particle Hydrodynamics (SPH) with a Genetic Algorithm (GA). In this setup, SPH simulates fluid and reactive behaviours at the particle level, capturing detailed spatial dynamics, while GA performs global optimisation by adjusting operational parameters based on SPH-generated outputs. This chemically informed SPH+GA hybrid system enables flexible exploration of design spaces and system performance tuning, particularly in scenarios with limited experimental data. These two methods are conceptualised within a hybrid modelling architecture (see Figure Fig. 2), where the PINN-based approach is presented as a hybrid model operating in parallel configuration and the SPH+GA framework exemplifies as a hybrid model operating in series configuration, offering a generalised view of how physics and data can be fused in the realisation of digital twin for chemical process systems. Our contributions are as follows:

1 We apply a PINN architecture to real flow-ramp experimental data in a plug flow reactor, under both data-rich and data-scarce conditions, with the aim of assessing predictive generalisability and enabling kinetic inference from partial observations. Our PINN workflow first identifies optimal hyperparameters in forward mode using full and reduced datasets and then applies these configurations in inverse mode to estimate kinetic parameters from unseen data. This is unlike previous work that used synthetic or lumped kinetic data with PDEs and ODEs in various chemical processes.
2 We show that our PINN architecture is able to generalise effectively and predict system behaviour beyond its training region, even when observational data from the most informative segments of the reaction dynamics is excluded.

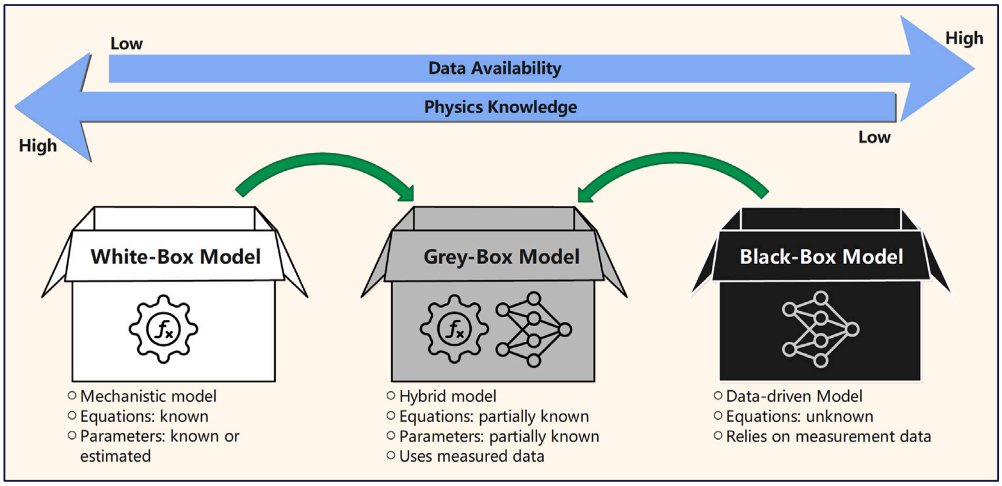


Fig. 2. Conceptual classification of modelling approaches based on physical knowledge and data availability. The figure outlines the positioning of white-box, grey-box, and black-box models according to their reliance on physics-based information and empirical data.

3. We introduce a chemically informed framework that couples Smoothed Particle Hydrodynamics (SPH) with a Genetic Algorithm (GA) thereby enabling detailed particle-level simulation and data-driven optimisation to support system tuning in data-scarce environments.
4. We contribute to the development of hybrid modelling frameworks for chemical process systems by exploring two complementary approaches for constructing digital twins using empirical flow reactor data. The developed PINN framework is presented as a hybrid model operating in parallel while integrating physics directly into the learning process. On the other hand, the SPH+GA framework is characterised as a hybrid model operating in series, whereby physical simulation is followed by data-driven optimisation. These contributions collectively advance the development of adaptable, accurate and computationally efficient digital twins for flow-based chemical processes.

The structure of this paper is as follows: Section 2 outlines the modelling methodologies, the mathematical theories, physical platform and reaction kinetic model applied in the development of our PINN and SPH+GA algorithms. This section also describes the architecture of the PINN framework. In Section 3, we present and analyse the results obtained from both methods, evaluating their predictive accuracy and computational efficiency. In Section 4, we provide a detailed comparison of the performance of PINNs and SPH+GA while providing insights into the strengths and limitations of each approach. We paid particular attention to the ability of PINNs to predict and determine unknown parameters across various chemical flow scenarios. Finally, the paper concludes in Section 5 by summarising the key findings and discussing future research directions.

2. Methodology

Hybrid modelling techniques combine the strengths of mechanistic models (physics-based or first-principles model) with machine learning data-driven techniques. This integration allows for improved system representation, particularly where a partial physical knowledge exists but it is insufficient to fully capture system dynamics on its own. Fig. 2 illustrates the conceptual classification of modelling approaches in terms of known physics and data availability.

Mechanistic models also known as a white-box models are grounded in fundamental physical knowledge. These models apply established physical laws such as conservation of mass, energy, as well as momentum. They are typically represented through mathematical formulations such as ordinary differential equations (ODEs) or partial differential equations (PDEs) to describe a particular system. The primary benefits include their interpretability, grounded in fundamental physics and chemistry, allowing for a clear understanding and explanation of underlying system behaviours. However, they typically require detailed knowledge of system properties, extensive experimental data for parameter estimation, and significant computational resources, especially when solving complex PDE systems. On the other hand, data-driven models which are also referred to as black-box models, rely on empirical data to establish the relationships between inputs and outputs without incorporating the underlying physical laws. Techniques such as artificial neural networks (ANNs), support vector machines (SVMs) and Gaussian process regression (GPR) are commonly used to model complex, non-linear systems. The benefits of data-driven models include their flexibility in handling complex and nonlinear systems, efficiency in providing rapid predictions for real-time decision-making, and adaptability to new data. However, they depend heavily on large, high-quality datasets, suffer from limited interpretability, and typically have poor performance when extrapolating beyond the range of their training data. This has led to an increasing interest in hybrid modelling which is also known as a grey-box model. It is an approach that combines mechanistic (white-box) models based on physical laws with data-driven (black-box) models that are derived from experimental or operational data. By combining both methodologies, hybrid models aim to improve prediction accuracy, enhance interpretability and reduce reliance on extensive datasets.

Hybrid models are generally classified into two core configuration namely series or parallel [39,40]. In a parallel hybrid model, both the physics-based and machine learning components are trained simultaneously, by embedding physical constraints directly into the learning objective. In a series hybrid model, the output of a physics-based simulation is passed into a data-driven algorithm, typically for optimisation. In this work, we compare the use of series and parallel configurations in developing a digital twin framework for modelling and optimisation of chemical processes in flow systems.

2.1. Physical system (the experimental reactor platform)

In order to conduct a comparative analysis, we used previously reported reaction models and results from a continuous flow reactor platform [41]. The main features of the reactor setup are described in detail in the Supplementary Information (SI) with a schematic represented in Figure S1. Further details about the reaction models are provided in the next section. The parameters for each model, including reactor volume and initial reactant concentrations are summarised in Table 1. The Table outlines the specific conditions under which

Table 1
Reaction model and its corresponding parameters.

Model	Reaction model	Temperature (°C)	Reactor volume (mL)	C_{A_0} (mol/L)	C_{B_0} (mol/L)
Model 1	$A+B\xrightarrow{k_1}C+D$	65	3.5	0.167	0.167
		75	3.5	0.167	0.167
Model 2	$A+B\xrightarrow{k_2}C,\quad A+C\xrightarrow{k_3}D$	130	5.0	0.5	1.5
		150	5.0	0.5	1.5
		190	3.5	0.5	0.3
		210	3.5	0.5	0.3

each reaction model was evaluated. We also utilised the time-series concentration data provided in [41] to evaluate the performance of SPH and PINNs under varying data availability conditions. For Model 1, two datasets were created: one with the full original data points, providing detailed concentration profiles across the reactor residence time and another with a reduced dataset comprising of one-third of the original points, sampled from the 0–5 min residence time interval. For Model 2, the analysis was conducted using all available experimental data without reduction due to the limited size of the dataset. This evaluation was undertaken to demonstrate our method's capability to process data from a variety of sources. The experimental data that were used for Model 1 and 2 are illustrated in Figure S2 and S3 of the Supplementary Information (SI).

2.2. Physical model (the reactor mathematical model)

In a Plug Flow Reactor (PFR), the change in the concentration of each species along the axis of the reactor can be described by differential equations derived from mass balances with respect to residence time. Assuming a steady-state operation and ignoring axial dispersion, the reactor model equation can be simplified as in Eq. (1). Where (τ) is the residence time, C_i and r_i are the concentration and the rate of reaction for species i, respectively.

$$\frac{dC_i}{d\tau}=r_i \tag{1}$$

To validate the reactor model and analyse the chemical kinetics, we explore two distinct reaction models, each subjected to different temperature conditions and ranges of data availability. These reaction models are now discussed.

2.2.1. Model 1: One-step reaction model

In this model, we consider a simple reaction where two reactants, A and B interact to form two products, C and D. This reaction can be applied to describe the exemplary transformation of phenol and acetyl chloride into phenyl acetate and hydrochloric acid. The reaction mechanism is described by the following equation:

$$A+B\xrightarrow{k_1}C+D \tag{2}$$

The corresponding differential equations describing the change in concentrations of the reactants and products are as follows:

$$\frac{dC_A}{d\tau}=-k_1\cdot C_A\cdot C_B \tag{3}$$

$$\frac{dC_B}{d\tau}=-k_1\cdot C_A\cdot C_B \tag{4}$$

$$\frac{dC_C}{d\tau}=k_1\cdot C_A\cdot C_B \tag{5}$$

$$\frac{dC_D}{d\tau}=k_1\cdot C_A\cdot C_B \tag{6}$$

The rate constant k_1 follows an Arrhenius-type temperature dependency. The reaction has been studied under two different operating temperatures at 65 °C and 75 °C.

2.2.2. Model 2: Two-step reaction model

The second reaction model introduces a more complex scenario involving consecutive reactions. In the first reaction, species A and B react to produce an intermediate product C. Subsequently, in the second reaction, the intermediate C reacts with A to form the final product D. This mechanism represents the typical formation reaction of metoprolol from isopropylamine, followed by a subsequent reaction to form a bis-substituted product. The reaction equations for this system are given in Eqs. (7) and (8).

$$A+B\xrightarrow{k_2}C \tag{7}$$

$$A+C\xrightarrow{k_3}D \tag{8}$$

The material balances for the four species involved in this reaction network are expressed by differential Eq. (9)–(12). In this kinetic model, k_2 and k_3 are the rate constants for the first and second reaction steps, respectively.

$$\frac{dC_A}{d\tau}=-k_2\cdot C_A\cdot C_B-k_3\cdot C_A\cdot C_C \tag{9}$$

$$\frac{dC_B}{d\tau}=-k_2\cdot C_A\cdot C_B \tag{10}$$

$$\frac{dC_C}{d\tau}=k_2\cdot C_A\cdot C_B-k_3\cdot C_A\cdot C_C \tag{11}$$

$$\frac{dC_D}{d\tau}=k_3\cdot C_A\cdot C_C \tag{12}$$

For each reaction model, the process starts with the identification of the optimal hyperparameters to develop a robust PINN model that accurately predicts the reactor behaviour. Once the optimal model is established through training in forward mode, this PINN model is applied in inverse mode at different temperatures to estimate the unknown parameter, specifically the rate constants k of the system. This methodology enables the validation of the model's consistency and adaptability across varying temperature levels, reaction complexities, and data availability. Subsequently, the predicted concentration profiles are compared against experimental data and the numerical solutions of the system's ODEs, facilitating a comprehensive evaluation of the model's accuracy.

2.3. Physics-informed neural network (PINN) framework

The application of Physics-Informed Neural Networks (PINNs) enables the prediction of concentration profiles and the determination of the unknown reaction rate constants. Fig. 3 illustrates PINNs parallel hybrid architectural approach that enables the integration of a neural network (Black-box) with the physical process governing equations (White-box). The integration of both white and black-box models is achieved by embedding the governing equations of the targeted reaction model into the neural network's loss function. The function of PINN's loss function is to include terms that account for deviations from the governing equations into the learning process, alongside the traditional data. This approach ensures that the outputs not only fit the available data but also satisfy the governing physical equations. PINNs stand out by ensuring that the predictions align with established

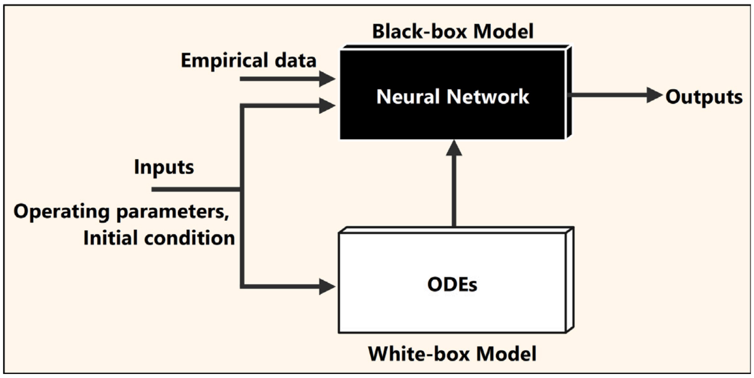


Fig. 3. Conceptual diagram of PINN as a parallel hybrid model.

physical principles, making them especially valuable when data is sparse or expensive to obtain.

In this work, we consider a physical system described by a differential equation of the form:

$$\mathcal{F}u(x,t) = f(x,t) \tag{13}$$

where:

- $\mathcal{F}$ is a differential operator of the system.
- $u(x,t)$ is the solution function, which the neural network aims to approximate.
- $f(x,t)$ is a known function, representing sources, sinks, or other terms in the equation.
- x and t represent spatial and time coordinates, respectively.

