



Review

Hybrid modeling of first-principles and machine learning: A step-by-step tutorial review for practical implementation

Parth Shah, Silabrata Pahari, Raj Bhavsar, Joseph Sang-Il Kwon *

Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77845, USA
Texas A&M Energy Institute, Texas A&M University, College Station, TX 77845, USA

ARTICLE INFO

Keywords:

First-principles models
Machine learning
Hybrid modeling
Tutorial review
Process systems engineering
Neural networks
Sensitivity analysis

ABSTRACT

In recent years, the integration of mechanistic process models with advanced machine learning techniques has led to the development of hybrid models, which have shown remarkable potential across various domains. However, despite numerous applications and reviews, there is a significant gap in practical resources that guide new researchers through the process of building these models from the ground up. This work addresses this gap by offering a comprehensive tutorial designed to demystify the development of hybrid models. We focus on the practical implementation, beginning with fundamental concepts and advancing to detailed mathematical formulations, providing a step-by-step walkthrough for constructing hybrid models. The tutorial includes detailed case studies illustrating the application of hybrid models in solving complex problems in process systems engineering. By following this guide, researchers will acquire the necessary tools and knowledge to apply hybrid modeling techniques effectively for real-world implementations, paving the way for further innovation and adoption in the field.

Contents

1. Introduction	2
2. Preliminaries	3
2.1. First-principles models	3
2.2. Data-driven Models (DDMs)	4
2.3. Hybrid models	5
2.3.1. Training algorithms for DNN	5
2.3.2. Hybrid model training	6
2.3.3. Sensitivity analysis	8
3. Step-by-step implementation of hybrid modeling framework	8
4. Application: Extension of hybrid modeling framework	12
4.1. Adoption of hybrid modeling in engineering and science	13
4.1.1. Process modeling	13
4.1.2. Process monitoring	15
4.1.3. Process control	16
4.1.4. Model discovery	18
4.1.5. Plant scale up	18
4.2. Case studies	19
4.2.1. Application of hybrid models to crystallization	19
4.2.2. Fermentation process	20
4.2.3. Application to hydraulic fracturing	22
4.2.4. Application to reaction-diffusion system	23
5. Outlook	24
5.1. Adaptive hybrid models	24
5.2. Multiscale hybrid modeling	25

* Corresponding author at: Artie McFerrin Department of Chemical Engineering, Texas A&M University, College Station, TX 77845, USA.
E-mail address: kwonx075@tamu.edu (J.S.-I. Kwon).

5.3. Integration with state-of-the-art ML models	26
5.4. Automatic determination of the hybrid model structure	26
5.5. Integration with uncertainty quantification techniques	26
5.6. Development of software tools.....	26
5.7. Real-world implementation	26
6. Conclusion	27
CRedit authorship contribution statement	27
Declaration of competing interest	27
Acknowledgments	27
Data availability	27
References.....	27

1. Introduction

Mathematical models serve as tools for representing physical systems, enabling the analysis of how these systems respond to changing operating conditions and external factors. In engineering, such models are indispensable for a range of applications, including large-scale process modeling (Ali et al., 2015), process control (Kiparissides et al., 1993), process monitoring (Bui et al., 2021), equation discovery from data (Raissi et al., 2019), and the critical tasks of process scale-up and development (Schmidt, 2005). In chemical engineering, the use of accurate mathematical models has revolutionized process development, significantly reducing the time required for transition from laboratory research to market implementation by effectively addressing scaling challenges (Jana, 2018; Cameron and Hangos, 2001). Models are generally categorized based on their underlying philosophy into three primary types. First-principles models (FPMs), also known as white-box models, are grounded in the fundamental physics of the process, offering complete transparency and interpretability (Nellis and Klein, 2008; Chorin, 1968). On the other hand, data-driven models (DDMs) rely on empirical data and statistical methods to make predictions, often sacrificing interpretability for flexibility and ease of development. Finally, hybrid models or gray-box models combine elements of both first-principles and data-driven approaches, leveraging the strengths of each to improve model accuracy and applicability (Zendehboudi et al., 2018; Schweidtmann et al., 2023). This classification highlights the spectrum of modeling approaches available, each suited to different types of engineering problems, depending on the availability of data, the need for interpretability, and the complexity of the system being modeled.

FPMs are deeply rooted in the physical principles of the process, typically derived from conservation laws of energy, mass, momentum, and transition state theory for chemical reactions (Grabow et al., 2008; Taylor, 2016). Due to their basis in process physics, FPMs offer high flexibility, extrapolation capability, and interpretability. They require no process data, making them ideal for building models for process monitoring, process control, model discovery, and plant scale-up. FPMs might be based on ordinary differential equations (ODEs), partial differential equations (PDEs) (Aziz and Khan, 2018), algebraic equations (Iglesias-Silva et al., 2003), and event-based modeling frameworks (Battaille and Srolovitz, 2002). These models are complex to develop, and capturing latent mechanisms through FPMs is challenging.

Conversely, black-box or DDMs are based entirely on process-scale data with minimal interpretability (Alauddin et al., 2023; Ghosh et al., 2022). These models are easier and cheaper to develop than FPMs but suffer from poor extrapolation capabilities and lack flexibility, as their application domain is confined to the range of collected process data (Bangi and Kwon, 2023). The recent paradigm shift in the more extensive application of DDMs over FPMs in the process industry is a result of the growing capability of computing architecture, which allows DDMs to capture long-term spatiotemporal dependencies in the process data (Lin et al., 1996). It is important to note that DDMs are only developed after the actual process is set up, as the accuracy of these models is highly dependent on high-quality process data. These models

excel in managing unknown model complexities and learning from historical process-scale data (Alhajeri et al., 2022b). Thus, the DDMs are extensively used in process operation activities such as predictive maintenance, fault detection and diagnosis, process monitoring, and process-scale decision-making (Wu et al., 2020; Kumari et al., 2022). While these models continuously improve, data-driven approaches depend entirely on high-quality, low-noise, and low-disturbance data, which is still challenging to obtain.

The limitations of FPMs and DDMs have led to the development of gray-box models, which combine the strengths of both approaches (Oussar and Dreyfus, 2001; Asgari et al., 2021). These models include first-principles and data-driven components, aiming to maintain interpretability while reducing reliance on process data (Bohlin and Graebe, 1995; Bohlin, 2013). They effectively use the DDMs' ability to capture unknown process complexities within the framework of first-principles equations. The term "gray-box modeling" emerged in the 90s within the systems theory literature (Psichogios and Ungar, 1992; Jørgensen and Hangos, 1995). Since then, various gray-box models have been applied in chemical engineering, like polymerization processes (Fiedler and Schuppert, 2008), flowsheet simulators for chemical processes (Asprion et al., 2019), distillation columns (Mahalec and Sanchez, 2012; Rodriguez et al., 2023), bacterial cultivations (Hong and Qiu, 2008), chemical reactors (Bangi and Kwon, 2023; Azarpour et al., 2017; Zendehboudi et al., 2014), separation systems (Romijn et al., 2008), crystallization (Bogaerts et al., 2009; Sitapure and Kwon, 2023c; Ghosh et al., 2021), fermentations (Laursen et al., 2007; Shah et al., 2023b; Tornøe et al., 2004), hydraulic fracturing (Bangi and Kwon, 2020), and intracellular signaling pathways (Lee et al., 2020a). These models encompass semi-analytical, semi-physical, and semi-parametric models (Talib et al., 2023).

This tutorial review centers on hybrid models, a distinct and increasingly pivotal subset of gray-box models. Unlike traditional approaches, hybrid models uniquely integrate first-principles equations with empirically derived or data-driven methods, creating a versatile framework capable of addressing the limitations of purely FPMs or DDMs. In this work, we define hybrid models as those that use first-principles equations as the structural backbone, while data-driven components capture the uncertain parameters or states that are difficult to model explicitly through physics-based equations alone (Fiedler and Schuppert, 2008; Kahrs and Marquardt, 2007; Kwon, 2024). The novelty of hybrid models lies in their ability to effectively capture nonlinear processes and address plant-model mismatches caused by uncertain parameters, which often challenge conventional modeling techniques (Thangavel et al., 2018). Furthermore, hybrid models excel as predictive tools in scenarios where process disturbances undermine the reliability of FPMs (Hosen et al., 2011). Recent studies also demonstrate that hybrid models have a broader domain of applicability (Bangi and Kwon, 2023). Hybrid models incorporate a diverse array of data-driven techniques, including neural networks (NNs), regression methods (linear and nonlinear), classification algorithms (such as support vector machines (SVMs) (Adib et al., 2013) and decision trees (Yarveicy et al., 2019)), subspace identification methods (Hassanpour et al., 2022; Ghosh et al., 2019) like, Approximate Dynamic Programming (ADP) (Singh Sidhu et al., 2018) Gaussian

Processes (GPs) (Gray and Schmidt, 2018; Jiang et al., 2019), sparse regression techniques like sparse Identification of Nonlinear Dynamical Systems (SINDy) (Bhadiraju et al., 2020, 2021b), ensemble techniques like random forests (Chen et al., 2020), and Bayesian networks (Eugene et al., 2020; Yu et al., 2024). This combination offers exceptional flexibility and robustness in modeling complex system behaviors, enabling hybrid models to capture dynamics that are difficult to describe using traditional approaches alone (Bohlin, 2006).

Hybrid models offer significant advantages over conventional DDMs, requiring less training data and providing improved interpretability and generalizability. The integration between FPMs and DDMs is either done through parameters that are a part of the FP equations or through the estimation of discrepancy between the FP model's predictions and experimental data (Liu et al., 2021). These parameters are selected through comprehensive domain knowledge or rigorous statistical sensitivity analysis, ensuring they hold physical significance and highly impact the final process states. This approach captures parametric and process uncertainties and reveals latent mechanisms that might be missed by traditional domain knowledge. Comprehensive reviews by Von Stosch et al. (2014) and von Stosch et al. (2014) have discussed some fundamental aspects of hybrid modeling and highlighted its advantages over traditional non-hybrid models in process engineering, showcasing their utility in various stages such as development, validation, optimization, monitoring, control, and scaling. There have been other review articles evaluating the strengths and limitations of hybrid modeling frameworks, highlighting their performance advantages, recent advances, and areas for improvement, such as uncertainty quantification, dynamic hybrid modeling, and constraint handling (Schweidtmann et al., 2023; Sansana et al., 2021; Bradley et al., 2022; Sharma and Liu, 2022). There have also been some available implementations of hybrid models in the literature (Moon et al., 2022; Nascimento et al., 2020; Cuomo et al., 2022; Di Caprio et al., 2023; Merkelbach et al., 2022). While these references provide comprehensive explanations of how physics can be incorporated into ML models, they do not offer a detailed tutorial with step-by-step implementation using the hybrid model structure. Our focus is on presenting a structured approach with complex case studies to guide readers through the implementation process, making our work distinct in its practical and tutorial-oriented approach.

Our contribution to this tutorial review is twofold:

- We offer a practical guide for building hybrid models from scratch, with detailed mathematical formulations, pseudo-code, and real-world case studies.
- We provide a comprehensive overview of the foundational concepts and recent advancements in hybrid modeling

This work aims to bridge the gap between theory and application, equipping researchers and practitioners with the tools to harness the full potential of hybrid models in process systems engineering. Fig. 1 illustrates the spectrum of different modeling techniques, emphasizing the unique position and advantages of hybrid models within this landscape.

The remainder of the manuscript is organized as follows: In Section 2, preliminary terminologies are presented to understand the hybrid model better. In Section 3, details about the building blocks of the hybrid model are presented along with a code for the hybrid model in TensorFlow. Section 4 highlights the application of the hybrid models to the different segments of chemical engineering, like process modeling, process control, process monitoring, plant scale-up, and model discovery. Then, more detailed case studies highlighting the application of the hybrid models to different systems is presented. Section 5 highlights an outlook on the future challenges of hybrid modeling. Finally, Section 6 offers concluding remarks.

2. Preliminaries

This section provides an overview of the fundamental concepts in hybrid modeling, encompassing both theoretical and ML components. We begin by exploring the essential modeling frameworks and associated terminologies, highlighting how FPMs and DDMs have been utilized for various engineering applications. The discussion then highlights the limitations of FPM and DDM, and the motivation behind hybrid modeling methodologies. It particularly focuses on their role in identifying missing physics and latent chemical mechanisms, expressed as time-varying and spatially-varying parameters, as well as spatiotemporally-varying parameters.