A PINN seeks to approximate the solution u using a neural network. The neural network, parametrised by weights and biases (θ), takes the spatial coordinates ($\mathbf{x}$) and time (t) as continuous inputs and outputs an estimate of the solution, $\hat{\mathbf{u}}(\mathbf{x}_i, t_i; \theta)$. The key innovation of PINNs is to include the differential equation itself as part of the loss function during training. This is achieved by computing the derivatives of the neural network's output with respect to its inputs using automatic differentiation (AD). These computed derivatives are then plugged into the differential operator $\mathcal{F}$. The residual of the differential equation is then minimised during training. The loss function now comprises the residuals of the differential equation, boundary or initial conditions, and a term for the data fit. It can be formulated as:

$$\mathcal{L}_{\text{Total}}(\theta; N) = w_f \mathcal{L}_f(\theta; N_f) + w_b \mathcal{L}_b(\theta; N_b) + w_i \mathcal{L}_i(\theta; N_i) + w_{obs} \mathcal{L}_{obs}(\theta; N_{obs}) \tag{14}$$

where:

$$\mathcal{L}_f(\theta; N_f) = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \mathcal{F}(\hat{\mathbf{u}}(\mathbf{x}_i, t_i; \theta)) \right|^2 \tag{15}$$

$$\mathcal{L}_b(\theta; N_b) = \frac{1}{N_b} \sum_{i=1}^{N_b} \left\| \mathcal{B}(\hat{\mathbf{u}}(\mathbf{x}_i, t_i; \theta)) \right\|^2 \tag{16}$$

$$\mathcal{L}_i(\theta; N_i) = \frac{1}{N_i} \sum_{i=1}^{N_\text{i}} \left\| \mathcal{I}(\hat{\mathbf{u}}(\mathbf{x}_i, t_i; \theta)) \right\|^2 \tag{17}$$

$$\mathcal{L}_{obs}(\theta; N_{obs}) = \frac{1}{N_{obs}} \sum_{i=1}^{N_{obs}} \left\| \hat{\mathbf{u}}(\mathbf{x}_i, t_i; \theta) - \mathbf{u}_{obs,i} \right\|^2 \tag{18}$$

In the series of equations in the loss function, the training data consists of four components, namely $\{N_f, N_b, N_i, N_{\text{obs}}\}$. These represent the number of residual points where the differential equation is enforced (known as collocation points), the number of boundary condition training points, the number of initial condition training points, and the number of observational data points, respectively. These are typically sampled across the domain of interest. Moreover $\{w_f, w_b, w_i, w_{obs}\}$ are the weights assigned to each of the loss function. Meanwhile $\hat{\mathbf{u}}(\mathbf{x}_i, t_i; \theta)$ is the neural network's output approximation of the solution at the coordinates $(\mathbf{x}_i, t_i)$ and $\mathbf{u}_{\text{obs},i}$ are the observed values at these points. Hence, considering the problem of Model 1 reaction in a PFR, we propose the loss function as follows:

$$\mathcal{L}_{ODE}(\theta; N_f) = \frac{1}{N_f} \sum_{i}^{N_f} \left| \left(\frac{dC_i}{d\tau} + r_i \right) \right|^2 \tag{19}$$

$$\mathcal{L}_{BC}(\theta; N_b) = \frac{1}{N_b} \sum_{j}^{N_b} \left| C_i(\tau_j; \theta) \right|^2 \tag{20}$$

$$\mathcal{L}_{IC}(\theta; N_i) = \frac{1}{N_i} \sum_{k}^{N_i} \left| C_{A0}, C_{B0}(\tau_k; \theta) \right|^2 \tag{21}$$

$$\mathcal{L}_{obs}(\theta; N_{obs}) = \frac{1}{N_{\text{obs}}} \sum_{l}^{N_{obs}} \left| \hat{C_A} - C_{A_{obs}}, \hat{C_C} - C_{C_{obs}}(t_l; \theta) \right|^2 \tag{22}$$

$$\begin{aligned} \mathcal{L}_{Total}(\theta; N) = {} & w_{ODE} \mathcal{L}_{ODE}(\theta; N_f) + w_{BC} \mathcal{L}_{BC}(\theta; N_b) + w_{IC} \mathcal{L}_{IC}(\theta; N_i) \\ & + w_{obs} \mathcal{L}_{obs}(\theta; N_{obs}) \end{aligned} \tag{23}$$

In forward mode, the goal is to train the neural network by optimising the trainable parameters θ, which include the weights and biases of the network, to minimise the total loss. This can be expressed as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{Total}(\theta; N) \tag{24}$$

In inverse problems, additional parameters λ, representing unknown physical parameters are introduced and optimised simultaneously with θ. We add an extra loss term to Eqs. (23) to the total loss function which incorporates these parameters and is expressed as:

$$\begin{aligned} \mathcal{L}_{Total}(\theta, \lambda; N) = {} & w_{ODE} \mathcal{L}_{ODE}(\theta, \lambda; N_f) + w_{BC} \mathcal{L}_{BC}(\theta, \lambda; N_b) \\ & + w_{IC} \mathcal{L}_{IC}(\theta, \lambda; N_i) + w_{obs} \mathcal{L}_{obs}(\theta, \lambda; N_{obs}) \end{aligned} \tag{25}$$

The solution for the inverse problem is then obtained by simultaneously optimising θ and λ together as:

$$\theta^*, \lambda^* = \arg\min_{\theta,\lambda} \mathcal{L}_{\text{total}}(\theta, \lambda; N) \tag{26}$$

Here, θ represents the trainable parameters of the neural network, while λ corresponds to unknown physical parameters. In this study λ represents the reaction rate constant k. The observed data provide additional constraints, particularly for the inverse problem, thereby improving the identifiability of the unknown parameter k. The optimisation process applies gradient-based methods, such as Adam or L-BFGS, that leverage an adaptive learning rate to ensure efficient and robust convergence for both θ and λ. To compute the partial derivatives required in the ODE residuals, automatic differentiation (AD) is utilised. AD evaluates these derivatives exactly by applying the chain rule through backpropagation, ensuring accuracy and computational efficiency in the training process.

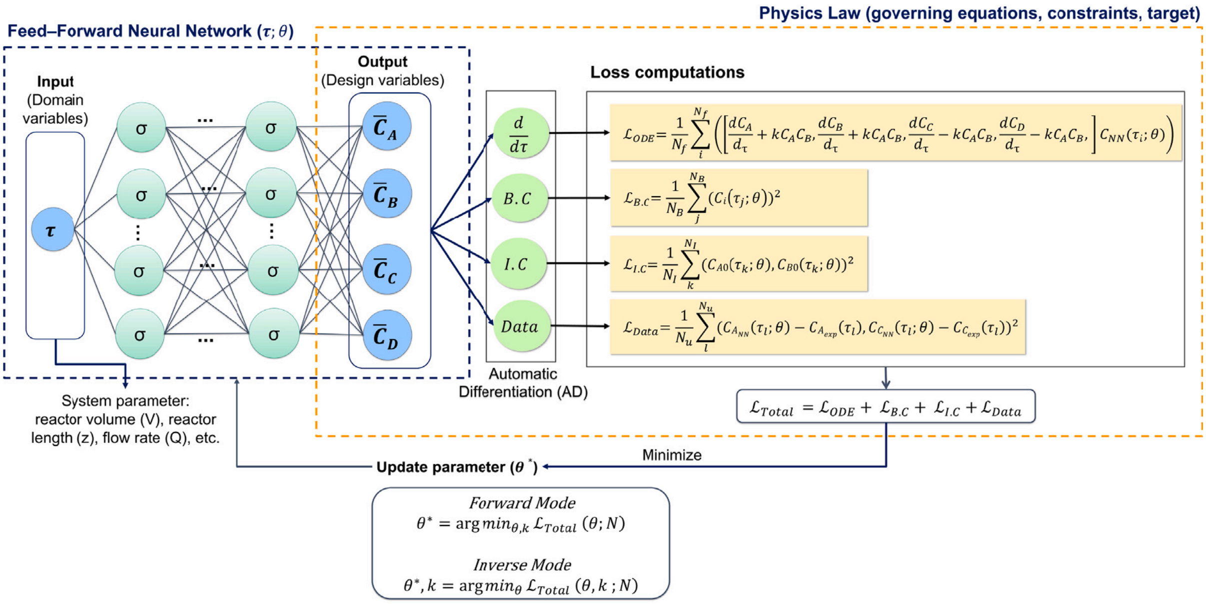


Fig. 4. Physics-Informed Neural Network (PINN) framework designed for the one-step reaction (Model 1) in a continuous flow reactor.

2.3.1. PINN architecture

The PINN architecture used for this work is illustrated in Fig. 4 which integrates a feed-forward neural network (FFNN) with fundamental physical laws. The FFNN takes inputs such as initial concentrations of the species (C_{A0}, C_{B0}), temperature (T), and residence time (τ), processing these through multiple hidden layers composed of neurons connected by weights ($w_{i,j}$) and biases (b_i). The weight and biases parameters are adjusted during training to minimise prediction errors. In the forward problem, the network outputs the predicted desired product concentrations ($\bar{C}_A$, $\bar{C}_B$, $\bar{C}_C$, $\bar{C}_D$). Meanwhile in the inverse problem, the network predicts that the unknown parameter for this case is the rate constant, k. The governing differential equations describe the dynamics of the reaction as shown in Eqs. (3)–(6) and ((9) to (12)) for Model 1 and Model 2 case study, respectively. The total loss functions L_{Total} in the PINN include physics-based losses, L_{ODE}, L_{BC} and L_{IC}, which are associated with the differential equations, boundary and initial conditions, respectively. Meanwhile, the empirical loss L_{obs}, measures the discrepancy between the network's predictions and observed data. For Model 1, L_{obs} is computed using concentration data of C_A and C_C, while for Model 2 using concentration data of C_A, C_C and C_D. Automatic differentiation is applied to compute the gradients of the loss functions with respect to the network parameters, facilitating their optimisation to minimise the total loss.

2.3.2. Optimising PINN architecture via its hyperparameters

Neural networks have various hyperparameters that dictate both the structure of the network and its training process. Hyperparameters that dictate the network structure include the activation function, the number of hidden layers and the number of hidden neurons. The training process also has hyperparameters, notably the learning rate and the number of training iterations. The learning rate controls the size of the steps taken when updating the trainable parameters. Due to the number of hyperparameters and their effect on the performance of PINN structures, systematic studies are necessary to identify an appropriate set of optimal hyperparameters. Towards this, a grid search algorithm is a straightforward tool for hyperparameter tuning. This involves discretising the hyperparameter space into a grid with every combination within the grid evaluated using performance metrics such as the coefficient of determination (R^2) and the Root Mean Square Error (RMSE). High R^2 values close to 1.0 and low RMSE values indicate a strong correlation between the predicted and actual concentrations. However, though the grid search method provides a systematic approach to explore the hyperparameter space, it is computationally intensive. Nevertheless, despite its computational intensiveness, the grid search ensures a comprehensive evaluation of all possible combinations, offering a thorough understanding of the impact of hyperparameters on model performance. In order to implement and evaluate the algorithm, we used DeepXDE, a Python-based library for scientific computing that leverages PyTorch or TensorFlow as its computational engine [42]. DeepXDE is an open source framework for Physics-Informed Neural Networks (PINN), specifically designed to solve forward and inverse problems involving ordinary differential equations (ODE) and partial differential equations (PDE) as shown in Algorithm 1 and 2, respectively. The framework addresses problems defined by initial and boundary conditions, along with the available measurements. For the ODE numerical solver, we applied the `solve_ivp` function from the `scipy.integrate` library.

2.4. Chemically-informed smoothed particle hydrodynamics (SPH) + genetic algorithm (GA) framework

We used a series hybrid modelling architecture for the chemically informed SPH+GA framework used in this study (See Fig. 5). In this configuration, SPH is used to simulate the physical behaviour of the system, such as particle motion, concentration distribution and flow dynamics based on established governing equations. This forms the mechanistic core (white box) of the hybrid modelling system. Subsequently, Genetic Algorithm or any other optimisation algorithm is then used for system parameter characterisation. This two-stage setup fits the definition of a series hybrid model in which the black-box model (GA) relies on the outputs of the white-box model (SPH). In the development and application of a chemically informed SPH, the chemical interactions between molecules and their dynamics in the simulated reactor were taken into account. As a result, we discuss the fluid dynamics equations governing SPH in Section 2.4.1. We also discuss in Sections 2.4.2 to 2.4.4 how we modelled the reactor chemical processes mentioned in Section 2.2.1. Furthermore, in Section 2.4.5, we discuss the black-box approach we used for system characterisation. This step is important in order to achieve a one-to-one representation between the physical system and a digital twin model.

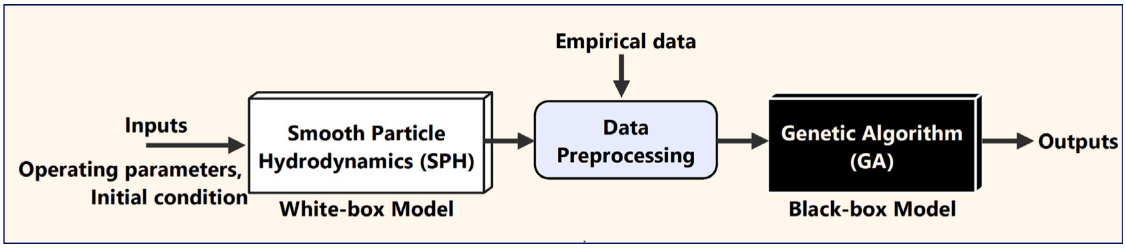


Fig. 5. Conceptual diagram of SPH+GA as a series hybrid model.

Algorithm 1 PINN algorithm for Forward Mode with DeepXDE

```
1: Define the reactor/reaction model ODE include the rate constant, $k_1$ and its related parameters using the Tensorflow grammar.
2: Define the boundary conditions.
3: Load and define the observation points using the PointSetBC module ($C_{A_{obs}}$ and $C_{C_{obs}}$) for reaction model 1, $C_{A_{obs}}$, $C_{C_{obs}}$ and $C_{D_{obs}}$) for reaction model 2.
4: Define the computational domain using the geometry module.
5: Specify the initial conditions (initial concentration of $C_{A_0}$ and $C_{B_0}$ using the IC module.
6: Combine the geometry module, ODE, observation points and IC module together into data.PDE.
7: Specify the training distribution, training data and set the number of points to be sampled in data.PDE.
8: Construct the neural network and choose the FFNN module to represent the feed-forward neural network, specify the FFNN input, output and layer and neuron size, activation function, and initialiser.
9: Define the Model by combining data.PDE in Step 6 and the FFNN in Step 8.
10: Call Model.compile and set the optimisation hyperparameters, such as optimiser and learning rate.
11: Call Model.train to train the network by minimising the loss function.
12: Call Model.predict to predict the ODE solutions and reactor behaviour.
```

Algorithm 2 PINN algorithm for Inverse Mode with DeepXDE

```
1: Define the reactor / reaction model ODE without the rate constant, $k_1$ and its related parameters using the Tensorflow grammar.
2: Define the unknown parameter to be identified, the rate constant using the Variable module ($k_1$ for reaction model 1, $k_2$ and $k_3$ for reaction model 2).
3: Apply the same network configuration from the forward mode (defined in Step 8 of Algorithm 1).
4: Apply the same optimiser settings from the forward mode(Step 10 of Algorithm 1).
5: Repeat Steps 2–7 from Algorithm 1.
6: Define callbacks module to monitor the behaviour at the specified period of iterations and stored the unknown parameter value.
7: Call Model.compile and set the optimisation hyperparameters, such as optimiser and learning rate.
8: Call Model.train to train the network by minimising the loss function.
9: Call Model.predict to predict the ODE solutions, reactor behaviour, and unknown parameters.
```

2.4.1. Fundamental equations of smoothed particle hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH) is a mesh-free computational method in which particles discretise both the fluid and the boundaries of a system. In SPH, particles are used to discretise both the fluid and boundaries of the system. Each particle carries physical properties such as mass, density, and viscosity [43]. Furthermore, the SPH methodology attempts to ensure that the principles of conservation of mass, momentum and energy are respected in the hydrodynamics system. In order to achieve the principles mentioned above, the domain must be discretised to approximate the behaviour of the hydrodynamic system at specific discrete locations [44]. The majority of the hydrodynamic problems are solved by the weakly compressible SPH (WCSPH) scheme. Throughout the SPH simulation, the mass of each particle is kept constant while the density of the particles changes according to the continuity Eq. (27) [44]. The conservation of mass is governed by the continuity equation (Eq. (27)), which describes the fundamental principle of mass preservation in a system. In this context, the density ρ_i of particle i is estimated using Eq. (28) where m_j is the mass of the particle j and $W(\mathbf{r}_i - \mathbf{r}_j, h)$ denotes the smoothing kernel function. This formulation applies a summation approach to account for the contributions of neighbouring particles, thereby ensuring mass conservation within the system.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{27}$$

$$\rho_i = \sum_j m_j W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{28}$$

The conservation of momentum in fluid dynamics is governed by the Navier–Stokes equation, which, for a compressible fluid, can be written as:

$$\frac{d\mathbf{v}}{dt} = -\frac{1}{\rho}\nabla P + \nu \nabla^2 \mathbf{v} + \mathbf{g} \tag{29}$$

In SPH, the momentum equation for a particle i is divided into pressure and viscosity components, given by:

$$f_{pv,i} = f_{pressure,i} + f_{viscosity,i} \tag{30}$$

where $f_{pressure,i}$ and $f_{viscosity,i}$ are the forces generated by pressure and viscosity of the particles and given in Eqs. (31). This summation accounts for the pressure contributions from neighbouring particles, with P_i and P_j being the pressures at particles i and j respectively. Viscosity forces, which account for the diffusive transport of momentum between neighbouring particles, are expressed by Eq. (32):

$$\mathbf{f}_{\text{pressure},i} = -\sum_j m_j \left(\frac{P_i}{\rho_i^2} + \frac{P_j}{\rho_j^2} \right) \nabla W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{31}$$

$$\mathbf{f}_{\text{viscosity},i} = \eta \sum_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W(\mathbf{r}_i - \mathbf{r}_j, h) \tag{32}$$

In addition to the forces listed above, the external forces generated by the pump are included in Eq. (33):

$$\mathbf{f}_{\text{ext},i} = \frac{V_P}{t} \tag{33}$$

This leads to the total acceleration generated on a particle i as Eq. (34), which influences the velocity of each particle and consequently the position (Eq. (35)).

$$\mathbf{a}_i = \frac{\mathbf{f}_{\text{pressure},i} + \mathbf{f}_{\text{viscosity},i} + \mathbf{f}_{\text{ext},i}}{\rho_i} \tag{34}$$

$$\mathbf{v}_i(t+\Delta t) = \mathbf{v}_i(t) + \mathbf{a}_i \Delta t \tag{35}$$

Where ρ is the density of the fluid, $\mathbf{v}$ the velocity vector field, t the time, P is pressure, v is the kinematic viscosity, $\mathbf{f}_{ext,i}$ represent external forces, $\nabla\cdot$ denotes the divergence operator, ∇^2 is the Laplacian operator, E is the total energy, T is the temperature of the fluid and μ is the dynamic viscosity.

2.4.2. *Reactor modelling (boundary conditions)*

In building physical helical coiled reactors, tubes are coiled around a heated column in order to save space. In this type of reactors, the residence time τ of the reagents is governed by the force generated by the pump and the length l of the reactor. As a result, the digital representation of the reactor was modelled by unfurling the coil of the physical helical coiled reactor into a straight line, as illustrated in Figure S4 in the Supporting Information. This simplified the simulation of the physical reacting system. In the future, this system can be improved by incorporating the actual design of the physical reactor in the simulation. This approach simplifies the representation of complex geometries in the simulation. Particles were used to represent the walls of the simulated reactor and used to define the limits as well as boundaries of the fluid domain. The wall particles were solved like any other fluid particles. However, the particle velocities were set to zero. This ensured that their positions were fixed and followed the boundaries of the geometry of the reactor.

2.4.3. *Reactor and particle temperature modelling*

We assume that each particle enters the reactor coil at 25 °C (room temperature). The particles are then heated up over time t, until they reach an energy threshold ρ, after which the probability of reaction increases. The rate of temperature increase δt of each particle p is governed by thermodynamic principles including heat transfer q (Eq. (37)), thermal conductivity Λ and properties of the carrying fluid ψ, **Newton's Law of Cooling** (Eq. (36)) and the concepts of **conduction**, **convection**, as well as **radiation**. In this work, we model the increase in temperature of each particle over time using Newton's Law of Cooling (Eq. (36)). According to the **Newton's Law of Cooling** the rate of change of the temperature of an object is proportional to the difference between its own temperature and the room temperature. Mathematically, this is expressed as:

$$\frac{dT}{dt} = -\epsilon(T - T_{\mathrm{env}}) \tag{36}$$

Where $\frac{dT}{dt}$ is the rate of change of the temperature of the object, T is the temperature of the object, T_{env} is the ambient temperature and ϵ is a constant that depends on the characteristics of the object and the environment such as surface area, heat transfer coefficient. This equation indicates that the greater the temperature difference $(T - T_{\mathrm{env}})$, the faster the rate of change in temperature. In the context of heat conduction, Fourier's Law states that the heat transfer rate q through a material is proportional to the temperature difference across the material, expressed as:

$$q = -\Lambda \cdot A \cdot \frac{dT}{dx} \tag{37}$$

Where q is the heat transfer rate, Λ is the thermal conductivity of the material, A is the cross-sectional area through which heat is transferred and $\frac{dT}{dx}$ is the temperature gradient across the material. The greater the temperature difference across the material, the higher the rate of heat transfer, resulting in a faster temperature change in the object. Since the principles of convection and radiation vary from reactor system to system depending on the materials used, we collectively model these effects as ζ, which is identified using a system characterisation process (see Section 2.4.5). When identifying ζ, we also incorporate Λ and ϵ.

$$\Delta p_T = \Lambda \Delta T \tag{38}$$

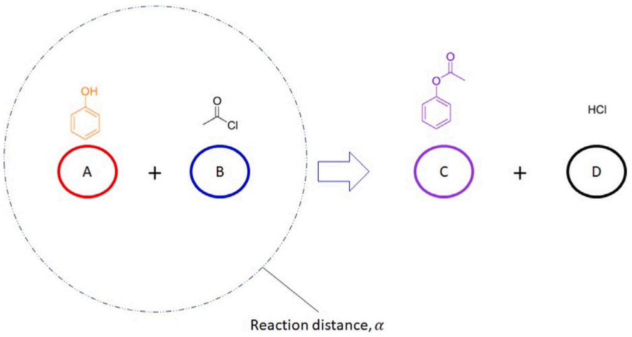


Fig. 6. Modelling concept of reaction distance α between two reactants within the chemical process.

2.4.4. *Kinetic model*

The design equation of the PFR is applied to characterise the concentration of each species in the system as a function of their residence time τ as described by Eq. (1). We used this as a macroscopic model, where each particle in our SPH framework models an ensemble of reactant or product "molecules". We embedded principles from molecular dynamics and stochastic reaction diffusion models into a SPH framework to model chemical reactions taking place while dealing with flow in the reactor geometry. This led to a hybrid modelling system which represents one of the key contributions in this manuscript. In the experimental section, we compare results obtained from this hybrid modelling approach with real-life experiments and physics informed neural network (PINN) models. The reaction distance α for the chemical reagents modelled in this work is defined by Eq. (39) [45,46] and is illustrated in Fig. 6.

$$\alpha = \frac{k}{4\pi(D_A + D_B)} \tag{39}$$

Where D_A and D_B are diffusion constants of reactants A and B respectively and k is the reaction rate constant. Depending on the reaction rate constant of the chemical reaction, the reaction distance changed accordingly. Residence time was calculated according to Eq. (40) where V is the volume of the reactor and Q is the volumetric flow rate. The flow rate in the reactor section was determined as the sum of the flow rates delivered by pumps A and B.

$$\tau = \frac{V}{Q} \tag{40}$$

Each particle's position (X) was influenced by the system of SPH equations (Eq. (27) to (35)), as follows:

$$X_{t+1} = X_t + v \tag{41}$$

When the distance between two particles of reactants A and B falls below α and the particle temperature exceeds the energy threshold constant for A (and respectively for B), a reaction occurs according to the probability $\sim U(0,1) < k$. When this happens, the two products C and D replace the reactants A and B in our system for the real-life experiments we were digital twinning.

2.4.5. *System characterisation through optimisation*

Due to the differences in the scales of the systems under consideration (microscopic to macroscopic), an appropriate optimisation was required to characterise the physical system that our digital twin was attempting to model. This involved discovering the values for the variables of the system including those governing convection, radiation and heat loss, modelled as ζ, as previously discussed. In order to determine ζ, we made use of Genetic Algorithm (GA), a population based algorithm with capability for handling: (1) multi-variable value discovery and optimisation, (2) optimisation problems in large and complex search spaces, (3) problems where a simple analytical solution

Table 2
Hyperparameters implemented for Reaction Model 1 using the PINN framework in forward mode at 65 °C, applied to both full and reduced datasets. As seen in the Table, the number of neurons for the reduced dataset was larger than that for the full dataset. This is because we used the same number of epochs in both cases thereby providing an opportunity for the grid search algorithm to find a complex model (in the form of increased number of neurons) that overfits to the reduced dataset.

Parameters	Full dataset	Reduced dataset
Layers	5	5
Neurons	8	10
Learning rate	0.001	0.001
Activation function	tanh	tanh
Optimiser	Adam	Adam

is not readily available, (4) data of any dimensionality, and (5) noisy or incomplete datasets by exploring multiple potential solutions [47,48]. Furthermore, the parallel nature of the GA framework allows for faster computations on multi-core systems which is very useful when handling large computational loads. These features are unlike ordinary gradient descent algorithm which also have the issue of getting trapped in local minima. Towards finding ζ, we minimised the differences between the data generated by our chemically informed SPH and the data obtained from physical experiments. A comprehensive description of the steps involved in this process, along with the configuration of the Genetic Algorithm, can be found in Section 3.0 of the Supplementary Information (SI). The pseudocode outlining the Genetic Algorithm for scaling factor optimisation is provided as Algorithm 3 in the Supplementary Information.

3. Results and analysis

The training process for the PINN involves constructing a comprehensive dataset composed of collocation points within the domain. This dataset includes a specific number of residual points sampled within the domain and boundary points sampled at the left and right endpoints of the interval. The complete training dataset is formed by integrating domain data and boundary data, providing the neural network with a thorough understanding of the solution's behaviour within the defined domain. The testing dataset, used to evaluate the model's performance, includes a combination of randomly sampled points within the domain, experimental data points, and initial condition points. The entire dataset is then split into 70% for training and 30% for testing, ensuring a balanced approach to model training and validation.