2.1. First-principles models

FPMs, also called mechanistic, semi-empirical, phenomenological, or white-box models, are recognized for their thoroughness, even though they are based on a simplified, idealized understanding of physical phenomena. Because they are constructed to align with the laws of physics, such as conservation, kinetics, or thermodynamics laws, through their mathematical formulations, FPMs are considered both interpretable and physically consistent. FPMs can be developed by modeling the equations in programming languages like MATLAB, Python, C++, and FORTRAN. FPMs can also be developed through equation-oriented modeling, which involves using specialized software frameworks that require the user to explicitly define all relevant variables and equations. These frameworks typically offer user-friendly programming environments tailored for chemical engineering applications. Notable examples include gPROMS, Aspen Custom Modeler, Dymola, ASCEND II, and JModelica.org (Elmqvist, 1978; Piela et al., 1991; Åström et al., 1998; Leal et al., 2017). FPMs have been used for applications like surface catalysis (Hansen and Neurock, 1999; Neurock and Hansen, 1998; Lee et al., 2023), polymerization (Dotson et al., 1996; Brandolin et al., 1996; Dubé et al., 1997), fermentation (Ordonez et al., 2016; Luedeking and Piret, 2000), reactors (Kiparissides et al., 1993; Raupp et al., 2001), crystallization (Sitapure et al., 2021b; Nagpal et al., 2024; Hoja et al., 2017; Quintana-Hernandez et al., 2004), kraft pulping (Choi and Kwon, 2019; Son et al., 2020; Fearon et al., 2020; Yang and Liu, 2005), lithium-ion batteries (Pannala et al., 2015; Katrašnik et al., 2021; Lee et al., 2021), solar cells (Crose et al., 2015; Basore, 1990; De Falco et al., 2012), distillation (Taylor and Krishna, 2000; Mueller and Kenig, 2007), quantum dots (Zhu et al., 2007; Sitapure et al., 2021a; Kuno et al., 2003), phase equilibria (Liu, 2009; Tsivintzelis et al., 2010), and material chemistry (Greeley and Mavrikakis, 2004; Van der Ven et al., 1998).

In certain instances, these modeling techniques may be unfamiliar to modelers or inappropriate for particular research projects. For example, there are no readily available open-source packages for conducting bifurcation analysis on thermally coupled distillation columns. Therefore, to carry out a hysteresis study, a blend of different libraries (MatCont and Aspen) is required (Caranza-Abaíd and González-García, 2020). Another challenge in developing FPMs is the significant need for various model subroutines to enhance the core FPM. For instance, modeling an absorption column necessitates supplementary models for vapor-liquid equilibrium, kinetics, viscosity, surface tension, diffusivity, and packing correlations.

To better understand FPMs, let us consider a system whose state at time t is described by a vector $x(t) \in \mathbb{R}^n$, where $x(t)$ could represent concentrations of chemical species in a reactor or temperatures in a thermal system. The evolution of this system is governed by an ODE of the form:

$$\frac{dx}{dt} = f(x(t), u(t), p), \quad x(0) = x_0 \quad (1)$$

Here, $x(t)$ is the state vector, $u(t)$ is an input or control vector (e.g., heat input, flow rates), p represents the parameters of the system (e.g., reaction rates, diffusion coefficients), and f is a function

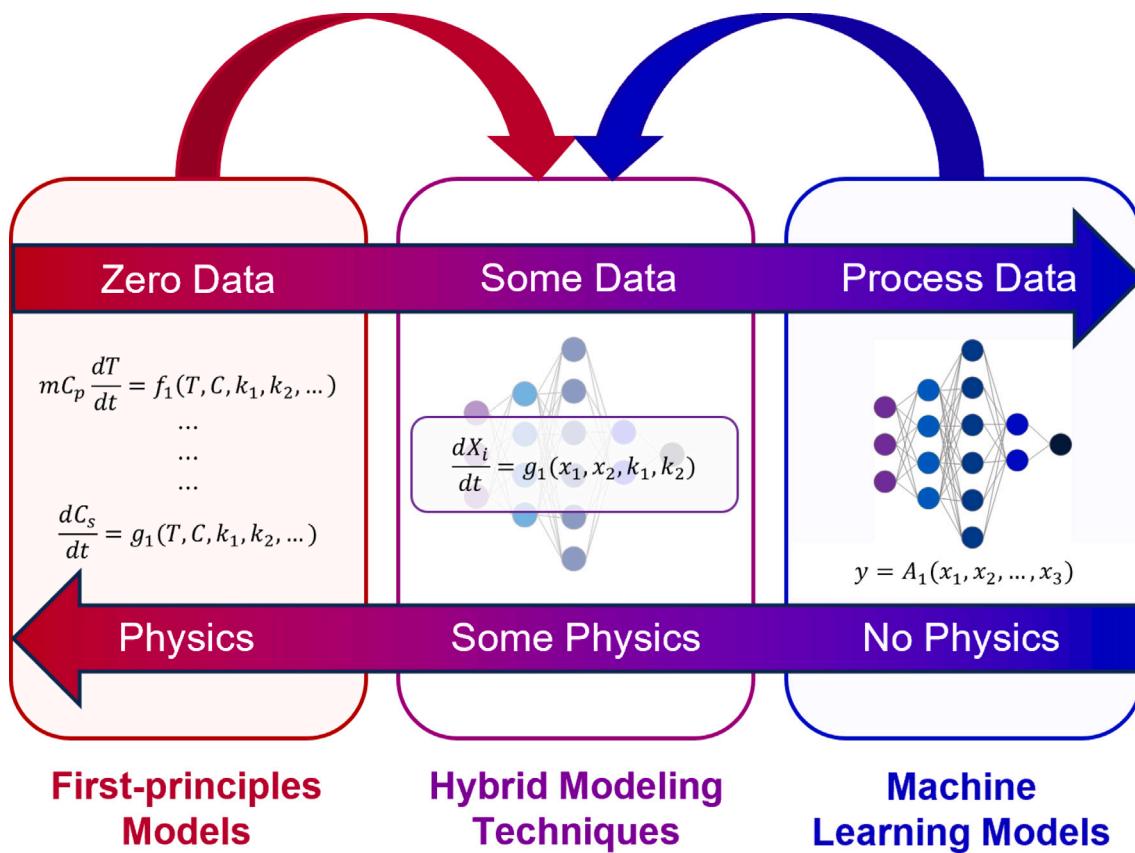


Fig. 1. Spectrum of different modeling methodologies.

describing the system's dynamics. The system's behavior is modeled based on well-understood physical laws, such as the conservation of mass, momentum, or energy. While FPMs can take the form of ODEs, PDEs, and even include algebraic equations or states, for the purposes of this tutorial review and step-by-step implementation, we primarily focus on systems modeled by ODEs, as illustrated in Eq. (1). However, it is important to note that the approach discussed here equally applies to systems involving algebraic equations, though our specific case studies in Section 4.2 focus on differential equations. This broader applicability allows for flexibility in extending the hybrid modeling framework to a variety of system types. In subsequent sections, we will demonstrate how hybrid models can integrate these FPMs with data-driven components to address uncertainties and complex mechanisms in system dynamics that are not captured by first-principles alone.

2.2. Data-driven Models (DDMs)

DDMs, based on data rather than explicit causal relationships between variables, utilize a set of equations that depend heavily on the data's quality rather than their mathematical form. This lack of clarity in how they process information earns them the nickname "black boxes". In chemical engineering, key data-driven modeling approaches can be categorized into several broad areas. Regression techniques, including both linear and nonlinear regression models, are often used to capture the relationship between input and output variables (Khataee et al., 2014; Özdemir et al., 2011). Classification methods, such as SVMs (Bansal et al., 2012) and decision trees (Yarveicy et al., 2019; Ghiasi and Mohammadi, 2017), are widely employed for tasks that require sorting data into predefined categories. Probabilistic models, like Bayesian networks, are used to handle uncertainties and probabilistic relationships between variables (Kumari et al., 2022; Galagali and Marzouk, 2015; Pahari et al., 2021b). Ensemble methods, including random forests, combine multiple models to improve prediction accuracy and

handle complex, nonlinear interactions (Ren et al., 2017; Chen et al., 2020). Nonlinear systems modeling techniques, such as deep neural networks (DNNs) (Shah et al., 2023a; Choi et al., 2023) and multivariate adaptive regression splines (MARS) (Yang et al., 2003; Alotaibi et al., 2020), are particularly useful for modeling complex, nonlinear systems. Additionally, dimensionality reduction methods, like principal component analysis (PCA) and other latent variable techniques, help in reducing the dimensionality of data while retaining its key features and are often used for noise reduction or feature extraction (Zhou et al., 2016; Garg and Mhaskar, 2017; Bao et al., 2019; Narasingam and Kwon, 2018). Koopman operator theory helps in representing nonlinear dynamical systems in a linear framework, which can facilitate the analysis and control of such systems in high-dimensional spaces (Narasingam and Kwon, 2019; Wang and Wu, 2024; Zhang et al., 2023). Dynamic Mode Decomposition (DMD) has also been widely used to decompose complex systems into dynamically relevant modes, providing insights into the temporal evolution of these systems by extracting coherent structures from time-series data (Narasingam and Kwon, 2017; Son et al., 2022a; Williams et al., 2015). SINDy is a data-driven method that uses sparse regression to discover the underlying governing equations of nonlinear dynamical systems directly from observational data (Brunton et al., 2016; Bhadriraju et al., 2019, 2021a).

ANNs stand out for their capability to handle complex, nonlinear systems (Bishop and Nasrabadi, 2006). They were initially inspired by the structure of biological brains, aiming to mimic how humans process information. Despite their neuroscientific origins, ANNs have found broader applications in fields like data science and computer science due to their ability to approximate any function, a property proven in the late 1980s. In the past 15–20 years, ANNs have been used for various engineering applications (Himmelblau, 2008; Pirdashti et al., 2013; Kirilova, 2022). Structurally, ANNs consist of input, hidden, and output layers. The input layer receives data, the output layer presents

the results, and the hidden layers, made up of neurons, process the input data to predict outputs. These neurons transform input data by combining input vectors, typically through linear combinations or element-by-element multiplication, primarily used in image processing and classification. The transformation process often involves applying a nonlinear or logical transfer function, such as the hyperbolic tangent, log-sigmoid function, or rectified linear unit (ReLU). However, linear functions can also be used in some cases. In traditional ANNs, there are typically only one or a few hidden layers, and these networks are designed to handle relatively simpler, nonlinear relationships between inputs and outputs. DNNs, on the other hand, are a specific type of ANN characterized by having multiple hidden layers—often several or more. This depth allows DNNs to learn hierarchical representations of data, capturing more complex patterns and features than a standard ANN can. In essence, while all DNNs are ANNs, not all ANNs are DNNs. The key distinction is the number of hidden layers, with DNNs being used for more complex tasks that require capturing intricate relationships in large datasets. In this review paper, we will use DNN as the standard in our hybrid model. While DNNs are a popular tool for hybrid modeling due to their flexibility and ability to approximate nonlinear functions, other machine learning techniques, such as SVMs, GPs, decision trees, subspace identification models, and multivariate regression models, have also proven effective in handling process uncertainties and providing predictive insights. The choice of method depends on the specific system dynamics, available data, and desired interpretability of the model. For the purpose of this review article, we use DNNs to illustrate the steps involved in integrating DDMs with FPMs.

DNNs consist of NNs incorporating multiple hidden layers, each comprising several neurons, as shown in Fig. 2 (Szegedy et al., 2013; Gaweht et al., 2016). Within each layer, every neuron connects to every neuron in the subsequent and preceding layers. Let $n^{k+1}(i)$ be the cumulative input received by unit i in layer $k + 1$ which is given as

$$n^{k+1}(i) = \sum_{j=1}^{S_k} w^{k+1}(j, i)a^k(j) + b^{k+1}(i) \quad (2)$$

where S_k is the number of neurons in layer k . These interconnections are characterized by a parameter known as the weight, denoted by w . Furthermore, the neurons in each layer are assigned a bias term, represented by b , and are associated with a specific activation function, denoted by f . $a^k(j)$ is the output of unit j in layer k which is given as

$$a^k(j) = f^k(n^k(j)), \quad j = 1, \dots, S_k \quad (3)$$

Assuming there are M layers in the network, the equations in matrix form can be represented as

$$\begin{aligned} A^k &= F^k(W^k A^{k-1} + B^k), \quad k = 0, 1, \dots, M - 1 \\ A^0 &= u_q, \quad q = 1, 2, \dots, Q \end{aligned} \quad (4)$$

where u_q is the input vector given to the NN whose corresponding output is A_q^M obtained at the final layer M . The column vectors A^k , F^k , and B^k contain the outputs, the activation functions, and biases of all the neurons in layer k , respectively. The matrix W^k contains weights associated with the neurons in layers k and $k - 1$; in particular, each row contains all the weights associated with a specific neuron of layer k , and the number of columns is equal to the number of neurons in layer $k - 1$. The aim of the DNN is to learn the functional relationship between input and output pairs $\{(u_1, y_1), (u_2, y_2), \dots, (u_Q, y_Q)\}$. The accuracy of the DNN is measured as follows:

$$\begin{aligned} V &= \frac{1}{2} \sum_{q=1}^Q e_q^T e_q \\ e_q &= y_q - A_q^M \end{aligned} \quad (5)$$

where V is the summation of error over the entire duration, and e_q is the error between the predicted output A_q^M , and the actual output y_q when the q th input (i.e., u_q) is fed to the DNN. The error matrix E can be defined as follows:

$$E = [e_1 \ e_2 \ \dots \ e_Q]^T \quad (6)$$

2.3. Hybrid models

FPMs often face limitations due to gaps in system knowledge. These gaps pose significant challenges in accurately identifying and incorporating physics components necessary for modeling latent chemical reactions and dependencies on unobservable state variables. The difficulty in determining which elements to integrate increases without a comprehensive system understanding, hindering the ability to capture the full spectrum of system dynamics. On the other hand, ML models depend heavily on the quality of the data on which they are trained. Despite their flexibility, the effectiveness of ML models is closely tied to acquiring high-quality data, a task complicated by noise and disturbances in real-world processes. This dependency on data quality challenges the creation of robust models that can reliably interpret and predict based on underlying process data. Hybrid modeling addresses these challenges by combining the strengths of both approaches. It integrates the broad applicability and interpretability of FPMs, which are based on system-independent physics laws, with the adaptability of ML models to leverage system-specific process data. Compared to FPMs, hybrid models offer superior extrapolation capabilities beyond the data range, although the conditions required to guarantee their extrapolation accuracy require further research.