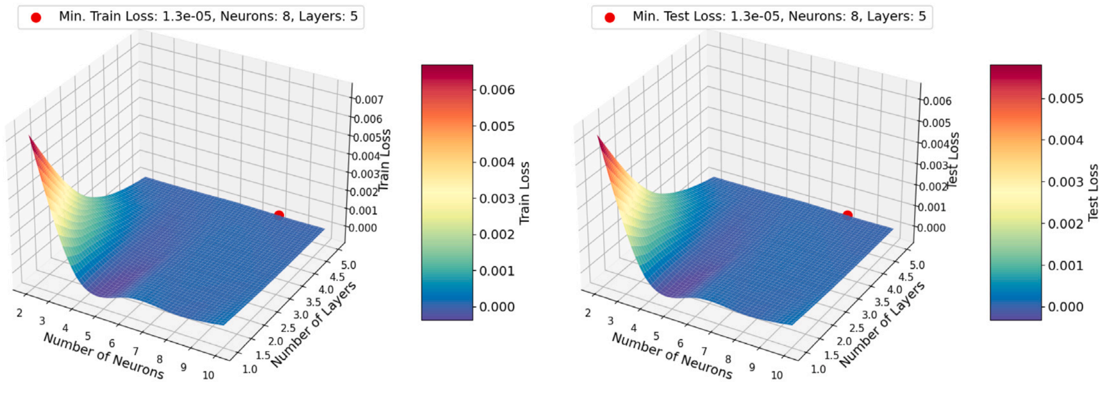
3.1. PINN forward mode for output prediction (model 1 at 65 °C)

The hyperparameter tuning for Model 1 was carefully explored in structured collocation points under various configurations. Specifically, the number of hidden layers and neurons was varied to evaluate their impact on model performance. Each configuration was trained for 10,000 epochs using the Adam optimiser with a learning rate of 0.001. Key training metrics, including training loss, test loss, and training time, were recorded for comprehensive performance evaluation. The detailed results of these hyperparameter configurations, along with the corresponding loss and prediction plots, are presented in Table S1 and Table S2 in the Supplementary Information (SI). The grid search conducted on the full dataset identified the optimal hyperparameter configuration as 5 hidden layers with 8 neurons per layer. This setup achieved the lowest training and test losses of 1.34×10^{-5} and 1.28×10^{-5}, respectively. Similarly, for the reduced dataset, the optimal configuration was also found to be 5 hidden layers, but with 10 neurons per layer, resulting in training and test losses of 1.39×10^{-5} and 1.31×10^{-5}, respectively. The surface plots show the relationship between the number of neurons, layers, and training loss, with the lowest point indicating the optimal configuration as visually represented in Fig. 7 for the full dataset and Fig. 8 for the reduced dataset. The result of the optimal hyperparameters for the PINN model are detailed in Table 2.

The loss curves in Figs. 9(a) and 9(c) illustrate the total training and testing loss $\{L_{\text{Total}}\}$ which are $\{L_{\text{ODE}}, L_{\text{BC}}, L_{\text{IC}}, L_{\text{data}}\}$ over the course of 10,000 steps. The area plot shown in Figs. 9(b) and 9(d), particularly for the input reagent and the desired product, visually highlights the narrowing error margins as training progresses, emphasising the increasing precision of the model predictions. Both losses decrease consistently, with the test loss closely following the training loss, indicating strong generalisation and minimal overfitting. The final losses converge to a magnitude of approximately 10^{-5}, demonstrating that the model has achieved a high degree of accuracy in its predictions. In the forward mode PINNs are applied to simulate the kinetics of a reactive system involving the species A, B, C, and D. Experimental data are available only for species A and C, and the initial concentrations of A and B are known. The reaction rate constant, k_1 is also known and treated as a fixed parameter in the model. The network takes domain coordinates as input and outputs the predicted concentrations of the chemical species. The residuals of 4 chemical species ODEs are computed at a set of collocation points throughout the domain that are incorporated into the total loss function using Automatic Differentiation (AD) during the training process. PINNs utilise AD to compute the derivatives of the approximated concentration functions. This permits the ODE residuals to be evaluated at the domain collocation points without discretisation the solution on a temporal grid. When trained with the full dataset, the network benefits from comprehensive observational information, which complements the embedded ODEs. As shown in Fig. 10(a) we observed a high correlation between the concentration profiles of phenol and phenyl acetate predicted by the PINN model, the numerical solution of the ODE, and the experimental data. For the full dataset, the RMSE values were as low as 0.0019 and 0.0018 for phenol and phenyl acetate, respectively, at 65 °C, with R^2 scores exceeding 0.99 in both cases, as depicted in Table 4. Remarkably, even when the observational data is reduced and active only in the first 0–6 min interval, the physics-informed component drives the network to reconstruct accurate full-domain concentration profiles. Data in the early region helps guide the network's initial learning, while the governing equations enable it to generalise and extrapolate the solution to the remaining time span while maintaining the correct trends in unobserved regions (6–15 min). As evidenced in Fig. 10(b), the prediction maintained high accuracy even for the reduced dataset, with RMSE values below 0.0027 and R^2 scores above 0.972 demonstrating the robustness of the PINN model under conditions of limited experimental data.

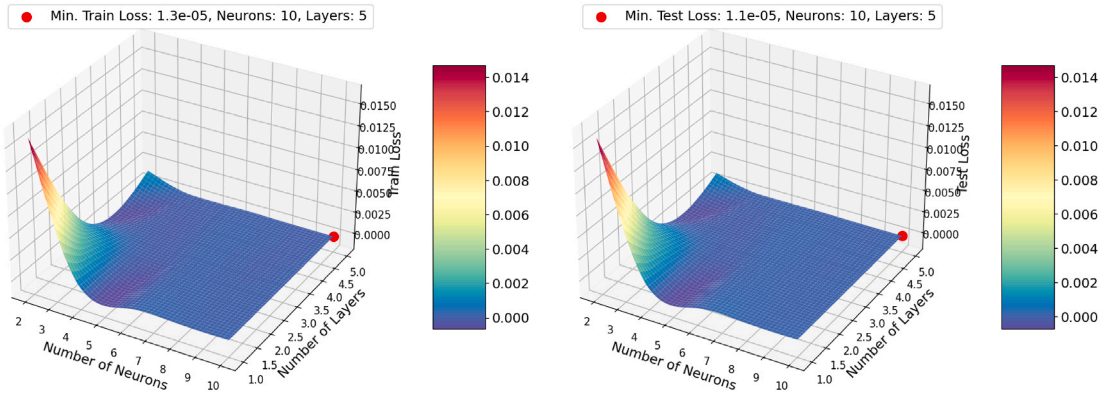
3.2. PINN inverse mode for simultaneous output prediction and parameter estimation (model 1 at 75 °C)

We also applied the PINN framework in inverse mode to simultaneously estimate the unknown reaction rate constant k_1 and predict a full-domain concentration profiles of the reaction model. The setup remains the same as in the forward mode, involving four species (A, B, C, and D), where experimental concentration data are available only for species A and C, and the initial concentrations of A and B are known. Unlike in PINN forward mode where kinetic parameters are predefined, in PINN inverse mode, the model must infer k_1 directly from the experimental observations, while still satisfying the system's ODEs. We adopt the same PINN architecture and optimal hyperparameters used in the PINN forward mode (Table 2). Using consistent architecture across forward and inverse modes ensures comparable model behaviour, simplifies hyperparameter tuning, and enables potential reuse for transfer learning (TL). The mechanism underlying inverse PINNs involves defining unknown physical parameters explicitly as trainable variables alongside the neural network's conventional weights and biases. In this setup, k_1 is treated as an optimisable parameter and the solution to the inverse problem is obtained by jointly optimising both NN parameters, θ and k_1 as shown in Eq. (26), where lambda



(a) (b)

Fig. 7. Surface plot of hyperparameter tuning. The red circle indicates the optimal network configuration using the full dataset. (a) Training loss (b) Testing loss.

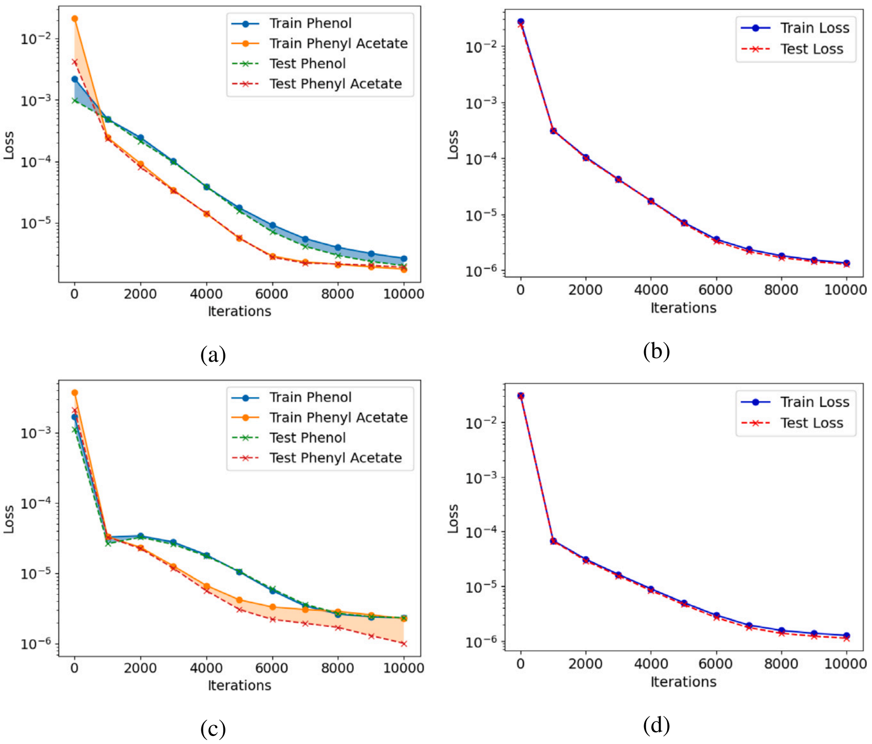


(a) (b)

Fig. 8. Surface plot of hyperparameter tuning. The red circle indicates the optimal network configuration using the reduced dataset. (a) Training loss (b) Testing loss.

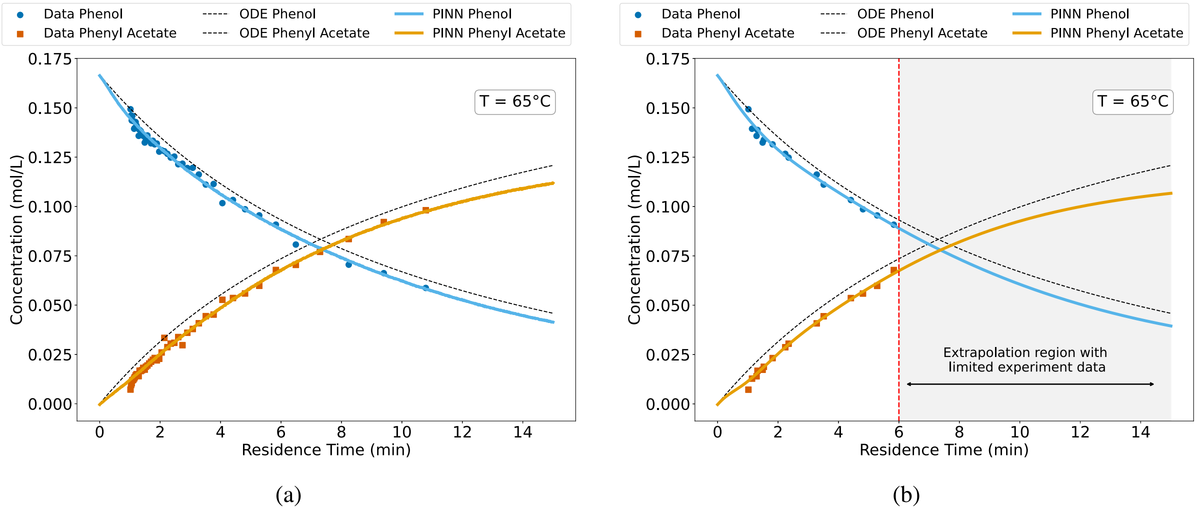
represents as k_1. When training begins, the unknown parameter k_1 is first initialised with a random or approximate value. Through the process of joint optimisation, both parameters, θ and k_1 are updated simultaneously at each iteration using a gradient-based optimiser such as Adam. AD enables the computation of gradients of the total loss function with respect to both parameters. This joint optimisation process allows the model to minimise the total loss function, as expressed in Eq. (25), by fitting the network to observed concentration data while also identifying the value of k_1 that best satisfies the governing ODEs. Once the training converges, the optimised parameters θ^* define a smooth and differentiable function that accurately approximates the concentration profiles of all four species (A, B, C, and D) across the domain. Simultaneously, the inferred value k_1^* represents the reaction rate constant that best aligns the model predictions with both the experimental data and the underlying reaction mechanism. Figs. 11 and 12 illustrate the performance of Physics-Informed Neural Networks (PINNs) in predicting the concentration profile and estimating the reaction rate constant k_1 using empirical datasets of different sizes. In both cases, the PINN model effectively estimated the k_1 value, as illustrated in Figs. 11(b) and 12(b), which show the convergence of the estimated k_1 values towards the true reaction rate constant. In the scenario using the full dataset, the predicted k_1 value converges to approximately 10.404×10^{-3} $\mathrm{M^{-1}s^{-1}}$ closely matching the true value of 10.45×10^{-3} $\mathrm{M^{-1}s^{-1}}$. This high level of agreement underscores the model's capability to leverage comprehensive data to produce precise parameter estimates. Conversely, when the dataset is reduced to one-third of its original size, the PINN model still provides a reasonable estimate of the rate constant, with a predicted k_1 value of 10.31×10^{-3} $\mathrm{M^{-1}s^{-1}}$. While this estimate is slightly lower than that obtained from the full dataset, the minimal difference highlights the robustness and effectiveness of the PINN model in parameter estimation even when the available data is limited. Figs. 11(a) and 12(a) further support these findings by displaying the concentration profiles predicted by the PINN model when trained on the full and reduced datasets, respectively. The predicted concentrations align closely with the experimental data in both cases, following the trend of the numerical solution. This close alignment indicates that the PINN model can accurately capture the system's dynamics, even when trained on a deliberately reduced dataset consisting of one-third of the original data points. These data were specifically sampled from the 0–5 min residence time interval allowing the model to utilise its inherent knowledge of the governing equations. By addressing challenges posed by missing or limited data, the model effectively generalises and predicts the system's behaviour beyond the training region, despite excluding observational data from potentially more informative regions of the reaction dynamics later in the timescale. The comparison of the predicted k_1 values between the full and limited datasets, is shown in Tables 3.