Hybrid models can be constructed using serial or parallel hybrid frameworks (Sansana et al., 2021) as shown in Fig. 3. The core concept remains consistent, but the interaction between FPMs and DDMs varies. In serial hybrid modeling, DDMs and FPMs are aligned sequentially. Input variables are introduced into the DDM, which then passes its output to the FPM, or vice versa (Tsen et al., 1996). Psichogios and Ungar (Psichogios and Ungar, 1992) first utilized this method in fed-batch bioreactors, replacing empirical elements of a kinetic model with an ANN. Applications include predicting pulp quality (Aguiar and Maciel Filho, 2001), polymerization kinetics (Tian et al., 2001), fermentation processes (Bazaei and Majd, 2003; Bangi et al., 2022), and ethylene glycol production (Kahrs et al., 2009). In contrast, the parallel hybrid modeling framework develops the FPM separately from the DDM. The DDM predicts the discrepancy between the FPM's outputs and experimental results. This approach was pioneered by Su et al. (1993) for reactor system models, where outputs from both models were combined. Thompson and Kramer (1994) later applied a similar methodology to predict fermentation kinetics. Recently, advanced algorithms have improved the accuracy of parallel hybrid models in applications such as flowmeters (Bikmukhametov and Jäschke, 2020).

2.3.1. Training algorithms for DNN

The error backpropagation (EBP) algorithm (Rumelhart et al., 1986; Werbos, 1988) represents a pivotal advancement in the training of the DNN. However, it is often criticized for its slow convergence, which can be attributed to two primary factors. First, the step size must be kept small to avoid oscillations around the desired minima, inevitably resulting in a slower training process. Second, the curvature of the error surface can vary across different directions, introducing the classic "error valley" problem (Osborne, 1992), which further hinders the convergence rate. Despite these drawbacks, the steepest descent algorithm remains a prevalent method for training NNs. Its update rule leverages the gradient g , the first-order derivative of the total error function, defined as follows:

$$g = \frac{\partial V(u, w)}{\partial w} = \left[\frac{\partial V}{\partial w_1} \quad \frac{\partial V}{\partial w_2} \quad \dots \quad \frac{\partial V}{\partial w_N} \right]^T \quad (7)$$

The update rule of the steepest descent algorithm is written as:

$$w_{k+1} = w_k - \alpha g_k \quad (8)$$

where α is the learning constant. The issue of slow convergence in the steepest descent algorithm can be mitigated by employing the Gauss–Newton algorithm (Osborne, 1992). This algorithm leverages the second-order derivatives of the error function to accurately assess

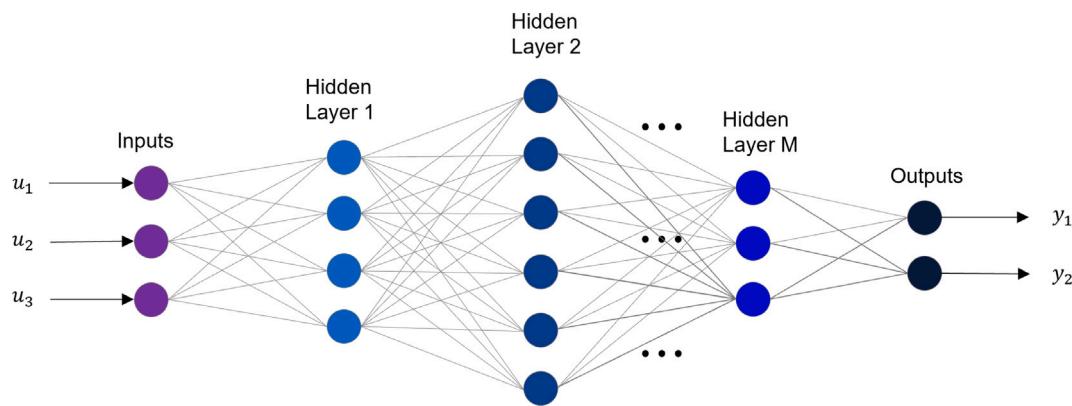


Fig. 2. Schematic of a DNN.

Table 1
Comparison of training algorithms.

Algorithm	Parameter updates	Convergence	Computational issue
EBP	$w_{k+1} = w_k - \alpha g_k$	Stable, slow	Gradient
Gauss–Newton	$w_{k+1} = w_k - (J_k^T J_k)^{-1} J_k E_k$	Unstable, fast	Jacobian
Levenberg–Marquardt	$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k E_k$	Stable, fast	Jacobian

the curvature of the error surface and determine the optimal step size for each direction. The Gauss–Newton algorithm achieves rapid convergence when the error function approximates a quadratic surface. However, in the absence of such a quadratic nature, the algorithm may exhibit significant divergence. The update rule of the Gauss–Newton method is defined as:

$$w_{k+1} = w_k - (J_k^T J_k)^{-1} J_k E_k \quad (9)$$

where J is the Jacobian matrix which is defined as:

$$J = \begin{bmatrix} \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \dots & \frac{\partial e_1}{\partial w_N} \\ \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \dots & \frac{\partial e_2}{\partial w_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial e_Q}{\partial w_1} & \frac{\partial e_Q}{\partial w_2} & \dots & \frac{\partial e_Q}{\partial w_N} \end{bmatrix} \quad (10)$$

The Levenberg–Marquardt algorithm (Levenberg, 1944; Marquardt, 1963) amalgamates the steepest descent method with the Gauss–Newton algorithm, thereby inheriting the stability of the former and the speed of the latter. This algorithm operates as the steepest descent method when the curvature of the error surface is complex. Once the local curvature approximates a quadratic surface, it functions as the Gauss–Newton algorithm. Essentially, the Levenberg–Marquardt algorithm alternates between utilizing the advantageous features of both the steepest descent and Gauss–Newton algorithms, depending on the situational requirements. The update rule of the Levenberg–Marquardt algorithm is defined as follows:

$$w_{k+1} = w_k - (J_k^T J_k + \mu I)^{-1} J_k E_k \quad (11)$$

where μ is the combination coefficient, and I is the identity matrix. When the μ value is large, the Levenberg–Marquardt algorithm behaves as the steepest descent method; otherwise, it behaves as the Gauss–Newton method (see Table 1).

2.3.2. Hybrid model training

Hybrid models combine the strengths of both ODEs and DNNs. One way to achieve this is by integrating a DNN within an ODE framework. Consider a scenario where the system dynamics f are

partially unknown or too complex to model explicitly. A DNN can be used to approximate this unknown function g :

$$\begin{aligned} \frac{dx}{dt} &= f(x(t), u(t), p) \\ p &= g(x, u) \\ A^M &= p \end{aligned} \quad (12)$$

This equation represents a hybrid model where the DNN learns the complex dynamics directly from data, guided by the structure provided by the ODE framework. The parameters p are not unknown but are understood to be important for accurate modeling of the process. For instance, in biological reactions, the cell growth rate and the corresponding reaction kinetics are usually unknown and difficult to derive from first-principles. Consequently, the FPMs of such processes will contain unknown terms. These parameters are related to the states x and inputs u through the equation g . The DNN, as shown in Eq. (4), can determine these unknown terms. The input to the NN model consists of the system's current state, x_k , and the external input, u_k , with its output being the current parameter value, p (this is equal to the DNN output A^M). This output is then integrated with the current state and input (x_k and u_k) as inputs to the FPM, which forecasts the system's state at the next time step (x_{k+1}), as shown in Fig. 4.

To activate the NN, it must first undergo training using input and state measurements. The network is initialized with specific hyperparameters, which are then adjusted by backpropagating an error signal throughout the network. In the context of the hybrid model, the output from the NN, which are physically interpretable parameters, is indirectly assessed since it is not directly measurable. Training is deemed complete when the discrepancy between the hybrid model's predictions and the actual observations is within an acceptable margin of error. Once adequately trained, the NN, in tandem with the FPM, operates as the hybrid model. Thus, hybrid modeling offers a robust framework for tackling complex systems where traditional modeling approaches fall short. By leveraging the strengths of DDMs and theoretical models which are in the form of ODEs, researchers can gain deeper insights into the underlying mechanisms of the systems they study.

To explain further, we consider an example system described by a set of ODEs representing multiple interacting components or species in a biological system:

$$\frac{dx}{dt} = f_1(x, y, p_1) = p_1 \cdot x \cdot (1 - x) - \alpha \cdot x \cdot y \quad (13)$$

$$\frac{dy}{dt} = f_2(x, y, p_2) = \beta \cdot x \cdot y - \delta \cdot y \quad (14)$$

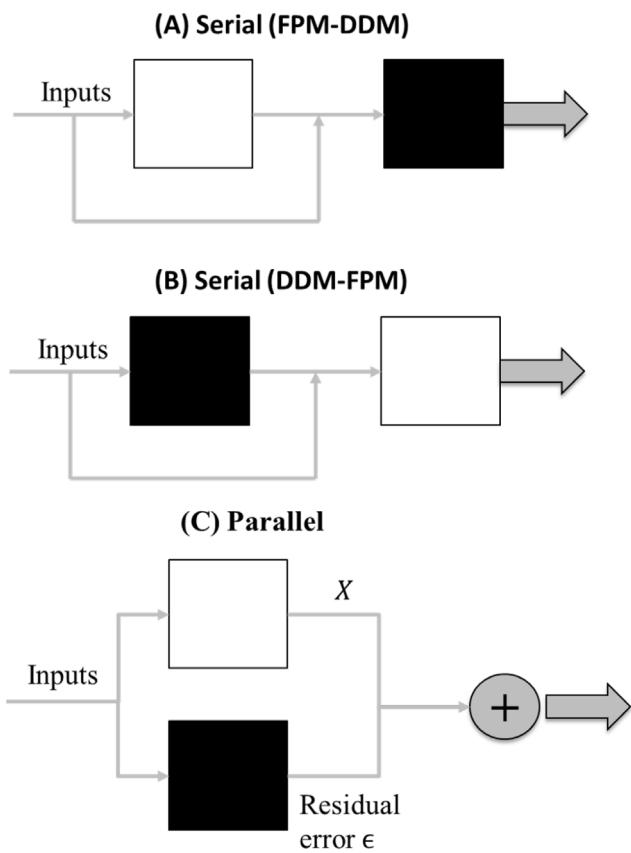


Fig. 3. Hybrid model structures with different configurations: (A) Series (FPM followed by DDM); (B) Series (DDM followed by FPM); and (C) Parallel.

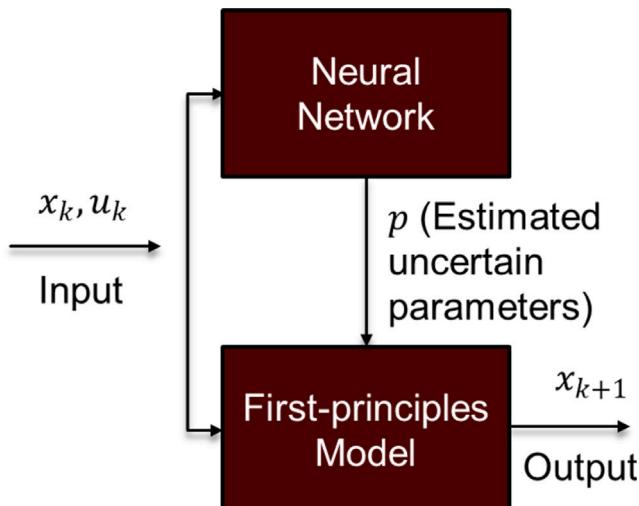


Fig. 4. NN-based hybrid model.

Here, x and y represent the populations of two species, p_1 and p_2 are parameters dictating growth rates, α , β , and δ are interaction coefficients, and f_1 and f_2 describe the interactions between these species.

Hybrid modeling involves using a DNN to either augment or replace components of an ODE system. For example, if the interaction term $\alpha \cdot x \cdot y$ in the ODE system is poorly understood, a DNN can be trained to approximate this term using experimental data:

$$\frac{dx}{dt} = p_1 \cdot x \cdot (1 - x) - g(x, y) \quad (15)$$

$$\frac{dy}{dt} = \beta \cdot x \cdot y - \delta \cdot y \quad (16)$$

In this case, $g(x, y)$ represents the output of a DNN designed to predict the interaction term, taking the state variables x and y as input. This approach allows for modeling complex interactions that are difficult to capture with traditional ODEs alone. This approach leverages the best of both worlds: the interpretability and theoretical underpinnings of ODEs and the flexibility and learning capabilities of DNNs, which include the different types of algorithms and activation functions that can be used for hybrid model training. If the input and output of the DNN vary over an order of magnitude, significant issues can arise during the training process. It is advisable to scale the inputs and outputs or render them dimensionless to mitigate these challenges.

We use an input-output training data set $(u_1, y_1), (u_2, y_2), \dots, (u_Q, y_Q)$ to train the hybrid model which includes a DNN to approximate the unknown parameters p . The inputs u are presented to the hybrid model, including the first principles model and the DNN. The inputs to the DNN propagate through its layers, resulting in the network's outputs at the final layer. Eqs. (2)–(4) are employed to compute the DNN outputs corresponding to the predicted parameter values p . These predictions are subsequently used as inputs to the first principles model to determine the outputs of the hybrid model, denoted as y' . For the hybrid model, the squared prediction error of the output across all Q training patterns is minimized as follows:

$$\hat{V} = \frac{1}{Q} \sum_{q=1}^Q (e_q)^T (e_q) \quad (17)$$

where

$$e_q = y_q - y'_q \quad (18)$$

The DNN's output y_q does not explicitly appear in the above error equation as it is generated and utilized internally in the hybrid model. In order to update the parameters of DNN using Eqs. (13) and (14), the effect of the DNN's output y_q on the prediction error of the hybrid model e_q needs to be quantified. For this purpose, we utilize finite difference methods to calculate the gradient of the hybrid model's output y'_q with respect to the DNN's output p_q . Hence, we obtain the following equations:

$$\frac{\partial e_q}{\partial y'_q} = -1 \quad (19)$$

$$\frac{\partial y'_q}{\partial p_q} = \frac{y'_{q+1} - 2y'_q + y'_{q-1}}{p_{q+1} - p_{q-1}} \quad (20)$$

and it follows that

$$\frac{\partial e_q}{\partial p_q} = \frac{\partial e_q}{\partial y'_q} \frac{\partial y'_q}{\partial p_q} = - \frac{y'_{q+1} - 2y'_q + y'_{q-1}}{p_{q+1} - p_{q-1}} \quad (21)$$

Let us define the sensitivity of the error e_q to changes in the net input of unit i in layer k as:

$$\delta_q^k(i) = \frac{\partial e_q}{\partial n_q^k(i)} \quad (22)$$

Now, the above equation can be rewritten using Eq. (4) as:

$$\delta_q^k(i) = \frac{\partial e_q}{\partial n_q^k(i)} = \frac{\partial e_q}{\partial a_q^k(i)} \frac{\partial a_q^k(i)}{\partial n_q^k(i)} = \frac{\partial e_q}{\partial a_q^k(i)} f^k(n_q^k(i)) \quad (23)$$

For the last layer M , the above equation leads to:

$$\delta_q^M = \frac{\partial e_q}{\partial A_q^M} f^M(n_q^M) \quad (24)$$

But the output A_q^M from the final layer M is the predicted parameter p_q , which results in

$$\frac{\partial e_q}{\partial p_q} = \frac{\partial e_q}{\partial A_q^M} \quad (25)$$

Using Eqs. (21)–(25), the δ_q^M value can be calculated. The Jacobian matrix encompasses the sensitivities of the error e_q with respect to

variations in the hyperparameters of the DNN, specifically the weights W^k and biases B^k . Mathematically, these sensitivities are expressed as $\frac{\partial e_q}{\partial W^k}$ and $\frac{\partial e_q}{\partial B^k}$, and can be computed for the neurons in the final layer M using δ_q^M and Eq. (2), as follows:

$$\frac{\partial e_q}{\partial W^M} = \delta_q^M A_q^{M-1} \quad (26)$$

$$\frac{\partial e_q}{\partial B^M} = \delta_q^M \quad (27)$$

For other layers, $k = 1, \dots, M - 1$, δ_q^k value can be calculated using the following recurrence relation:

$$\delta_q^k = F^k(r_q^k) W^{k+1T} \delta_q^{k+1} \quad (28)$$

Using the value of δ_q^k , the Jacobian matrix can be computed like that used for deriving Eqs. (26) and (27). Once the Jacobian matrix is calculated, the parameters of the DNN can be updated using Eq. (11). The Levenberg–Marquardt training algorithm begins with an initial value for μ , multiplied by a factor β whenever an update increases \hat{V} . Conversely, when an update reduces the \hat{V} value, μ is divided by β . The training algorithm continues iteratively until the \hat{V} value reaches a predefined tolerance.

2.3.3. Sensitivity analysis

One central aspect we have not discussed yet is what exactly these parameters p are and how they are selected. These unknown parameters p are any physically interpretable parameters used in FPMs. There might be very complex models with hundreds of these parameters (Lee et al., 2019). A sensitivity analysis-based approach is paramount to selecting which ones to estimate using the DNN. A sensitivity analysis helps understand which parameters significantly influence the outputs, and a local or global sensitivity analysis (GSA) can be performed.

Local sensitivity analysis around the nominal values of the model parameters is initially carried out to understand how the model parameters and initial conditions influence the different outputs. A sensitivity matrix is first derived from developing the parameter set, which affects the outputs (Chu and Hahn, 2007; Lee et al., 2018). To capture the effect that these parameters have on the outputs, a criterion called the Fisher Information Matrix (FIM) is calculated in the form of a sensitivity matrix as follows:

$$FIM = S^T \Sigma S \quad (29)$$

where Σ is an identity matrix of size $n \times n$ (where n represents the number of measurements or outputs), and S is the sensitivity matrix with dimensions $n \times p$ (where p represents the number of parameters being estimated). The FIM itself is a matrix of size $p \times p$, meaning it encapsulates the sensitivity of p parameters across the n outputs. If the goal is to select a single parameter that provides the most information, we can evaluate the FIM for each parameter individually. By extracting the i th column S_i of the sensitivity matrix, we can compute the FIM for each parameter as:

$$FIM_i = S_i^T \Sigma S_i \quad (30)$$

where S_i is the sensitivity of the system with respect to the i th parameter. Since Σ is typically an identity matrix, the FIM for a single parameter simplifies to:

$$FIM_i = S_i^T S_i \quad (31)$$

This scalar value represents the Fisher information for the i th parameter. A specific criterion is required to evaluate the information in the FIM, and the D-optimality criterion (ϕ_D) is used for this purpose. It minimizes the logarithm of the determinant of the inverse of the FIM. Using the inverse determinant property:

$$\phi_D^* = \max \phi_D(FIM) = \max \log \det(FIM) \quad (32)$$

This criterion helps identify the most informative parameters by maximizing the determinant of the FIM, thus ensuring the sensitivity analysis considers the full range of parameters and their impact on the model's output.

For this sensitivity analysis, one parameter is evaluated at a time, and ϕ_D is computed to show its effect on output. A higher ϕ_D value implies that the concerned model parameter influences the output more. As the process states in the kinetic model are measurable, these states are the model outputs y , i.e., $y = x$, where x is the state. It is important to note that this is one approach, and different sensitivity analysis approaches can be utilized to identify the most critical parameters p .

Given the significant parameter variation across different datasets, operating conditions, and measurement changes, conducting a GSA is crucial. This analysis assesses the output sensitivity over a wide range of parameter values. By running multiple experimental datasets with slight changes in initial parameter guesses and bounds, one can determine the parameter range for the global sensitivity model. From these runs, the estimated parameter values establish an overall range for each parameter and initial condition, with a 20%–50% margin of error to cover the maximum possible range. A comprehensive range is preferred to account for varying parameter estimates across different datasets and to evaluate how outputs respond to parameter values far from their nominal values.

For the GSA, *lhsdesign* (Latin hypercube sampling) is one method that can be used to construct a matrix of random values between 0 and 1 for each parameter. One hundred or more different cases of parameter sets can be considered for this analysis and are averaged at the end. All output effects are considered together in the combined GSA of the developed kinetic model. The combined GSA of the model and the range of each parameter is as shown below:

$$\phi_D(model) = \phi_D(State 1) + \phi_D(State 2) + \phi_D(State 3) + \dots \quad (33)$$

$$Range of each parameter = [LB \quad LB + (UB - LB) \cdot lhsdesign[0, 1]] \quad (34)$$

Incorporating dynamic process data into FPMs via hybrid modeling approaches significantly boosts prediction accuracy. Traditional FPMs often rely on assumptions and simplifications that, while necessary for tractability, can distance the model from real-world complexities. Dynamic process data, rich in detail and reflective of the system's actual behavior, offers a wealth of information that can be harnessed to bridge this gap. By dynamically updating model parameters or structures based on real-time or historical data, hybrid models can adapt to changes in system states or operating conditions (Lee et al., 2020b). This adaptability ensures that the model remains closely aligned with the underlying physical processes it seeks to represent, thereby enhancing the accuracy of its predictions. Furthermore, the iterative process of integrating process data helps identify and correct discrepancies between model predictions and observed outcomes, leading to continuous model improvement. As such, hybrid models hold significant promise for advancing our understanding of complex phenomena in science and engineering.

3. Step-by-step implementation of hybrid modeling framework

Section 3 provides the methodology proposed by researchers to build a NN-based hybrid model, giving a brief background about DNNs and the backpropagation algorithm used to train them (Shah et al., 2022). The section describes the components of hybrid models, including theoretical equations and ML algorithms, and further explains how these components interact within the hybrid framework. This section of the review will discuss the foundational aspects of this approach. Initially, the tables below will be used to describe the steps involved in the different blocks of hybrid modeling, and then the detailed code will be discussed. The pseudo-code for integrating a DNN into a hybrid model is a series of steps that take the output of a pre-trained DNN and combine it with other inputs to create a model informed by ML and domain-specific knowledge, such as physics laws or empirical equations. We break down each step in detail.

The fermentation process is taken as an example to demonstrate the steps to build a hybrid model. The simulation of the fermentation

Table 2

Simulation of fermentation process using ODEs.

Step	Description/Code
1	$X_0, P_0, S_0 \leftarrow \text{Initial_Conditions}['biomass'], \text{Initial_Conditions}['product'], \text{Initial_Conditions}['substrate']$
2	$t_{start}, t_{end}, dt \leftarrow \text{Time_Frame}['start'], \text{Time_Frame}['end'], \text{Time_Frame}['delta_t']$
3	Biomass, Product, Substrate $\leftarrow [X_0], [P_0], [S_0]$ Time $\leftarrow [t_{start}]$
4	For t from t_{start} to t_{end} with step dt do: a. $\mu \leftarrow \mu_{max} \times (\text{Substrate}[t] / (K_s + \text{Substrate}[t]))$ b. $dXdt \leftarrow \mu \times \text{Biomass}[t] - k_d \times \text{Biomass}[t]$ c. $dPdt \leftarrow k_p \times \text{Biomass}[t]$ d. $dSdt \leftarrow -(1/Y_{xs}) \times \mu \times \text{Biomass}[t]$ e. $X_{new} \leftarrow \text{Biomass}[t] + dXdt \times dt$ f. $P_{new} \leftarrow \text{Product}[t] + dPdt \times dt$ g. $S_{new} \leftarrow \text{Substrate}[t] + dSdt \times dt$ h. Biomass.append(X_{new}) i. Product.append(P_{new}) j. Substrate.append(S_{new}) k. Time.append($t + dt$)
5	Return Time, Biomass, Product, Substrate

Table 3

Hybrid model structure for fermentation.