To evaluate the performance of the PINN model in predicting the concentrations of phenol and phenyl acetate at 65 °C and 75 °C, we conducted a series of comparative analyses between the model's predictions and two different experimental datasets. The accuracy of the model was quantified using the coefficient of determination (R^2) and the Root Mean Square Error (RMSE). High R^2 values close to 1.0 and low RMSE values indicate a strong correlation between the predicted and actual concentrations, demonstrating the model's precision and reliability. These metrics were computed for both phenol and phenyl



(a) (b) (c) (d)

Fig. 9. Showing the training and testing loss curves for Model 1 using full and reduced dataset. These curves are for the optimal values of the hyperparameters (See Table 2) obtained during grid search. ***For Model 1 with full dataset:*** (a) Shows loss curves and area plot, L_{ODE} for C_A (phenol) and C_B (phenyl acetate) while (b) Shows the overall training and testing loss, L_{Total} across all iterations. ***For Model 1 with reduced dataset:*** (c) Shows loss curves and area plot t, L_{ODE} for C_A (phenol) and C_B (phenyl acetate) while (d) show the overall training and testing loss, L_{Total} across all iterations.



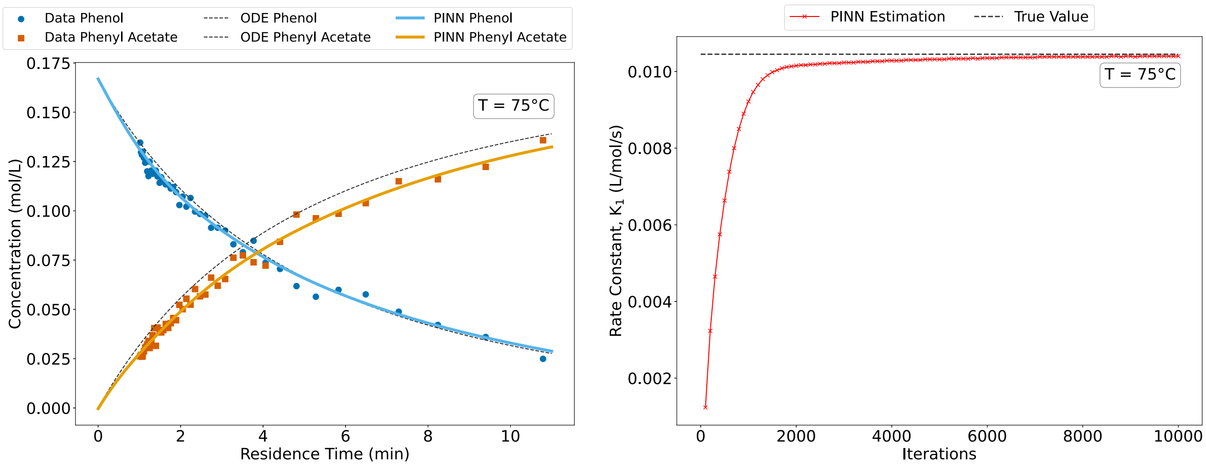
(a) (b)

Fig. 10. Model 1: Comparison of concentration profiles predicted by PINN and ODE solutions against experimental data for phenol and phenyl acetate at 65 °C: (a) full dataset, (b) reduced dataset.

Table 3
Model 1 at 75 °C: Comparison of the true rate constant k_1 and the PINN estimates with full and reduced datasets, showing close agreement in both cases.

Parameter	Rate constant value, k_1 ($M^{-1}s^{-1}$)
True value	0.01045
PINN estimation (full dataset)	0.01040
PINN estimation (reduced dataset)	0.01031

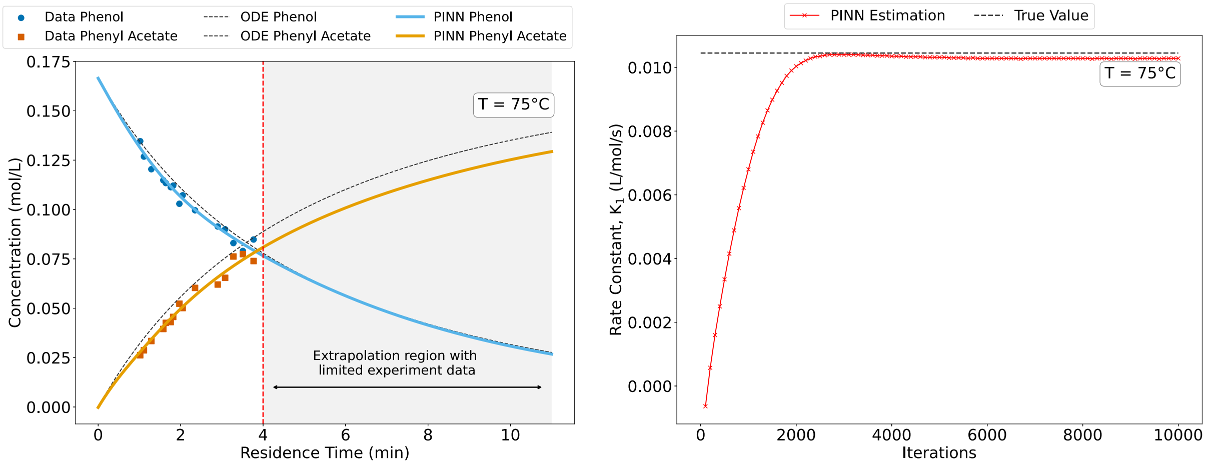
acetate at each temperature, allowing us to determine how well the PINN model performs in replicating experimental results under both complete and limited data scenarios. Fig. 13 illustrates the results obtained using the complete original dataset. The plots show a strong correlation between the PINN model predictions and the experimental data for both phenol and phenyl acetate. This is evidenced by the near-unity R^2 values and the low RMSE values for each case, indicating the model's high accuracy in predicting the concentrations at both temperatures. The data points closely align with the red dashed line, which represents perfect agreement between the predicted and actual values, further confirming the model's robustness. Fig. 14 compares the same predictions but utilises a reduced dataset for the analysis. Despite the reduction in data, the PINN model maintains a high degree of accuracy, as shown by the slight increase in RMSE values and minor



(a)

(b)

Fig. 11. Model 1 using the full dataset: (a) PINN-predicted concentration profiles showing strong agreement with experimental data, and (b) evolution of PINN identification of rate constant k_1 over the iterations.



(a)

(b)

Fig. 12. Model 1 using the reduced dataset: (a) PINN-predicted concentration profiles showing strong agreement with experimental data, and (b) evolution of PINN identification of rate constant $k_1$over the iterations.

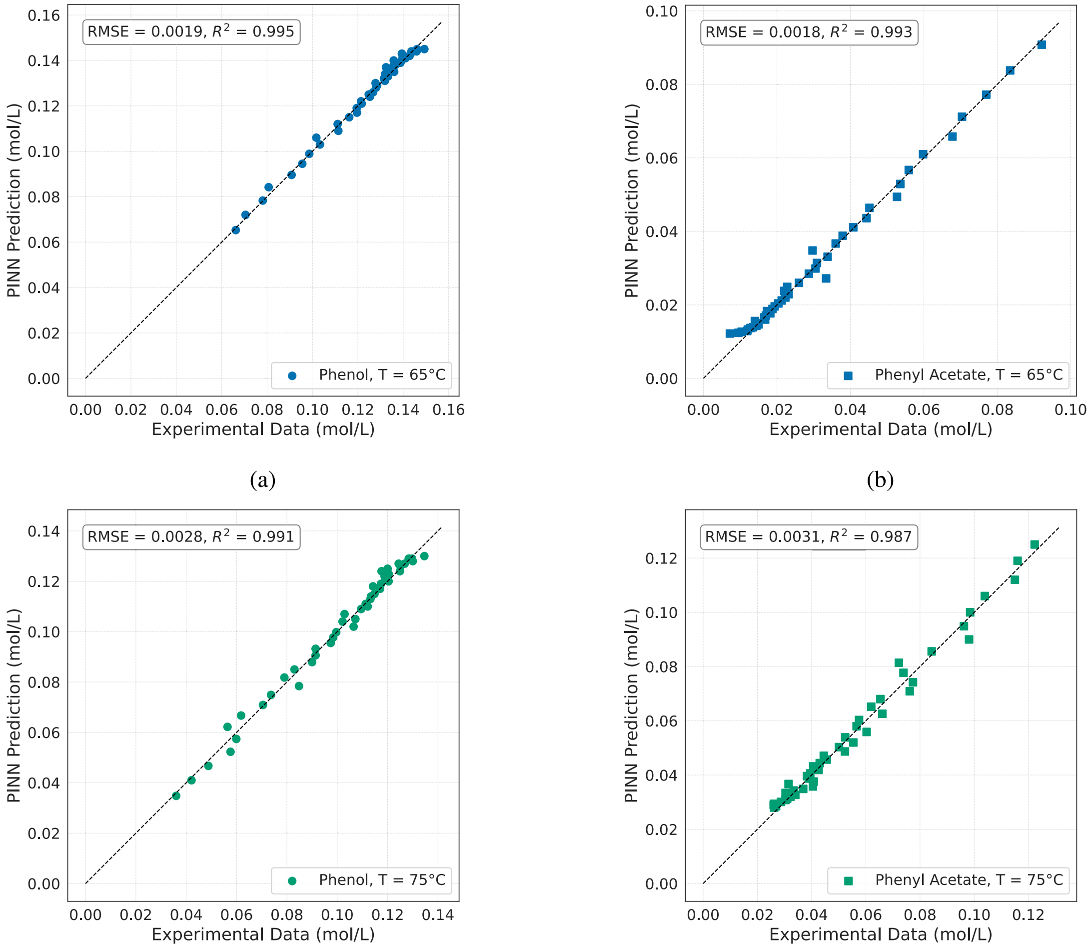
deviations in R^2 values. The results suggest that the model can still reliably predict the concentrations with limited data, although with marginally reduced precision. Table 4 summarises the RMSE and R^2 values for phenol and phenyl acetate at both temperatures. The results indicate that, while the model performs slightly better with the full dataset, the reduced dataset still yields acceptable accuracy, underscoring the robustness of the PINN model in both scenarios. In Model 1, we predicted the concentration profiles and identified the reaction rate constant k for a simple one-step reaction. This model was characterised by a consistent set of system parameters, such as the same reactor volume and flow rate, which resulted in a uniform residence time. Additionally, the initial conditions were identical for both temperature conditions (65 °C and 75 °C), as detailed in Table 2. This uniformity allowed us to isolate and understand the behaviour of a straightforward reaction under controlled conditions, focusing specifically on the effect of the temperature on the reaction kinetics.

3.3. Application of a chemically-informed SPH+GA series hybrid modelling framework

Table 5 outlines the Genetic Algorithm (GA) hyperparameters used to optimise the scaling factor (ζ) within the chemically informed Smoothed Particle Hydrodynamics (SPH) framework. The implemented GA settings included a population size of 100, 100 generations, and a mutation rate of 0.2. The optimal scaling factors obtained were 0.000453 for Model 1 at 65 °C, 0.000481 for Model 1 at 75 °C, and 0.881377 for Model 2 at 190 °C. These scaling factors were applied to tune the thermal and kinetic behaviour of the SPH model, ensuring closer alignment with experimental concentration data. The GA convergence behaviour, shown in Supplementary Information (SI) Figure S5, demonstrates rapid mean squared error (MSE) reduction within the first 10 generations, confirming efficient parameter optimisation. To achieve this, a structured GA configuration was adopted. The algorithm employed a roulette wheel selection method, single-point crossover, and mutation with a small random perturbation to maintain diversity and avoid premature convergence. A fitness function based on the sum of absolute differences between scaled SPH predictions and experimental data guided the optimisation, favouring scaling factors that minimised this error. This approach ensured that the SPH model could be calibrated effectively with minimal reliance on physical experimentation, helping to reduce experimental cost and time. Nevertheless, strategically selected experimental points were still required to inform the optimisation and validate the digital twin model's predictions.

Table 4
Performance Metrics (RMSE and R^2) of the PINN model for predicting Phenol and Phenyl Acetate concentrations at different temperatures and dataset sizes.

Species	PINN mode	Temperature	RMSE	R^2 scores	RMSE	R^2 scores
			(Full dataset)		(Reduced dataset)	
Phenol	Forward	65 °C	0.0019	0.995	0.0021	0.986
Phenol	Inverse	75 °C	0.0028	0.991	0.0026	0.974
Phenyl Acetate	Forward	65 °C	0.0018	0.993	0.0015	0.994
Phenyl Acetate	Inverse	75 °C	0.0031	0.987	0.0027	0.972



(a) (b)

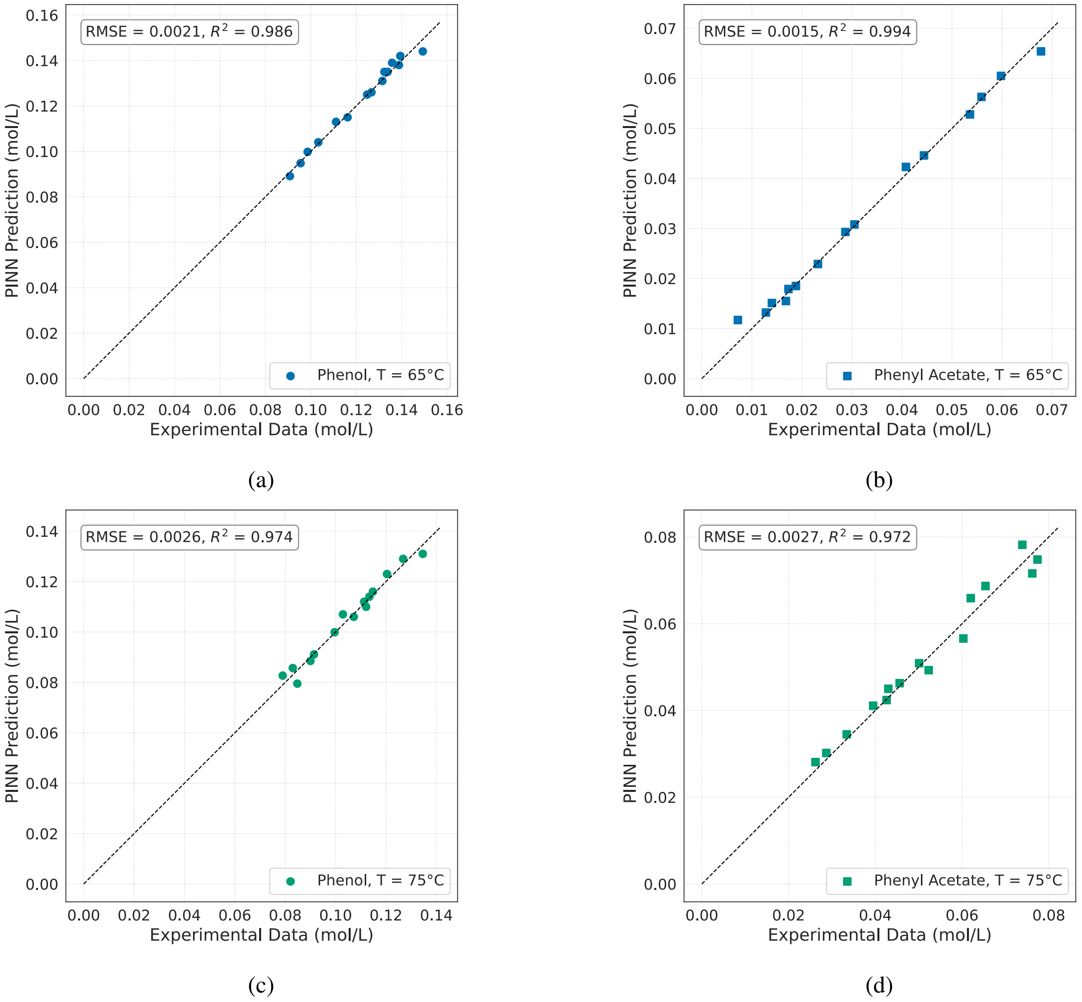
(c) (d)

Fig. 13. Parity plots showing PINN-predicted versus experimentally measured concentrations for (a) phenol at 65 °C, (b) phenyl acetate at 65 °C, (c) phenol at 75 °C, and (d) phenyl acetate at 75 °C. Results are based on the full experimental dataset. The associated RMSE and R^2 values quantify predictive accuracy of the model.

Further implementation details, including the full procedural logic and parameter settings, are presented in Section 3.0 of the Supplementary Information (SI), with pseudocode outlined in Algorithm 3. This includes the use of two input datasets (SPH raw simulation data and experimental data), initialisation of a population of scaling factors, fitness evaluation via MSE, parent selection, crossover, mutation, and iterative replacement of poorly performing candidates. This SPH+GA approach exemplifies a series hybrid modelling framework, in which physics-based simulation (SPH) is sequentially enhanced through data-driven parameter tuning (GA). This hybridisation strengthens the model's predictive fidelity and adaptability across various reaction systems and temperature regimes. Fig. 15 illustrates the resulting model performance, with sub-figures (a), (c), and (e) showing concentration profiles for phenyl acetate (Model 1) and metoprolol (Model 2) compared to experimental data. Sub- Figs. 15(b), (d), and (f) present parity plots that quantitatively assess the model's accuracy using RMSE and R^2 metrics. Together, these results demonstrate the effectiveness of the SPH+GA hybrid model in reproducing complex chemical dynamics. Table 6 complements this analysis by comparing performance across PINN and SPH+GA models. While PINNs exhibit superior accuracy, particularly in data-scarce settings, the SPH+GA framework provides critical insights into flow and transport dynamics, making it a valuable complementary tool for digital twinning of chemical processes.

3.4. PINN inverse mode for simultaneous output prediction and parameter estimation in two-step reaction model

In this section, we analysed the performance of the PINN model to describe the kinetics of a more complex reaction network. Model 2 introduces additional complexity by requiring the simultaneous estimation of two rate constants, k_2 and k_3 while also predicting the concentration profiles of the species involved in the system. This model



(a)

(b)

(c)

(d)

Fig. 14. Parity plots showing PINN-predicted versus experimentally measured concentrations for (a) phenol at 65 °C, (b) phenyl acetate at 65 °C, (c) phenol at 75 °C, and (d) phenyl acetate at 75 °C. Results are based on the incomplete/limited experimental dataset. The associated RMSE and R^2 values quantify predictive accuracy of the model.

Table 5
GA hyperparameters implemented.

Parameters	Value
Population Size	100
Number of Generations	100
Mutation Rate	0.2
Best scaling factor (Model 1, 65 °C)	0.000453
Best scaling factor (Model 1, 75 °C)	0.000481
Best scaling factor (Model 2, 190 °C)	0.881377

Table 6
Performance metrics (RMSE and R^2) for PINN and SPH models at different temperatures for Models 1 and 2.

Model	Reaction system	Temperature (°C)	RMSE	R^2
PINN	Model 1	65	0.0018	0.993
	Model 1	75	0.0030	0.987
	Model 2	190	0.0033	0.988
SPH	Model 1	65	0.0114	0.758
	Model 1	75	0.0122	0.822
	Model 2	190	0.0090	0.912

was designed to evaluate the PINN's ability to simultaneously predict multiple rate constants within a complex reaction network. The introduction of changes in reactor volume, residence time, and initial concentration across different temperatures added further complexity, making it more challenging for the model to accurately capture the system's kinetic behaviour. We evaluated Model 2 with PINN in Inverse mode using the same optimal hyperparameters reported in Table 7, which were previously used for Model 1. However, due to the limited experimental data available for Model 2, we did not perform further dataset reduction as in the case of Model 1. We evaluated Model 2 at four temperatures of 130 °C, 150 °C, 190 °C and 210 °C. The PINN model was used to predict the species output and the same time identify the unknown two rate constants, k_2 and k_3, within a reaction network involving two consecutive reactions, described by Eqs. (7) and (8).

Fig. 16 illustrates the convergence of these values across the varying temperature conditions, further validating the model's accuracy. Table 8 provides a comparison of the experimental and PINN-estimated rate constants k_2 and k_3 across different temperatures, showing their close agreement. The figure illustrates PINN predictions of species concentrations and estimation of rate constants of k_2 and k_3 across varying temperatures. The results validate the PINN model by comparing the predicted concentration profiles of the reactants and products with the experimental data. The concentration profiles in Fig. 16 (a, d, g, j) demonstrate that the predicted concentrations align closely with both

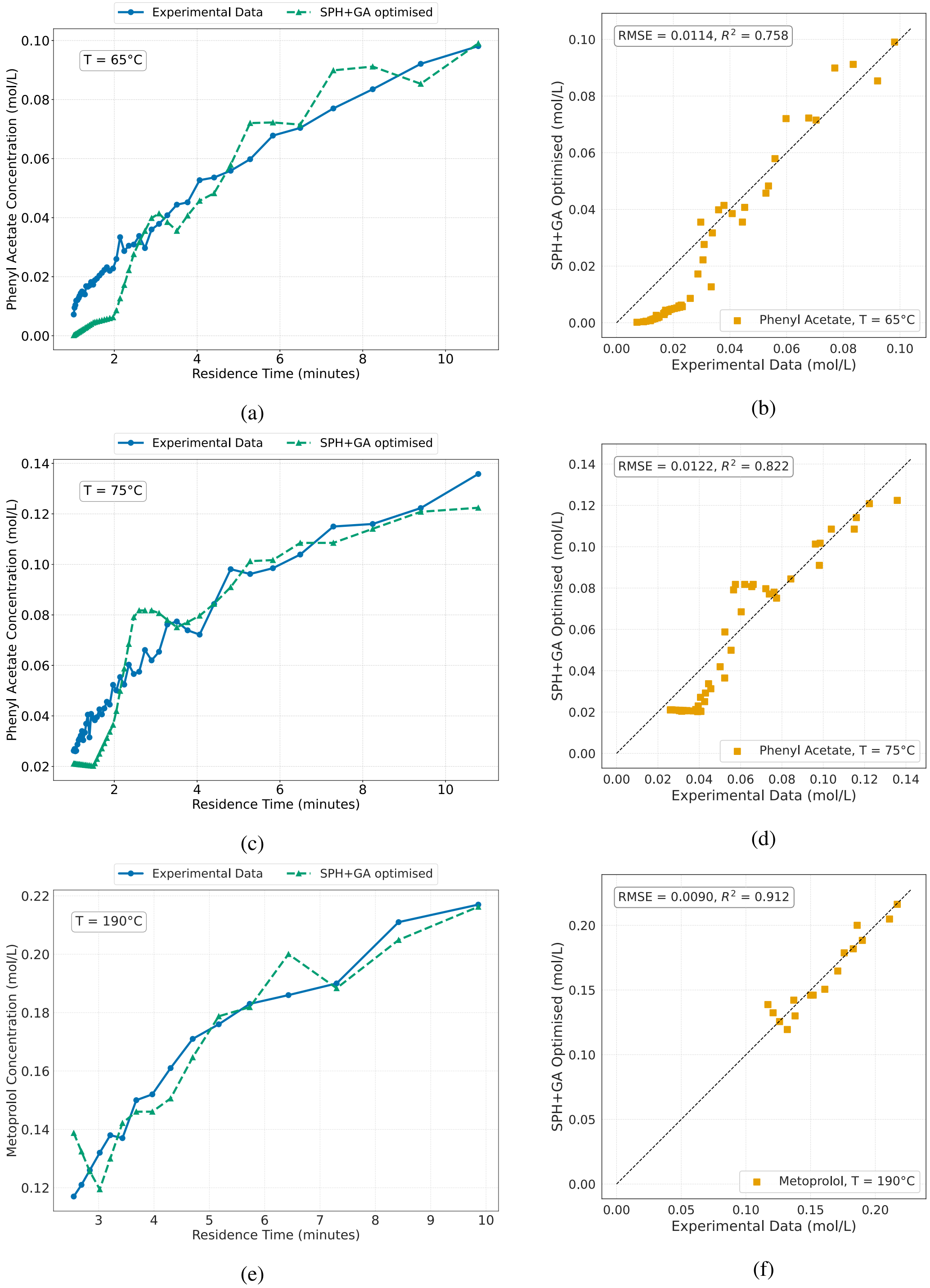
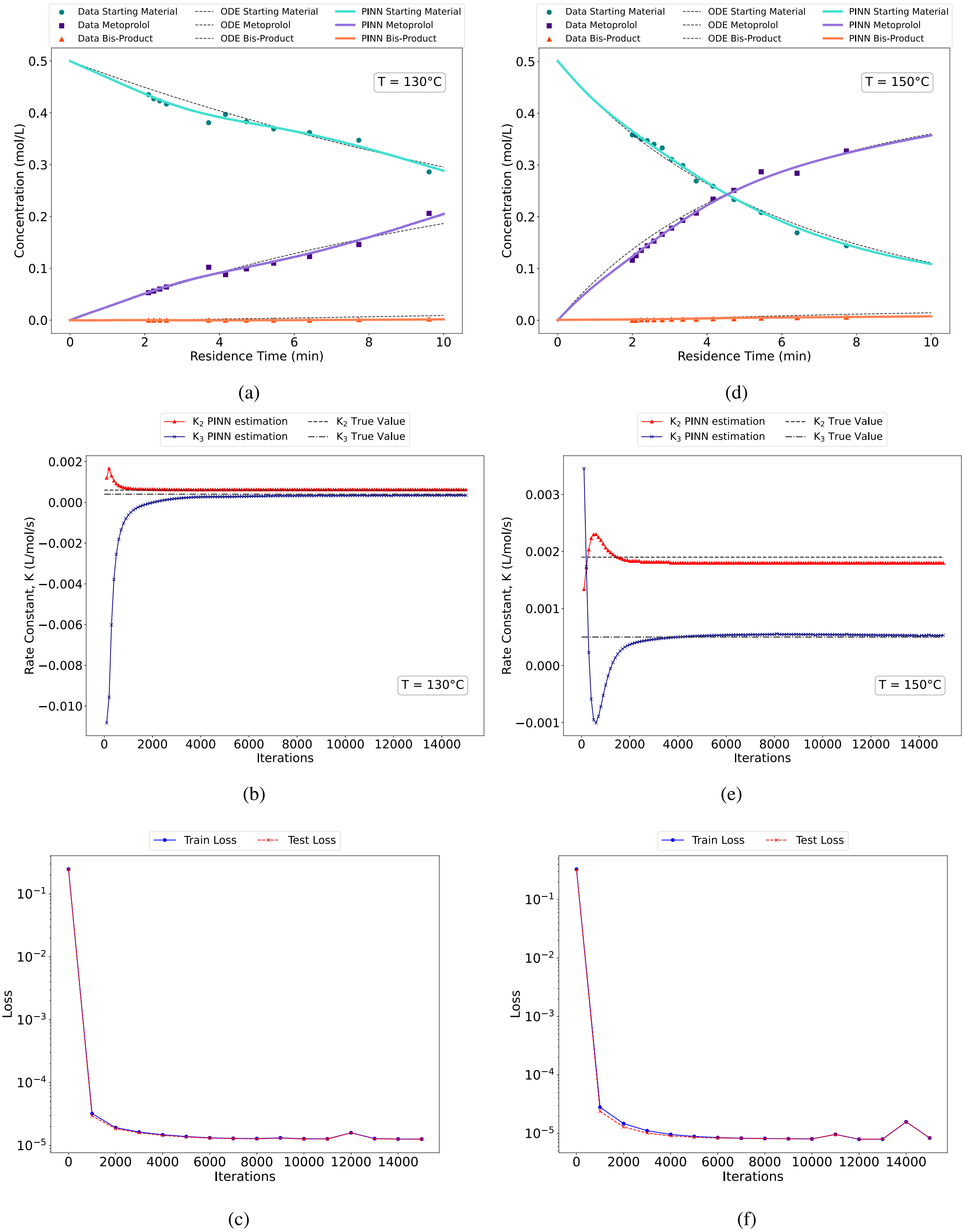