Step	Description/Code
1	Load fermentation data into a DataFrame (DF) structure.
2	DF \leftarrow preprocess(DF)
3	Parameters \leftarrow initialize()
4	RPM, pH, Air, DO, O ₂ \leftarrow interpolate(DF)
5	Parameters \leftarrow set_targets()
6	DNN \leftarrow define_DNN_architecture()
7	DNN \leftarrow compile_and_train(DNN, DF)
8	HybridModel \leftarrow integrate_DNN_ODE(DNN, ODEs)
9	HybridModel \leftarrow compile_and_train_hybrid(HybridModel)
10	Predictions \leftarrow predict(HybridModel)

process begins with setting initial conditions for biomass (X_0), product (P_0), and substrate (S_0) as shown in **Table 2**. These initial conditions are essential for starting the simulation accurately. In Step 2, the time frame for the simulation is then defined, specifying the start time (t_{start}), end time (t_{end}), and time step (dt). In Step 3, the arrays are initialized to store the values of biomass, product, and substrate over the simulation period. The time array is also initialized to keep track of the simulation time. In Step 4, a loop is run over the defined time frame, updating the values of biomass, product, and substrate at each time step using the ODEs. In each iteration of the loop, the specific growth rate (μ) is calculated using the Monod equation. The rate of change of biomass ($dXdt$), product ($dPdt$), and substrate ($dSdt$) are then computed. These rates are used to update the current values of biomass, product, and substrate. The updated values are appended to their respective arrays, and the time is incremented by the time step (dt). Once the loop completes, the arrays containing the time, biomass, product, and substrate values are returned in Step 5. These results provide a detailed simulation of the fermentation process, capturing the system's dynamics over the specified time frame.

Once the FPM is defined, it is vital to look at the overall structure of the hybrid model as shown in **Table 3**. Step 1 starts by loading the fermentation data into a DataFrame (DF) structure. This step ensures that all relevant data is organized and ready for preprocessing. The data is then preprocessed to handle missing values, normalization, and other necessary cleaning tasks to prepare the data for model training in Step 2. In Step 3, the initial parameters required for the fermentation process are initialized. These parameters include initial conditions and other relevant factors influencing fermentation. In Step 4, interpolation is performed on the data to obtain continuous values for process variables such as RPM, pH, airflow rate, dissolved oxygen (DO), and O₂. These interpolated values are crucial for the accurate modeling of the fermentation process. After interpolating the data, target values for the model

Table 4

DNN integration into hybrid model.

Step	Description/Code
1	Time_Input \leftarrow DEFINE_LAYER(shape=(1,), name='time')
2	Init_State_Vars \leftarrow DEFINE_LAYER(shape=(3,), name='initial_variables')
3	Inlet_Flow \leftarrow DEFINE_LAYER(shape=(1,), name='inlet_flow')
4	Merged_Layer \leftarrow CONCATENATE(DNN.output, Time_Input, Init_State_Vars)
5	Integrated_Layer \leftarrow CUSTOM_LAYER(Merged_Layer, function=custom_op)
6	Reactor_States_Output \leftarrow DEFINE_LAYER(shape=(3,), activation='relu', name='reactor_states', non_trainable=True)
7	Hybrid_Model \leftarrow ASSEMBLE_MODEL(inputs=[DNN.input, Time_Input, Init_State_Vars], outputs=[Reactor_States_Output])

parameters are set in Step 5. These targets guide the training process to ensure the model achieves the desired performance. The architecture of the DNN is then defined in Step 6, specifying the layers, neurons, activation functions, and other architectural details. In Step 7, the DNN is compiled and pre-trained using the preprocessed data. The inputs for this pre-training process typically consist of features derived from the system being modeled, such as process variables (e.g., temperature, pressure, concentration levels) or time-series process data for the states. The outputs correspond to the nominal parameters of the ODE that need to be optimized further during the hybrid model training. These nominal values are from literature studies or process knowledge. This pre-training process involves optimizing the weights and biases of the DNN to minimize the initial prediction error between the predicted outputs and the actual target values from the training data. This step helps ensure that the ODEs do not suffer from ill-conditioning or convergence issues. In Step 8, the pre-trained DNN is integrated with the ODEs to form the hybrid model. The interaction between the DNN and the ODEs is facilitated by passing certain parameters predicted by the DNN, such as reaction rates, yield coefficient, or other unknown parameters (i.e., the parameter p), into the ODEs. These parameters, which are difficult to model explicitly using FPMs, are refined by the DNN based on the training data. Once these parameters are predicted by the DNN, they are used within the ODEs to compute the evolution of the system's state variables over time. This iterative exchange allows the hybrid model to make more accurate predictions. In Step 9, the hybrid model is compiled and trained to fine-tune the integration of DNN and ODEs in a tandem manner. Finally, in Step 10, the trained hybrid model is used to make predictions on the fermentation process, providing insights into the behavior and outcomes of the process under various conditions.

Integrating FPMs and DNNs begins by defining the input layer for the time variable as shown in **Table 4**. This layer is crucial for incorporating the temporal aspect of the fermentation process into the hybrid model. Next, Step 2 defines the input layer for the initial state variables. These variables include initial conditions such as biomass, product, and substrate concentrations. An input layer for the inlet flow is now defined in Step 3 to account for the flow rates into the system, which is the manipulated variable. These input layers are then concatenated with the output of the pre-trained DNN in Step 4. This merged layer combines the strengths of the data-driven DNN and the physics-based first-principles components. A custom layer function is applied to the merged layer in Step 5, allowing additional domain-specific knowledge and empirical equations to be incorporated into the hybrid model. In Step 6, the output layer for the reactor states is then defined with non-trainable parameters. This layer provides the final output of the hybrid model, representing the states of the reactor. Finally, the hybrid model is assembled using the defined input and output layers in Step 7.

The assembled model integrates the DNN and the ODEs, leveraging the strengths of both approaches. This integration enhances the prediction accuracy and interpretability of the model, providing a comprehensive framework for modeling complex fermentation processes. Github link to a code for developing a hybrid model with spatiotemporally varying parameters is mentioned here: [Github code](#) The detailed code block is given in Listing 1.

```

1 % =====
2 % Step 1: Import Necessary Libraries
3 % =====
4 import numpy as np
5 import pandas as pd
6 from scipy.integrate import solve_ivp
7 import tensorflow as tf
8 from tensorflow.keras.models import Model
9 from tensorflow.keras.layers import Input, Dense
10 from sklearn.model_selection import train_test_split
11 from sklearn.preprocessing import StandardScaler
12
13 % =====
14 % Step 2: Load and Prepare Data
15 % =====
16 % Load dataset
17 df = pd.read_csv(your_dataset.csv)
18
19 % Preprocess the dataset: Normalize, handle missing values, etc.
20 % Example: Normalize features
21
22 scaler = StandardScaler()
23 features = scaler.fit_transform(df[[feature1, feature2]].values)
24 target = df[target].values
25
26 % Split the data into training and testing sets
27 X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2,
28 random_state=42)
29
30 % =====
31 % Step 3: Define First Principles Model
32 % =====
33 def first_principles_model(t, y, params):
34     """
35         Define the ODEs for the FPM.
36         t: time
37         y: state variables as a NumPy array
38         params: a dictionary of parameters
39         Returns the derivatives of the state variables.
40     """
41     % Example model: simple decay process (Enter kinetic model here)
42     % dy/dt = -k * y where k is the parameter
43     dydt = -k * y
44     return dydt
45
46 % =====
47 % Step 4: Define the DNN
48 % =====
49 def create_dnn(input_shape):
50     """
51         Define and compile a DNN model.
52         Input_shape: Shape of the input data
53         Returns a compiled Keras model.
54     """
55     Define the DNN here
56     inputs = Input(shape=input_shape)
57     x = Dense(128, activation=relu)(inputs)
58     x = Dense(128, activation=relu)(x)
59     outputs = Dense(1, activation=linear)(x)
60     model = Model(inputs=inputs, outputs=outputs)
61     model.compile(optimizer=adam, loss=mse)
62     return model

```

Remark 1. In line with the goal of providing a comprehensive tutorial, we have opted to include the code blocks directly within the text rather than as external links. This decision stems from our experience with students and researchers who, despite pseudocode availability, have faced difficulties in independently developing hybrid modeling solutions. Including the full code ensures a more seamless learning process and eliminates the need for readers to navigate between the

```

1
2 % =====
3 % Step 5: Pretraining DNN
4 % =====
5 % Create a DNN model
6 dnn_model = create_dnn((X_train.shape[1],))
7 % Train the DNN
8 dnn_model.fit(X_train, y_train, batch_size=32, epochs=100, validation_split=0.1)
9
10 % =====
11 % Step 6: Combine FPM and DNN into a Hybrid Model
12 % =====
13 % This step is highly specific to the application.
14 % An example approach could be using the DNN to predict parameters for the FPM.
15 % Readers will need to customize this step to fit their specific problem.
16
17 def hybrid_model(t, y):
18     """
19         A hybrid model combining FPM and DNN predictions.
20         t: time
21         y: state variables
22         Uses the DNN to predict parameters and integrates this into the FPM.
23     """
24
25     % Example: Use DNN to predict parameter k
26     k_predicted = dnn_model.predict(np.array([[t, y]]))[0]
27     params = {k: k_predicted}
28     % Integrate the FPM with predicted parameters
29     dydt = first_principles_model(t, y, params)
30     return dydt
31
32 % =====
33 % Step 7: Train the Hybrid Model with a custom layer
34 % =====
35 % Define the training loop, loss functions, etc.
36 % This will likely involve solving the FPM with the integrated DNN predictions and comparing
37 % the results to the training data to adjust the DNN.
38
39 % Define a custom training loop
40 def custom_training_loop(hybrid_model, X_train, y_train, epochs=100, batch_size=32):
41     """
42         Custom training loop for the hybrid model.
43         hybrid_model: The combined FPM-DNN hybrid model.
44         X_train: Training input data.
45         y_train: Training target data.
46         epochs: Number of training epochs.
47         batch_size: Size of each training batch.
48     """
49
50     % Initialize optimizer
51     optimizer = tf.keras.optimizers.Adam()
52
53     % Iterate over the number of epochs
54     for epoch in range(epochs):
55         print(f'Epoch {epoch+1}/{epochs}')
56
57         % Shuffle the training data at the beginning of each epoch
58         indices = np.arange(X_train.shape[0])
59         np.random.shuffle(indices)
60         X_train = X_train[indices]
61         y_train = y_train[indices]
62
63         % Iterate over batches
64         for i in range(0, X_train.shape[0], batch_size):
65             X_batch = X_train[i:i+batch_size]
66             y_batch = y_train[i:i+batch_size]
67
68             with tf.GradientTape() as tape:
69                 % Forward pass
70                 predictions = hybrid_model(X_batch, training=True)

```

```

1      % Compute loss
2      loss = tf.reduce_mean(tf.square(y_batch - predictions))
3
4      % Backward pass and update weights
5      gradients = tape.gradient(loss, hybrid_model.trainable_variables)
6      optimizer.apply_gradients(zip(gradients, hybrid_model.trainable_variables))
7
8      % Print loss after each epoch
9      print(f'Loss: {loss.numpy()}')
10
11
12 % Instantiate and train the hybrid model using the custom training loop
13 custom_training_loop(hybrid_model, X_train, y_train)
14
15 % =====
16 % Step 8: Evaluate the Model
17 % =====
18 % Evaluate hybrid model performance on test data.
19 % This might involve comparing predicted trajectories with actual trajectories, computing
20   error metrics, etc.
21
22 % Note: Steps 6 and 7 are highly conceptual and will require significant customization based
  on the specifics of the system and the nature of the data.

```

Listing 1 Hybrid Modeling Template Code

manuscript and external resources. By presenting the code alongside explanations, we aim to support a broader audience in understanding and adopting hybrid modeling techniques more effectively

Step 1 in constructing a hybrid model involves importing essential libraries. This includes ‘*numpy*’ for numerical operations, ‘*pandas*’ for data manipulation, ‘*scipy.integrate.solve_ivp*’ for solving initial value problems for systems of ODEs, and ‘*TensorFlow*’ along with ‘*keras*’ for building and training deep learning models. Additionally, ‘*scikit-learn*’ libraries are used to preprocess and split the dataset. Step 2 involves loading and preparing the dataset, which is critical for modeling. The data is loaded, and preprocessing steps are applied to normalize features and handle missing values. For example, features are normalized using ‘*StandardScaler*’ to ensure a mean of zero and a standard deviation of one. The dataset is then split into training and testing sets using ‘*train_test_split()*’ to evaluate the model’s performance on unseen data.

In Step 3, the FPM is defined using a function that describes the system’s behavior through differential equations. In this example, a simple decay process is modeled, where the rate of change of the state variable is proportional to its current value. The parameter k is unknown and will later be estimated using a DNN. The process can be seamlessly replaced by any other process as long as the equation structure is known. A DNN is created to learn from the data and predict unknown parameters for the FPM. The network is constructed using Keras, with an input layer, multiple hidden layers with ReLU activation functions, and an output layer with a linear activation function, as outlined in Step 4. The model is compiled using the Adam optimizer, which is known for its efficiency in handling sparse gradients and adaptive learning rates. During hybrid model training, the mean squared error (MSE) loss is minimized. The MSE quantifies the difference between the predicted and actual parameter values by averaging the squared differences, helping the model fine-tune its predictions for better accuracy. This helps the model minimize prediction errors during training, ensuring that the DNN closely approximates the unknown parameters to refine the FPM’s performance. Key structural hyperparameters — such as the number of hidden layers, the number of neurons per layer, batch size, and learning rate — are optimized using a combination of grid search, cross-validation, and Keras Tuner, which automates the search for the best model architecture. Keras Tuner explores a wide range of hyperparameter values and selects the optimal set based on performance metrics like validation loss or MSE during training. The DNN is trained

on the normalized features and target values with a specified batch size, number of epochs, and a validation split to monitor its performance during training. An optional Step 5 can be added to train the DNN separately, providing insights into its performance before integrating it into the hybrid model. By using Keras Tuner, the model’s architecture is fine-tuned to improve predictive accuracy while minimizing overfitting.

Step 6 involves integrating the DNN predictions into the FPM to create a hybrid model. The DNN is used to predict unknown parameters for the FPM dynamically. In this example, the DNN predicts the parameter k based on the current time and state variables. These predictions are then used in the FPM to compute the rate of change of the state variables. Training the hybrid model involves solving the FPM using the parameters predicted by the DNN and adjusting the DNN based on the discrepancies between the predicted and actual trajectories, as shown in Step 7. This process requires defining a training loop and loss functions tailored to the specific application, ensuring the model learns to accurately predict the system’s behavior. Finally, the hybrid model’s performance is evaluated on the test data in Step 8. This involves comparing the predicted trajectories with actual trajectories and computing error metrics to assess how well the model generalizes to unseen data. The evaluation helps fine-tune the model and validate its effectiveness in capturing the system’s dynamics.

Remark 2. One significant challenge in implementing hybrid models on platforms like TensorFlow is the suboptimal performance of built-in ODE solvers when handling dynamic process models. For instance, TensorFlow’s ODE solvers often struggle with stiff systems, requiring sophisticated workarounds such as custom ODE solvers or integration of external libraries like SciPy’s solvers to achieve efficient and accurate simulation of dynamic processes

4. Application: Extension of hybrid modeling framework

This section offers an in-depth exploration of the application of hybrid modeling across diverse fields of engineering and science. It elucidates how hybrid models are employed to develop robust, adaptive, and precise solutions for complex challenges in process modeling, control, monitoring, model discovery, and plant scale-up. By integrating the strengths of both first-principles and data-driven approaches, hybrid models significantly enhance overall system performance, reliability, and efficiency, addressing intricacies that traditional methods

alone cannot effectively manage. Through detailed case studies and examples, this section demonstrates the transformative impact of hybrid modeling in improving predictive accuracy, optimizing control strategies, and enabling real-time adaptive monitoring across various industrial and scientific domains. The application of hybrid modeling can be divided into two parts: systems governed by ODEs and systems governed by PDEs. Based on this, this section also discusses two case studies for both systems.

4.1. Adoption of hybrid modeling in engineering and science

Hybrid models, which blend traditional FPMs with DDMs, are increasingly being adopted in various areas such as process modeling, process control, process monitoring, model discovery, and plant scale-up. In complex process modeling, hybrid models enhance predictive accuracy and extrapolation by integrating mechanistic knowledge with empirical data, allowing for the simulation of intricate chemical and physical processes, parameter estimation, and the capturing of nonlinear system behaviors. In process control, hybrid models improve model predictive control (MPC) frameworks, adapt to changing dynamics for adaptive control, and ensure fault tolerance by detecting and compensating for system faults. For monitoring, hybrid models are crucial in anomaly detection, state estimation, and predictive maintenance, providing accurate diagnostics and proactive maintenance strategies. Additionally, hybrid models facilitate model discovery and plant scale-up by leveraging ML to identify underlying chemical mechanisms and optimize the transition from pilot to industrial scale. Tables 5, 6, and 7 give a descriptive encapsulation of various challenges addressed by hybrid models and the corresponding studies.

4.1.1. Process modeling

This section delves into the integration of FPM and DDM in process modeling, highlighting the primary challenges in traditional modeling that hybrid models can address. Hybrid modeling approaches vary widely, employing structures from simple algebraic equations to complex PDEs, enhancing predictive accuracy, operational efficiency, and interpretability across various industrial and plant-scale applications. These challenges include handling nonlinearities in complex dynamics, optimizing process performance, estimating unmeasurable parameters, and improving predictive accuracy with limited data. Moreover, hybrid models enhance robustness and extrapolation capabilities, integrate multiple data sources seamlessly, reduce computational complexity, and adapt to real-time changes effectively. In this section, we delve into these challenges in detail, illustrating how hybrid models provide robust, accurate, and efficient solutions for complex process modeling applications.

A primary challenge that many real-world processes exhibit is highly nonlinear and complex behaviors that are difficult to capture accurately with traditional FPMs and DDMs alone. These models often fail to represent the intricate interactions and nonlinearities inherent in such systems, leading to suboptimal performance and inaccuracies in predictions. Hybrid models can effectively capture these nonlinearities and complex dynamics as shown in Fig. 5. For instance, in reaction engineering, series hybrid modeling frameworks have been used to determine dissolution rates and dynamic concentrations of reactants by tracking the nonlinear reaction kinetics of industrial reactors (Bhutani et al., 2006; Chen and Ierapetritou, 2020). This showcases the model's ability to handle complex reaction systems with enhanced predictive capabilities (Simon et al., 2006; Zahedi et al., 2011).

Traditional optimization methods might not efficiently explore the vast parameter space, especially in complex systems with numerous interacting variables. This inefficiency can lead to suboptimal solutions and missed opportunities for process improvement. Hybrid models combined with advanced optimization techniques, such as genetic algorithms (GA) or particle swarm optimization (PSO), can optimize process performance more effectively. For example, a hybrid model

Table 5
Summary of process modeling studies.

Challenge	Specific process
Handling nonlinearities and complex system dynamics	<ul style="list-style-type: none"> Hybrid model is used to predict the heterotrophic growth of Chlorella (Wu and Shi, 2007). Multi-objective particle swarm optimization based on Pareto dominance hybrid algorithm is integrated with ANN for modeling industrial cracking furnace (Li et al., 2007). The hybrid model predicts the flux evolution and duration of cross-flow ultrafiltration processes for various proteins and membranes (Krippl et al., 2020). The hybrid model captures process nonlinearities and complexities of the reaction systems (Zendehboudi et al., 2014; Lauret et al., 2000; Simon et al., 2006; Zahedi et al., 2011). Gas production rates were predicted by hybrid model for fluidized bed gasifier (Guo et al., 2001).
Improving predictive accuracy with limited data	<ul style="list-style-type: none"> Viable cell concentration of mammalian cell cultivation is estimated using the hybrid model (Aehle et al., 2010; Dors et al., 1995). Hybrid models can be used to rate the performance of industrial HDS reactors (Bellos et al., 2005).
Optimizing process performance	<ul style="list-style-type: none"> Hybrid models have been proposed from a modeling and prediction perspective which utilize design of experiments (DoE) data (Bayer et al., 2020, 2021). Hybrid models have been used for optimization of ethanol production by a flocculating yeast grown on cashew apple juice (da Silva Pereira et al., 2021). Overall accuracy of process predictions is shown to be improved for applications like activated sludge process, enzymatic synthesis, solvent flux modeling, and wastewater systems (Côte et al., 1995; Anderson et al., 2000; Lee et al., 2002; Santos et al., 2007; Silva et al., 2008).
Robustness and Extrapolation capability	<ul style="list-style-type: none"> Modeling of the enzymatic conversion of penicillin G (Van Can et al., 1999). Hybrid models have been shown to be more accurate and robust than FPMs in processes like continuous PHA production, chromatography processes, and crossflow microfiltration (Luna et al., 2021; Nagrath et al., 2004; Wang et al., 2017; Piron et al., 1997).
Reducing computational complexity	<ul style="list-style-type: none"> Hybrid models are used for perfusion cultures of hybridoma cells (Maton et al., 2022). Coupling of universal differential equations (UDE) and population balance model (PBM) offers a less complex, effective model to capture essential aspects of crystallization kinetics and variables' dynamics (Lima et al., 2023).
Parameter estimation and handling poorly known parameters	<ul style="list-style-type: none"> Hybrid model predicts the kinetics-deactivation term or other time-varying terms vital for accurate modeling (Azarpour et al., 2017; Xiong and Jutan, 2002; Bellos et al., 2005). Poorly known kinetic parameters (nucleation rate, growth rate, agglomeration kernel) are replaced by a feedforward ANN (Georgieva et al., 2003; Sharma and Liu, 2022; Bishara and Li, 2023).
Managing high-dimensional and multivariable processes	<ul style="list-style-type: none"> Groundwater model simulations are subject to both aleatoric and epistemic errors, which the hybrid model eliminates (Xu and Valocchi, 2015).
Integrating multiple data sources and models	<ul style="list-style-type: none"> Hybrid hydroclimatic forecasting systems employ data-driven (statistical or ML) methods to harness and integrate a wide variety of predictions from dynamical, physics-based models into a final prediction product (Slater et al., 2023; Richardson et al., 2020; Essenfelder et al., 2020; Bennett et al., 2016).

Table 6

Summary of process control and process monitoring studies.

Applications	Challenge	Specific process
Process control	Managing nonlinear and complex dynamic systems	<ul style="list-style-type: none"> MPC implementation with Hybrid model for the control of monomer conversion and DO levels in sequencing batch reactors (SBRs) (Cubillos et al., 2001; Azwar et al., 2006).
	Improved predictive control	<ul style="list-style-type: none"> Predictive control of the jacket temperature to track set-points of product conversion (Vega et al., 1997). Controllers using PINNs have been used for bioprocess models (Rogers et al., 2023). Economic nonlinear model predictive control (eNMPc) to optimize the performance of an air separation unit (Schäfer et al., 2019).
	Optimal control input sequence determination	<ul style="list-style-type: none"> Direct control of a nonlinear semi-batch polymerization process is done by using inverse model for obtaining the input sequence (Ng and Hussain, 2004; Ferreira et al., 2014).
	Adaptive control and real-time adjustments	<ul style="list-style-type: none"> Hybrid model is used to get the input profiles for controlling PSDs (Doyle III et al., 2003). Physics-informed recurrent neural network (PIRNN)-based modeling approach is used for nonlinear dynamic systems (Zheng and Wu, 2023).
	Model-based optimal control strategies	<ul style="list-style-type: none"> Nonlinear optimization problem formulated using the hybrid model and solved for the optimal control trajectory (Ghosh et al., 2021; McKay et al., 2022; He et al., 2020).
	Computational efficiency	<ul style="list-style-type: none"> PINNs-based controllers capture system dynamics in a computationally tractable way and estimate model parameters from online data (Zheng and Wu, 2023; Alhajeri et al., 2022a).
	Complex systems with multiple variables and interactions	<ul style="list-style-type: none"> Hybrid models combining FPM and PCA identify faults in HVAC systems and devise sequencing control strategies for fault monitoring (Hassanpour et al., 2020).
	Capturing intricate and deep process features	<ul style="list-style-type: none"> Hybrid models integrating FPMs with deep learning techniques extract detailed process features crucial for monitoring processes like complex metallurgical, catalyst, and biochemical reactions (Bui et al., 2021; Sun et al., 2020; Safavi et al., 1999; Pinto et al., 2022).
	State estimation	<ul style="list-style-type: none"> Hybrid estimators combining DNN and EKF improve accuracy by eliminating errors from different sources. Applied in lithium-ion battery monitoring for accurate SOC estimation (Li et al., 2019; Bai et al., 2023; Qi and Moore, 2002; Narayanan et al., 2020).
	Adaptive monitoring	<ul style="list-style-type: none"> Hybrid models adapt to dynamic changes by continuously learning from real-time data and updating the monitoring strategy, improving accuracy in bioprocess conditions (Mahalec and Sanchez, 2012; Shah et al., 2023b).

(continued on next page)

Table 6 (continued).

Detecting faults early and providing accurate prognosis	<ul style="list-style-type: none"> Hybrid models leverage predictive capabilities and mechanistic understanding for early fault detection and prognosis, as seen in catalyst monitoring (Sun et al., 2020).
Enhanced state-space models	<ul style="list-style-type: none"> Hybrid models improve MHE performance by enhancing prediction accuracy and minimizing discrepancies between predicted and measured states (Xie et al., 2024; Li et al., 2023; Ye et al., 2022; Sohlberg, 2007).

Table 7

Summary of model discovery and plant scale-up studies.