Fig. 15. Comparative evaluation of SPH+GA model predictions against experimental data across three reaction conditions. Sub-figures (a, c, e) show the temporal concentration profiles predicted by the SPH+GA model alongside experimental measurements for phenyl acetate (Model 1) at 65 °C and 75 °C, and metoprolol (Model 2) at 190 °C. Sub-figures (b, d, f) present parity plots assessing model performance, highlighting the model correlation and predictive capability of the SPH+GA approach under varying operating conditions.

the experimental data and trends observed from its ODEs solutions, confirming the model's accuracy in capturing the dynamic behaviour of the reaction network even with sparse data. The estimated values of rate constant k_2 and k_3 show n Fig. 16 (b, e, h, k) progressively converge towards the experimental values as iterations progress shows effective convergence towards true values, with faster convergence observed at higher temperatures. Training and test loss curves in Fig. 16 (c, f, i, l) further confirm the consistency of the model's performance across iterations. These results collectively validate the PINN model's accuracy in capturing the reaction network's dynamics, confirming the model's accuracy in capturing the dynamic behaviour of the reaction network. Its consistent performance across varying temperatures highlights the potential of the PINN model as a powerful tool for analysing complex reaction networks in chemical engineering. Model 2 further exemplifies the ability of PINNs to estimate multiple rate constants and predict concentration profiles in systems involving consecutive reactions. Despite the limited experimental data, the model's ability to generalise and provide accurate predictions underscores the robustness of the selected hyperparameters, making it applicable to a range of reaction conditions.



(a) (d)

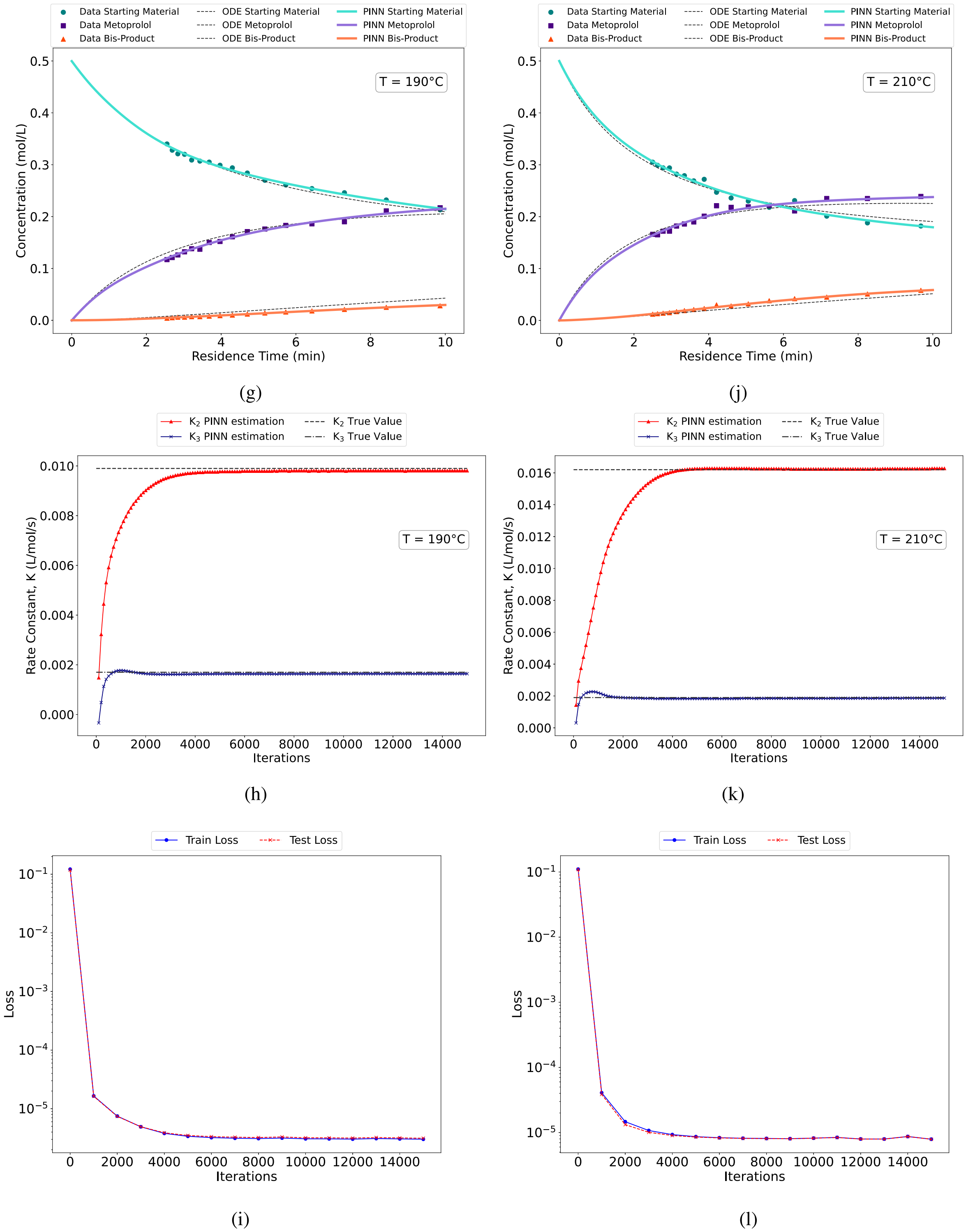
(b) (e)

(c) (f)

Fig. 16. PINN predictions for Model 2 (Metoprolol) at 130 °C (a–c), 150 °C (d–f), 190 °C (g–i), and 210 °C (j–l). At each temperature, the concentration profiles prediction (a, d, g, j), evolution of k_2 and k_3 identification over iterations (b, e, h, k) and training and test loss curves (c, f, i, l).

4. Discussion

As seen from the results above, both PINN and SPH are highly effective strategies for modelling flow chemistry particularly in the context of mixing. However, with an increasing number of particles, SPH rapidly becomes computationally expensive, requiring the use of Graphical Processing Units (GPUs) to run simulations effectively. In order to compensate for this, it is possible to use a single particle to represent an ensemble of particles. Furthermore, SPH excels at capturing complex reactor geometries as well as multiphase mixing, aspects that are not explicitly represented or defined in PINNs. As a result, in contrast to PINN, SPH offers better scalability as reactor sizes or process complexity increases. On the other hand, PINNs are well-suited for rapidly testing out changes in reactor geometry as long as the governing equations are known. For a similar change, the new reactor geometry would need to be digitally constructed and then converted



(g) (j)

(h) (k)

(i) (l)

Fig. 16. (*continued*).

into boundary particles for SPH simulations, thereby taking longer to start the simulation.

While plug flow reactors (PFRs) are traditionally modelled using systems of ordinary differential equations (ODEs) with known reaction kinetics, recent developments in reaction engineering have significantly increased the complexity of kinetic models. Mechanistic microkinetic models now routinely involve hundreds to thousands of elementary reaction steps and intermediates [49–51]. These reaction networks are often generated automatically through rule-based algorithms that classify reactions into families based on chemical knowledge [50,52]. Though it is theoretically possible to solve these large stiff systems with conventional numerical solvers, integrating them into advanced simulation platforms such as computational fluid dynamics (CFD) frameworks or digital twin environments presents major computational challenges.

In this study, we explore the use of Physics-Informed Neural Networks (PINNs) and Smoothed Particle Hydrodynamics coupled with Genetic Algorithm optimisation (SPH+GA) as hybrid modelling strategies. These approaches enable the direct embedding of physical laws into the learning process or simulation framework, allowing for scalable, efficient prediction of system behaviour even when kinetics are

Table 7
Hyperparameters and model performance metrics for the two-step reaction (Model 2) using the PINN framework in inverse mode, evaluated at different temperatures.

Parameters	130 °C	150 °C	190 °C	210 °C
Layers	5	5	5	5
Neurons	10	10	10	10
Learning rate	0.001	0.001	0.001	0.001
Activation function	tanh	tanh	tanh	tanh
Optimiser	Adam	Adam	Adam	Adam
Iterations	15,000	15,000	15,000	15,000
Training loss	1.41e−04	8.64e−05	3.28e−05	8.70e−05
Testing loss	1.40e−04	8.67e−05	3.32e−05	8.65e−05
Training time (s)	42.3907	36.3376	53.5021	43.5168

Table 8
Model 2: Comparison of experimental and PINN-estimated rate constants k_2 and k_3 at different temperatures.

Temperature	True value		PINN estimation	
	k_2 ($M^{-1}s^{-1}$)	k_3 ($M^{-1}s^{-1}$)	k_2 ($M^{-1}s^{-1}$)	k_3 ($M^{-1}s^{-1}$)
130 °C	0.0006	0.0004	0.0006	0.00035
150 °C	0.0019	0.0005	0.0018	0.00014
190 °C	0.0099	0.0017	0.0095	0.00140
210 °C	0.0162	0.0019	0.0147	0.00186

only partially known or when sparse experimental data are available. By demonstrating their application on a model reaction system, we illustrate the potential of PINNs and SPH+GA to bridge the gap between fundamental mechanistic modelling and practical reactor simulation, particularly in the context of digital twin development for chemical processes.

The evaluations across the case studies demonstrated that the PINN model achieved high predictive accuracy, with coefficient determination, (R^2) of 0.99 and RMSE values as low as 0.0018 for phenol and phenyl acetate concentrations under both full and reduced datasets. In inverse mode, the model also accurately estimated the values of the kinetic rate constants with errors within 1%–2% of the true values. SPH+GA framework similarly showed strong predictive performance, achieving R^2 scores up to 0.91 and RMSE values around 0.009 at 190 °C. Additionally, it demonstrated flexibility in capturing complex particle-level dynamics and enabled effective parameter optimisation through GA, with convergence achieved within 10 generations. Due to the SPH+GA series hybridisation architecture, this approach ensures generalisation across physical systems. By collecting experimental data while running simulations, we are able to perform both system identification of the parameter space towards capturing intrinsic parameters that the simulations are missing while at the same time performing tuning of the parameters that have been captured. This ensures that the SPH+GA approach can be used to predict regions of the experimental space not captured by the experimental dataset.

These results demonstrate that PINNs are highly effective in predicting reaction rate constants and concentration profiles, while solving forward problems to predict system behaviour and inverse problems to infer unknown parameters in the governing ODEs, even with sparse experimental data. Meanwhile SPH enhances the simulation of fluid dynamics and mixing processes at a granular level. The SPH model, while effective to some extent, demonstrates higher prediction errors and weaker correlations with experimental data, as evidenced by the evaluation metrics in Table 6 and Fig. 15. This might be due to the SPH model's reliance on particle-based methods, which may not fully capture complex chemical kinetics or interactions in the system, particularly at lower concentrations or early reaction stages. The PINN model benefits from incorporating physical laws (governing equations) directly into the neural network, which allows it to maintain strong predictive accuracy even with limited data. Its ability to solve both forward and inverse problems while adhering to physical constraints makes it a more robust model for complex chemical systems. Overall, the findings of this research demonstrate that digital twin created using physics-informed mechanisms can significantly enhance the design, optimisation, and control of chemical reactors. This approach offers promising directions for the development of advanced computational tools that combine the strengths of both machine learning and physics-based models thereby enabling more robust, efficient and accurate simulations of industrial processes.

Another approach gaining traction is the use of data-driven transformer based machine learning approaches. However, transformer models remain predominantly data-driven which typically require large and high-quality datasets to achieve robust predictive performance. In scenarios of sparse or incomplete data, which is common within chemical synthesis, transformers can struggle to generalise accurately or extrapolate reliably to untested conditions. To mitigate these limitations, future research should explore integrating transformer architectures with physics-informed methodologies. This would include embedding fundamental physical constraints such as conservation laws, thermodynamic principles, and chemical reaction kinetics into the transformer training processes.

Furthermore, generative transformer architectures such as decoder-only or encoder–decoder models present an opportunity for chemical reaction outcome prediction. These architectures, inspired by sequence-to-sequence tasks in natural language processing, are well-suited to tasks where both inputs and outputs are sequences, analogous to translating reactants into products. This enables chemical reactions to be modelled as text-like sequences, allowing transformers to translate sequences of reactants and reagents into corresponding product structures [53]. However, embedding geometrical constraints of the physical reactor might be challenging and require large datasets to represent their effects in these architectures.

5. Conclusion

In this work, we have demonstrated how series and parallel hybrid modelling approaches can be used for Digital Twinning of a chemical reactor platform. We have investigated the use of both Physics-Informed Neural Network (PINN) and Smoothed Particle Hydrodynamics with Genetic Algorithm (SPH+GA) approaches. We presented a Smoothed Particle Hydrodynamics with Genetic Algorithm (SPH+GA) as a series hybrid model, in which a physics-based simulation is followed by data-driven optimisation. In contrast, the Physics-Informed Neural Network (PINN) exemplifies a parallel hybrid model, which integrates mechanistic information and data-driven learning concurrently by embedding physical laws directly into the neural network's loss function. Both techniques rely on replicating the physical dynamics of the platform as well as optimisation and machine learning principles to support system identification and scaling. The comparative evaluation of PINN and SPH+GA in this study highlights their potential in advancing the modelling of chemical flow processes particularly in flow reactors where experimental data may be sparse or limited. Results show that both of our approaches present good predictive capabilities in such data-sparse scenarios.