Applications	Challenge	Specific process
Model discovery	Discovery of governing equations	<ul style="list-style-type: none"> ANN-based method has been developed to accurately denoise data and estimate partial derivatives, which is effective in identifying underlying PDE models (Lagergren et al., 2020).
		<ul style="list-style-type: none"> A PINN has been introduced and tailored for discovering thermodynamically consistent equations, demonstrating the use of hybrid models in shock hydrodynamics (Patel et al., 2022).
	Incorporation of physical constraints in DDMs	<ul style="list-style-type: none"> PINNs were developed to solve forward and inverse problems involving nonlinear PDEs and to learn linear differential equations using Gaussian processes, collectively enhancing the accuracy and robustness of models by embedding physical constraints into the ML process (Raissi et al., 2017, 2019).
Plant scale-up	Geometric and Kinetic nonlinearities	<ul style="list-style-type: none"> Hybrid models adjust for nonlinear scaling effects, such as changes in heat transfer and fluid dynamics, to maintain accurate predictions of reaction kinetics across different scales (Bollas et al., 2003; Michalopoulos et al., 2001; Acosta-López and de Lasa, 2023).
	Material and Energy balances	<ul style="list-style-type: none"> Hybrid models optimize material and energy inputs, adapting to the increased scale by accurately predicting key performance indicators like yield and energy efficiency (Bollas et al., 2003; Simon et al., 2006; Te Braake et al., 1998).
	Operational variabilities	<ul style="list-style-type: none"> In bioreactor operations, hybrid models dynamically adjust to variations in feedstock quality and environmental conditions, ensuring consistent output quality and process stability as scale increases (Bellos et al., 2005).

optimized the ethanol production process by integrating PSO with an ANN, leading to enhanced performance and efficiency (da Silva Pereira et al., 2021). By leveraging the strengths of both DDM and FPM, hybrid models can explore the parameter space more comprehensively and identify optimal operating conditions more reliably. Many processes have parameters that are difficult to measure or estimate accurately, leading to model inaccuracies and reduced reliability of the simulations and predictions. Hybrid models can estimate latent parameters using operating data, improving the accuracy of the models even when some parameters are poorly known. As an example, in systems with variations in operating conditions (Shah et al., 2022), feed quality, and fermentation growth characteristics, hybrid models accurately predict uncertain parameters such as heat and mass transfer coefficients and time-varying kinetic parameters (Azarpour et al., 2017; Xiong and

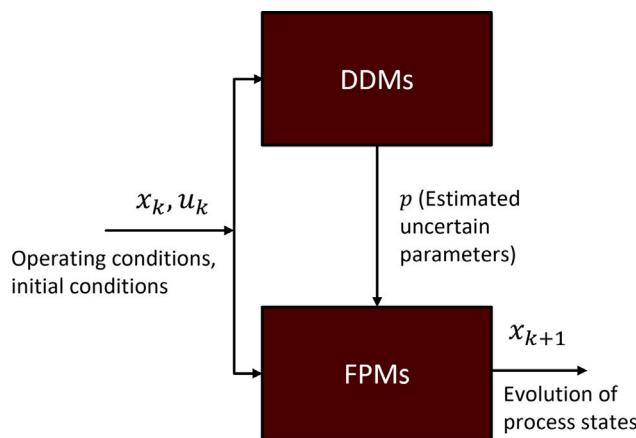


Fig. 5. Schematic for the use of hybrid models for process modeling.

Jutan, 2002; Bellos et al., 2005). Furthermore, due to the complexity of crystallization phenomena (Kwon et al., 2013), hybrid models have been used to estimate the growth and nucleation rates. Specifically, researchers have used complex DDMs for growth rate predictions instead of simple empirical expressions, which do not capture the complex crystal growth behaviors (Georgieva et al., 2003; Lima et al., 2023; Bishara and Li, 2023; Sitapure and Kwon, 2023c).

ML models typically require large amounts of data for training, which might not always be available, especially in industrial settings where data collection can be costly or time-consuming. Hybrid models require less training data while still enhancing predictive accuracy by leveraging the underlying mechanistic models. For instance, the hybrid model for viable cell concentration estimation in mammalian cell cultivation combines FPMs and DDMs, effectively utilizing available data to improve predictions (Wu and Shi, 2007; Aehle et al., 2010; Dors et al., 1995). This integration allows for accurate modeling even with limited datasets, making hybrid models highly practical for various applications. Hybrid models also offer better extrapolation capabilities by embedding first-principles knowledge into the model structure. This is evident in the modeling of the enzymatic conversion of penicillin G, where the hybrid model demonstrated good extrapolation properties (Van Can et al., 1999). By combining mechanistic insights with data-driven components, hybrid models can make reliable predictions even in uncharted territories, enhancing their robustness and applicability. Techniques, like intensified DoE (iDoE), have also been integrated with series hybrid modeling to expedite experimental processes and reduce costs, particularly at larger scales where experimentation becomes increasingly expensive. The iDoE provides more process response information in less overall process time, whereas hybrid modeling serves to describe this behavior effectively. Combining both approaches is beneficial for faster design screening and is particularly advantageous at larger scales, where costs per experiment rise significantly (Bayer et al., 2020, 2021).

Combining various data sources and models to improve predictions is complex and often leads to inconsistencies and integration challenges. Hybrid models can seamlessly integrate multiple data sources and models, providing a unified framework for comprehensive analysis. For example, hybrid hydroclimatic forecasting systems combine predictions from dynamical, physics-based models with data-driven methods, resulting in more reliable and comprehensive forecasts (Slater et al., 2023; Richardson et al., 2020; Essenfelder et al., 2020; Bennett et al., 2016). This integration allows for better utilization of available information and improves the overall accuracy and reliability of the predictions.

Adapting models to changing process conditions in real-time is challenging with static FPMs, which are not designed to handle dynamic changes and new data continuously. Hybrid models, particularly

those integrating adaptive learning algorithms, can adjust to new data and changing conditions in real-time. This adaptability is crucial for applications like biomass gasification, where the hybrid model predicts gas production rates under varying conditions (Guo et al., 2001).

Overall, hybrid modeling offers a powerful solution to the challenges faced by traditional FPMs and DDMs in process modeling. By integrating these two approaches, hybrid models effectively capture complex dynamics, optimize performance, and improve predictive accuracy even with limited data. Their adaptability to real-time changes and ability to integrate multiple data sources make them invaluable across various industrial applications. As these models evolve, they will play an increasingly vital role in advancing process efficiency and innovation.

4.1.2. Process monitoring

Hybrid models have emerged as a powerful tool in addressing the multifaceted challenges associated with process monitoring. By integrating the robustness of FPM with the adaptability and predictive power of ML techniques, hybrid models offer enhanced capabilities for fault detection, state estimation, and handling complex, high-dimensional data. These models excel in dynamic environments, where they adapt to real-time changes and improve the accuracy and reliability of predictions. Hybrid models reduce computational demands and can infer unmeasured states with minimal data, making them ideal for inferential monitoring and observer design.

Hybrid models enhance traditional methodologies, such as the Kalman filter and moving horizon estimator (MHE), by incorporating data-driven components, thereby mitigating plant-model mismatches and improving overall monitoring performance. These methods rely heavily on accurate models to estimate states and correct predictions. However, in real-world applications, plant-model mismatches often occur due to unmodeled dynamics, external disturbances, or uncertainties in system parameters. These mismatches can lead to degraded performance, inaccurate state estimations, and, ultimately, suboptimal process monitoring. By integrating ML techniques into these traditional frameworks, hybrid models can dynamically adapt to discrepancies between the model and the actual process. For instance, in a hybrid model-based Kalman filter, the data-driven component can adjust the process noise covariance or directly influence the state update equations, making the filter more resilient to unexpected variations. This integration has been particularly effective in improving state estimation accuracy, as demonstrated by hybrid model-based estimators, where the hybrid model is employed within the extended Kalman filter (EKF) framework to improve predictive monitoring accuracy and control as shown in Fig. 6. The combined hybrid-EKF approach yields superior results compared to traditional methods, such as partial least squares (PLS) models, particularly in predictive monitoring and soft sensor applications for titer estimation (Bai et al., 2023; Qi and Moore, 2002; Narayanan et al., 2020). Similarly, in an MHE framework, hybrid modeling techniques can refine the objective function or improve constraint handling, leading to more accurate and robust state estimations even under significant plant-model mismatches (Xie et al., 2024; Li et al., 2023; Ye et al., 2022; Sohlberg, 2007).

Traditional monitoring methods often struggle to capture the intricate and deep process features essential for understanding system behaviors and detecting potential faults. They rely on simplified models or direct analysis of high-dimensional data, which can miss intricate patterns, nonlinear interactions, and subtle anomalies. These methods may also lack the adaptability to respond to dynamic changes in process conditions, leading to reduced accuracy and reliability. Hybrid models offer a robust solution to this challenge. By transforming complex, high-dimensional process data into a lower-dimensional latent space, hybrid models can more effectively extract and analyze the most critical features. This transformation enhances the model's ability to manage complex interactions, improves predictive accuracy, and reduces uncertainty. For example, in the monitoring of cobalt removal within a zinc

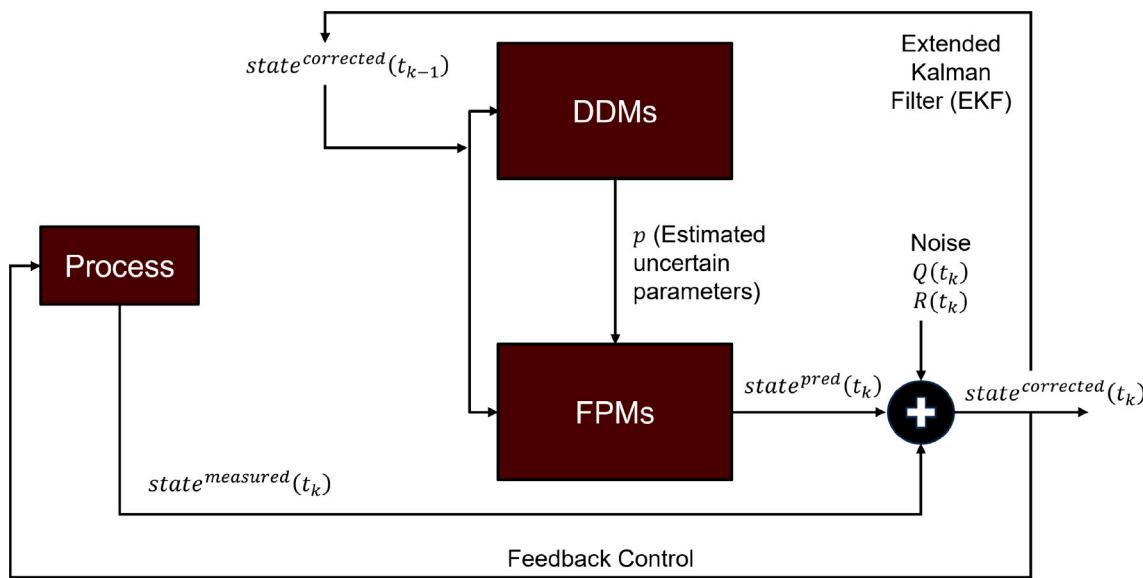


Fig. 6. Schematic for using hybrid models for process monitoring using Kalman filters.

hydrometallurgy unit, hybrid models that combine a state-space model with deep learning have significantly improved accuracy (Sun et al., 2020). More specifically, the hybrid model can focus on identifying patterns, correlations, and anomalies that might not be apparent in the raw, high-dimensional data. This approach allows the hybrid model to concentrate on key process features—such as reaction kinetics, energy transfer, or material interactions that are critical for accurate monitoring and control. By operating in the latent space, the DDM enhances the model's ability to respond to changes in process conditions, adapt to new data, and minimize uncertainties that could arise from incomplete or imperfect knowledge of the system. This approach has also proven effective in other complex processes, such as catalyst monitoring and biochemical reactions, where detailed feature extraction is crucial for accurate monitoring and fault detection (Safavi et al., 1999; Pinto et al., 2022; Bui et al., 2021).

Monitoring systems often face challenges when processing and analyzing high-dimensional data, as it can be computationally intensive and complex (Narasingam et al., 2023; Son et al., 2022b). Specifically, the data-driven component of a hybrid model addresses this by integrating dimension reduction techniques, such as PCA, with FPMs to simplify high-dimensional data without losing critical information. This approach is particularly effective in applications like fault monitoring of HVAC systems, where sensor data is condensed while preserving essential features (Hassanpour et al., 2020).

Beyond managing high-dimensional data, another significant challenge in process monitoring is the need for models that can adapt to dynamic changes in real-time. Static models, which are often used in traditional monitoring systems, may fail to adjust to changing process conditions, leading to undetected faults or false alarms. In contrast, hybrid models are designed to continuously learn from real-time data, updating their monitoring strategies accordingly. For example, a hybrid model combining FPM and ANN has been shown to adapt effectively to changes in bioprocess conditions, thereby improving monitoring accuracy (Mahalec and Sanchez, 2012; Shah et al., 2023b).