In terms of performance, fidelity and resources required, PINNs have a lower computational footprint due to their reliance on ODEs and PDEs. However, their fidelity is lower compared to SPH based methods and they do not have the capability to capture the geometry of the reactor especially in situations where mixing impacts the chemical reactions of the experiments under consideration. On the other hand, SPH can do this but requires more compute resources as the level of fidelity increases. In terms of performance, these intricacies could be the reason why SPH+GA achieved lower predictive values when compared with PINNs.

Our future work will focus on extending the framework to more complex reaction networks and exploring its application in real-time monitoring and control of industrial reactors. In addition, our goal is to refine both the SPH+GA and PINN models to improve computational efficiency and scalability. This will make them even more practical for large-scale industrial applications especially in scenarios of low compute resources.

CRediT authorship contribution statement

Nur Aliya Nasruddin: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Nazrul Islam:** Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation. **Sergio Vernuccio:** Writing – review & editing, Supervision, Project administration, Conceptualization. **John Oyekan:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Software, Project administration, Methodology, Data curation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

We gratefully acknowledge the Engineering and Physical Sciences Research Council (EPSRC) in the funded projects; DigiCORTEX (EP/W014688/2), NanoMan (EP/V055089/1), and the Researcher in Residence scheme (EP/X528493/1). We also acknowledge support from the Graduate Excellence Programme (GrEP) by Majlis Amanah Rakyat (MARA), Malaysia.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ceja.2025.100775.

Data availability

Data will be made available on request.

References

[1] S.G. Espley, S.S. Allsop, D. Buttar, S. Tomasi, M.N. Grayson, Distortion/interaction analysis via machine learning, Digit. Discov. 3 (12) (2024) 2479–2486.

[2] S. Gow, M. Niranjan, S. Kanza, J.G. Frey, A review of reinforcement learning in chemistry, Digit. Discov. 1 (5) (2022) 551–567.

[3] Z. Tu, T. Stuyver, C.W. Coley, Predictive chemistry: machine learning for reaction deployment, reaction development, and reaction discovery, Chem. Sci. 14 (2) (2023) 226–244.

[4] M.S. Johnson, W.H. Green, A machine learning based approach to reaction rate estimation, React. Chem. Eng. 9 (6) (2024) 1364–1380.

[5] L. Zimányi, Á. Sipos, F. Sarlós, R. Nagypál, G.I. Groma, Machine-learning model selection and parameter estimation from kinetic data of complex first-order reaction systems, PLoS One 16 (8) (2021) e0255675.

[6] A.F. Psaros, X. Meng, Z. Zou, L. Guo, G.E. Karniadakis, Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons, J. Comput. Phys. 477 (2023) 111902.

[7] E.S. Muckley, J.E. Saal, B. Meredig, C.S. Roper, J.H. Martin, Interpretable models for extrapolation in scientific machine learning, Digit. Discov. 2 (5) (2023) 1425–1435.

[8] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nat. Rev. Phys. 3 (6) (2021) 422–440.

[9] L. Wang, Z. Yan, Data-driven rogue waves and parameter discovery in the defocusing nonlinear Schrödinger equation with a potential using the PINN deep learning, Phys. Lett. A 404 (2021) 127408.

[10] R. Mattey, S. Ghosh, A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations, Comput. Methods Appl. Mech. Engrg. 390 (2022) 114474.

[11] F. Pioch, J.H. Harmening, A.M. Müller, F.-J. Peitzmann, D. Schramm, O.e. Moctar, Turbulence modeling for physics-informed neural networks: Comparison of different RANS models for the backward-facing step flow, Fluids 8 (2) (2023) 43.

[12] L. Queiroz, F. Santos, J. Oliveira, M. Souza Jr., Physics-informed deep learning to predict flow fields in cyclone separators, Digit. Chem. Eng. 1 (2021) 100002.

[13] S. Cai, Z. Wang, F. Fuest, Y.J. Jeon, C. Gray, G.E. Karniadakis, Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks, J. Fluid Mech. 915 (2021) A102.

[14] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707.

[15] A. Rajendran, S.G. Subraveti, K.N. Pai, V. Prasad, Z. Li, How can (or why should) process engineering aid the screening and discovery of solid sorbents for CO2 capture? Acc. Chem. Res. 56 (17) (2023) 2354–2365, http://dx.doi.org/10.1021/acs.accounts.3c00335.

[16] S.I. Ngo, Y.-I. Lim, Solution and parameter identification of a fixed-bed reactor model for catalytic CO2 methanation using physics-informed neural networks, Catalysts 11 (11) (2021) 1304, http://dx.doi.org/10.3390/catal11111304.

[17] W. Ji, W. Qiu, Z. Shi, S. Pan, S. Deng, Stiff-pinn: Physics-informed neural network for stiff chemical kinetics, J. Phys. Chem. A 125 (36) (2021) 8098–8106, http://dx.doi.org/10.1021/acs.jpca.1c05102.

[18] X. Zou, H. Guo, C. Jiang, D.V. Nguyen, G.-H. Chen, D. Wu, Physics-informed neural network-based serial hybrid model capturing the hidden kinetics for sulfur-driven autotrophic denitrification process, Water Res. 243 (2023) 120331, http://dx.doi.org/10.1016/j.watres.2023.120331.

[19] H. Chen, E. Kätelhön, R.G. Compton, The application of physics-informed neural networks to hydrodynamic voltammetry, Analyst 147 (9) (2022) 1881–1891, http://dx.doi.org/10.1039/D2AN00456A.

[20] X. Chen, L.G. Wang, F. Meng, Z.-H. Luo, Physics-informed deep learning for modelling particle aggregation and breakage processes, Chem. Eng. J. 426 (2021) 131220, http://dx.doi.org/10.1016/j.cej.2021.131220.

[21] Z. Sun, H. Du, C. Miao, Q. Hou, A physics-informed neural network based simulation tool for reacting flow with multicomponent reactants, Adv. Eng. Softw. 185 (2023) 103525, http://dx.doi.org/10.1016/j.advengsoft.2023.103525.

[22] S.-Y. Tang, Y.-H. Yuan, Y.-C. Chen, S.-J. Yao, Y. Wang, D.-Q. Lin, Physics-informed neural networks to solve lumped kinetic model for chromatography process, J. Chromatogr. A 1708 (2023) 464346, http://dx.doi.org/10.1016/j.chroma.2023.464346.

[23] V. Bibeau, D.C. Boffito, B. Blais, Physics-informed neural network to predict kinetics of biodiesel production in microwave reactors, Chem. Eng. Process-ing Process. Intensif. 196 (2024) 109652, http://dx.doi.org/10.1016/j.cep.2023.109652.

[24] K. Prantikos, L.H. Tsoukalas, A. Heifetz, Physics-informed neural network solution of point kinetics equations for a nuclear reactor digital twin, Energies 15 (20) (2022) 7697, http://dx.doi.org/10.3390/en15207697.

[25] A. Moser, C. Appl, S. Brüning, V.C. Hass, Mechanistic mathematical models as a basis for digital twins, Digit. Twins: Tools Concepts Smart Biomanufacturing (2021) 133–180.

[26] I. Onaji, D. Tiwari, P. Soulatiantork, B. Song, A. Tiwari, Digital twin in manufacturing: conceptual framework and case studies, Int. J. Comput. Integr. Manuf. 35 (8) (2022) 831–858.

[27] A. Rasheed, O. San, T. Kvamsdal, Digital twin: Values, challenges and enablers from a modeling perspective, IEEE Access 8 (2020) 21980–22012.

[28] N. Padoin, T. Matiazzo, H.G. Riella, C. Soares, A perspective on the past, the present, and the future of computational fluid dynamics (CFD) in flow chemistry, J. Flow Chem. 14 (1) (2024) 239–256.

[29] J. Tu, G.H. Yeoh, C. Liu, Y. Tao, Computational Fluid Dynamics: A Practical Approach, Elsevier, 2023.

[30] G. Jimeno, Y.C. Lee, X.-W. Ni, Smoothed particle hydrodynamics–a new approach for modeling flow in oscillatory baffled reactors, Comput. Chem. Eng. 124 (2019) 14–27.

[31] J.Z. Yan, P. Kumar, W. Rauch, Effect of turbulent diffusion in modeling anaerobic digestion, Bioresour. Technol. 393 (2024) 130068, http://dx.doi.org/10.1016/j.biortech.2023.130068.

[32] M. Rezavand, et al., A fully Lagrangian computational model for the integration of mixing and biochemical reactions in anaerobic digestion, Comput. & Fluids 181 (2019) 224–235, http://dx.doi.org/10.1016/j.compfluid.2019.01.025.

[33] K. Ren, J. Chen, H. Qing, R. Chen, P. Chen, Y. Lin, B. Guo, Study on shock-induced chemical energy release behavior of Al/W/PTFE reactive material with mechanical-thermal-chemical coupling SPH approach, Propellants Explos. Pyrotech. 45 (12) (2020) 1937–1948.

[34] A.A. Ali, C.L. Dobbs, T.J. Bending, A.S. Buckner, A.R. Pettitt, Star cluster formation and feedback in different environments of a Milky Way-like galaxy, Mon. Not. R. Astron. Soc. 524 (1) (2023) 555–568.

[35] Y. Hong, et al., Adaptive resolution SPH method for non-Newtonian slurry mixing, Int. J. Mech. Sci. 212 (2025) 106846, http://dx.doi.org/10.1016/j.ijmecsci.2024.106846.

[36] R. Cherepanov, et al., SPH simulation of shock-induced chemical reactions in reactive powder mixtures, Mater. Today Commun. 38 (2024) 108261, http://dx.doi.org/10.1016/j.mtcomm.2024.108261.

[37] L.-T. Zhu, J.-X. Tang, Z.-H. Luo, Machine learning to assist filtered two-fluid model development for dense gas–particle flows, AIChE J. 66 (6) (2020) e16973.

[38] L.-T. Zhu, X.-Z. Chen, B. Ouyang, W.-C. Yan, H. Lei, Z. Chen, Z.-H. Luo, Review of machine learning for hydrodynamics, transport, and reactions in multiphase flows and reactors, Ind. Eng. Chem. Res. 61 (28) (2022) 9901–9949.

[39] A.M. Schweidtmann, D. Zhang, M. von Stosch, A review and perspective on hybrid modeling methodologies, Digit. Chem. Eng. 10 (2024) 100136.

[40] P. Shah, S. Pahari, R. Bhavsar, J.S.-I. Kwon, Hybrid modeling of first-principles and machine learning: A step-by-step tutorial review for practical implementation, Comput. Chem. Eng. (2024) 108926.

[41] C.J. Taylor, M. Booth, J.A. Manson, M.J. Willis, G. Clemens, B.A. Taylor, T.W. Chamberlain, R.A. Bourne, Rapid, automated determination of reaction models and kinetic parameters, Chem. Eng. J. 413 (2021) 127017, http://dx.doi.org/10.1016/j.cej.2020.127017.

[42] L. Lu, X. Meng, Z. Mao, G.E. Karniadakis, DeepXDE: A deep learning library for solving differential equations, SIAM Rev. 63 (1) (2021) 208–228, http://dx.doi.org/10.1137/19M127406.

[43] R. Pérez-Illanes, G. Sole-Mari, D. Fernàndez-Garcia, Smoothed particle hydrodynamics for anisotropic dispersion in heterogeneous porous media, Adv. Water Resour. 183 (2024) 104601.

[44] S. Laha, G. Fourtakas, P.K. Das, A. Keshmiri, Smoothed particle hydrodynamics based FSI simulation of the native and mechanical heart valves in a patient-specific aortic model, Sci. Rep. 14 (1) (2024) 6762.

[45] R. Erban, S.J. Chapman, Stochastic modelling of reaction–diffusion processes: algorithms for bimolecular reactions, Phys. Biology 6 (4) (2009) 046001, http://dx.doi.org/10.1088/1478-3975/6/4/046001.

[46] H. Sano, M. Tachiya, Effective radius of a contact diffusion-controlled reaction between spherical particles, J. Chem. Phys. 161 (22) (2024) 224105, http://dx.doi.org/10.1063/5.0226416.

[47] S. Katoch, S.S. Chauhan, V. Kumar, A review on genetic algorithm: past, present, and future, Multimedia Tools Appl. 80 (2021) 8091–8126.

[48] G.A. Sgarro, L. Grilli, D. Santoro, Optimal multivariate mixture: a genetic algorithm approach, Ann. Oper. Res. (2024) 1–22.

[49] K. Ghosh, S. Vernuccio, A.W. Dowling, Nonlinear reactor design optimization with embedded microkinetic model information, Front. Chem. Eng. 4 (2022) 898685.

[50] E. Koninckx, J.G. Colin, L.J. Broadbelt, S. Vernuccio, Catalytic conversion of alkenes on acidic zeolites: automated generation of reaction mechanisms and lumping technique, ACS Eng. Au 2 (3) (2022) 257–271.

[51] S. Vernuccio, E.E. Bickel, R. Gounder, L.J. Broadbelt, Propene oligomerization on Beta zeolites: Development of a microkinetic model and experimental validation, J. Catalysis 395 (2021) 302–314.

[52] S. Vernuccio, L.J. Broadbelt, Discerning complex reaction networks using automated generators, AIChE J. 65 (8) (2019) e16663.

[53] N. Yoshikawa, M. Skreta, K. Darvish, S. Arellano-Rubach, Z. Ji, L. Bjørn Kristensen, A.Z. Li, Y. Zhao, H. Xu, A. Kuramshin, et al., Large language models for chemistry robotics, Auton. Robots 47 (8) (2023) 1057–1086.