Hybrid models also offer significant advantages in early detection and prognosis by combining their predictive capabilities with a deep mechanistic understanding of the process. This integration allows hybrid models to not only monitor but also anticipate potential faults, as demonstrated in catalyst monitoring, where advanced algorithms identify issues before they can impact the process (Sun et al., 2020).

Overall, hybrid modeling approaches not only enhance the predictive capabilities of traditional methods but also offer greater flexibility

and robustness, ensuring that process monitoring remains accurate and reliable even under challenging conditions. This fusion of FPM and ML thus resolves issues of plant-model mismatches by continuously learning and adapting to the actual process dynamics, ultimately improving overall monitoring performance and decision-making in complex industrial systems.

4.1.3. Process control

The hybrid model's ability to integrate the FPM with a DDM allows it to address several key challenges in process control. This integration results in improved predictions over nominal FPMs, enhancing control strategies by managing nonlinearities, adapting to real-time changes, and improving robustness against model uncertainties and external disturbances.

Hybrid models significantly enhance predictive accuracy, addressing the challenge of managing nonlinear and complex dynamic systems in the context of process control. This is exemplified in the predictive control of jacket temperature, where hybrid models have been successfully applied to track set-points of product conversion with greater precision (Vega et al., 1997). The improved predictive capabilities of hybrid models facilitate the development of more robust control strategies that can adapt to real-time data and dynamic process conditions. For instance, MPC implementations have benefited from hybrid models in controlling monomer conversion and DO levels in sequencing batch reactors (SBR), resulting in enhanced process stability and efficiency (Cubillos et al., 2001; Azwar et al., 2006). Hybrid models have also been utilized for economic nonlinear model predictive control (eN-MPC) to optimize the performance of an air separation unit (ASU) in the context of fluctuating electricity prices, enabling the controller to make efficient, real-time decisions by reducing computational complexity while maintaining the accuracy required for dynamic process optimization (Schäfer et al., 2019). Traditional ML models often require large datasets for effective training, a challenge in industrial settings where data may be sparse or expensive to obtain. Hybrid models mitigate this issue by incorporating first-principles knowledge, thereby reducing the data requirements while still improving control performance. An example is the hybrid model used for obtaining input profiles to control particle size distributions (PSDs), which demonstrates effective control with limited data (Doyle III et al., 2003). In addition to enhancing control accuracy, hybrid models also improve the robustness of control systems against uncertainties and external disturbances. By combining the theoretical rigor of mechanistic models with the adaptability of

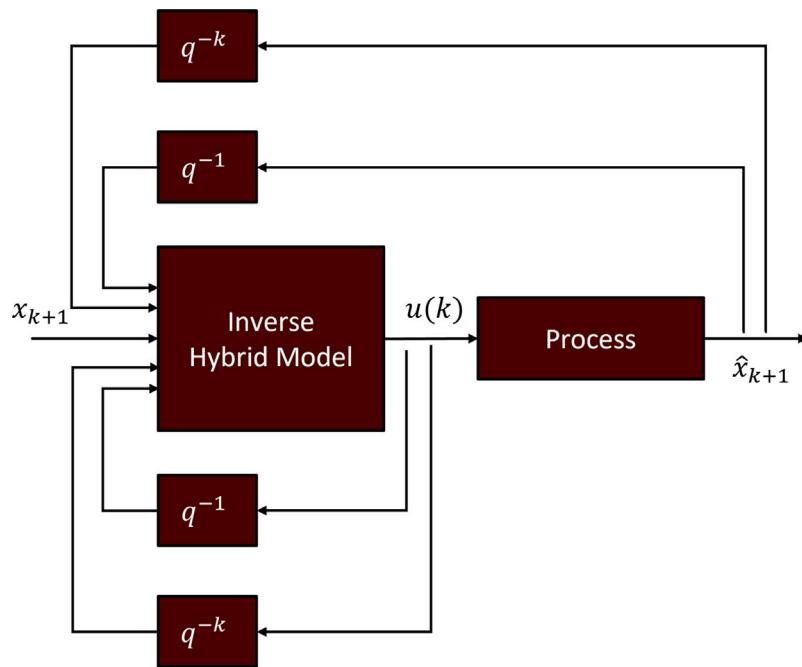


Fig. 7. Schematic for the use of hybrid models for inverse process control.

machine learning models, hybrid approaches offer a balanced solution that maintains control performance under varying conditions. This dual capability makes hybrid models particularly valuable in industrial process control, where maintaining robustness and reliability is crucial for optimal operation.

Hybrid models have extensive applications in developing direct inverse control, addressing the challenge of determining optimal control input sequences in complex systems. In direct inverse control, a control strategy is developed where the control system directly computes the necessary input actions to achieve desired output setpoints by inverting the process model. The direct inverse controller uses the model to predict the required inputs that will drive the system's outputs to match the target setpoints. The inverse calculation, when done using hybrid models, benefits from improved accuracy due to the model's enhanced predictive capabilities. In the study by Ng and Hussain (2004), researchers developed an inverse controller for a semi-batch reactor by leveraging three NNs. These networks were employed to predict uncertain parameters, refine the model's residuals, and correct errors in the calculated input trajectories, thereby enhancing the control strategy's overall accuracy. Similarly, Xiong and Jutan (2002) applied a hybrid model for inverse calculations in controlling reactor temperatures within exothermic semi-batch reactors. This approach effectively managed the nonlinear dynamics of the process, demonstrating the robustness and adaptability of hybrid models in complex control scenarios. It is important to note that in Fig. 7, which showcases the use of hybrid models in the context of direct inverse control, q^{-1} and q^{-k} are used to delay inputs to the NN Inverse Model. q^{-1} is a single time-step delay, and q^{-k} is a more general form that represents a delay of k time steps. The inverse model uses these delayed inputs (past values of the process outputs and control inputs) to compute the control signal $u(k)$.

Hybrid models are also applicable in internal model-based control (IMC), where the FPM augmented with NN is used to take accurate predictive control actions to track set-points of output trajectories. This addresses the challenge of enhancing predictive control accuracy and robustness. Multiple studies have showcased how hybrid models improve IMC by integrating accurate predictive capabilities with first-principles knowledge, thus providing better control performance and adaptability (Vega et al., 1997; Aoyama and Venkatasubramanian, 1995).

In recent strategies for batch-to-batch control, hybrid models are used to obtain initial control strategies and refine models iteratively. This approach addresses the challenge of optimizing control strategies with limited data while improving model accuracy through continuous learning. Researchers have also highlighted how pre-trained hybrid models provide initial control strategies, which are then refined using new experimental data (Ferreira et al., 2014; Doyle III et al., 2003). The iterative process continues until the model predictions align closely with experimental data, demonstrating the hybrid model's ability to adapt and improve over time. This was further illustrated in bioprocess control, showing how hybrid models can effectively refine control strategies through batch-to-batch learning (Teixeira et al., 2006).

Recent studies have increasingly implemented hybrid models to develop model-based optimal control strategies. In this approach, a nonlinear optimization problem is formulated using the hybrid model, which is then solved to determine the optimal control trajectory (Ghosh et al., 2021; McKay et al., 2022; He et al., 2020). Hybrid models are highly effective in capturing complex system behaviors, offering enhanced predictive accuracy and adaptability. However, a key challenge with conventional hybrid models is their computational expense, especially when applied to real-time control scenarios. To address this, researchers have explored the use of physics-informed neural networks (PINNs) for process control. Unlike traditional hybrid models that rely on integrating mechanistic and DDMs, PINNs directly embed physical laws into the NN architecture (Raissi et al., 2019). This allows PINNs to solve differential equations governing the system dynamics while simultaneously learning from data. The result is a model that not only captures system behaviors accurately but does so in a more computationally efficient manner. The computational efficiency of PINNs arises from their ability to approximate complex dynamics without the need for extensive data-driven training, which can be time-consuming in conventional hybrid models due to the presence of FPMs. Additionally, PINNs can estimate model parameters from online data, allowing them to adapt to changing conditions in real-time (Zheng and Wu, 2023; Alhajeri et al., 2022a). This combination of efficiency and adaptability makes PINNs particularly well-suited for optimizing control strategies in complex systems, addressing both the challenges of computational load and the need for real-time responsiveness.

By addressing these challenges, hybrid models offer robust, adaptive, and accurate solutions for process control applications, enhancing

overall system performance and reliability. The integration of FPM and NN in hybrid models enables them to manage complexities, adapt to real-time changes, and optimize control strategies effectively, making them invaluable tools in modern process control.

4.1.4. Model discovery

The discovery of governing equations enhances dynamic systems' forecasting, control, and analysis capabilities, which is critical in fields like neuroscience, cell biology, ecology, and epidemiology, where such equations are often unknown. Despite abundant data, the lack of explicit analytic descriptions challenges traditional modeling approaches. This section explores the integration of hybrid modeling with deep learning to address these challenges, focusing on the discovery and application of PDEs.

Hybrid models leverage ANNs to encode solutions to PDEs while ensuring adherence to fundamental physics laws. This approach is particularly advantageous when data collection is prohibitively expensive and traditional ML methods do not perform well due to insufficient robustness. The methodology relies on automatic differentiation, facilitating efficient derivative computation, which is critical for training models that incorporate physics constraints. Raissi et al. (2017, 2019) pioneered this approach with PINNs, which optimize an objective function to minimize model prediction discrepancies and physics constraint violations. This dual focus ensures that models are not only data-driven but also grounded in physical reality, enhancing their applicability and robustness. Recently, researchers have developed a methodology for using PINNs to handle plant-model mismatch, utilizing both first-principles knowledge and plant data (Moayedi et al., 2024). The PINN framework directly incorporates physical laws as constraints during training, making it particularly suited for complex dynamic systems where purely data-driven models such as DNNs or RNNs struggle with accuracy. In contrast to our hybrid approach, which leverages data-driven models to correct unknown parameters in the FPM, PINNs ensure that the neural network adheres to the physical constraints of the system throughout the learning process. Improvement in the predictive accuracy in plant-model mismatch scenarios indicates the potential benefits of employing such frameworks in systems with significant dynamics or uncertainties.

Remark 3. Like hybrid models, PINNs aim to leverage both physical laws and data-driven approaches to improve model accuracy. The key distinction lies in the way these models incorporate physical knowledge: while hybrid models often combine separate components like FPMs and DDMs to correct or enhance predictions, PINNs embed physical constraints as part of the network's architecture. By keeping FPMs and DDMs separate, hybrid models allow clear interpretation and easier integration with existing systems. This separation is especially useful when physical laws are well-understood, but certain parameters need data-driven correction. In contrast, it is harder to isolate and decouple the effects of physics-based knowledge in PINNs, limiting the transparency in certain applications. Hybrid models are also quicker to adapt in industrial workflows, where existing FPMs can be used without retraining an entire model. Thus, while PINNs provide a promising alternative by enforcing physical constraints directly in the learning process, hybrid models retain greater flexibility, modularity, and interpretability, making them particularly suitable for scenarios where both physical laws and data-driven components are needed but should remain distinct.

Innovative methods for learning PDEs from spatiotemporal data have been demonstrated in the literature. One such work is by Lagergren et al. (2020), who developed an ANN-based method to denoise data and estimate partial derivatives accurately. Their approach, tested on models such as the advection-diffusion and Fisher-KPP equations, outperformed traditional denoising techniques, proving highly effective in identifying underlying PDE models. Additionally, hybrid models are

advantageous in rooting for governing physics equations for a particular process. We see such an approach in Patel et al. (2022), where they introduced a PINN tailored for discovering thermodynamically consistent equations, ensuring hyperbolicity in inverse problems related to shock hydrodynamics.

These advancements underscore the potential of hybrid modeling combined with deep learning in unveiling complex and previously obscured dynamical systems, paving the way for significant contributions across various scientific domains.

4.1.5. Plant scale up

The integration of FPM and DDM is valuable in scaling operations from pilot plants to full-scale industrial units, a transition often fraught with challenges due to the change in scale and operating conditions. Scaling up poses significant challenges due to the emergence of different dominant effects that may not be present at smaller scales. A model developed at a pilot scale may lack the necessary detail and accuracy when applied to a larger scale, as it must extrapolate beyond the operating and process conditions for which it was initially designed (Te Braake et al., 1998). Pilot plants are designed for flexibility and controlled experimentation, often under idealized conditions. In contrast, commercial units operate under a broader range of conditions influenced by economic, material, and logistical factors. This transition is non-trivial due to several factors:

- Operational Variabilities: Larger-scale operations often deal with greater variability in input materials and environmental conditions, significantly affecting process stability and output quality.
- Geometric and Kinetic Nonlinearities: Changes in scale can alter the fundamental behavior of the process. For example, heat transfer and fluid dynamics do not scale linearly, affecting reaction rates and mixing efficiency.
- Material and Energy Balances: The quantities of materials, energy inputs, and outputs must be adjusted and optimized for larger scales, which can introduce new constraints and efficiencies.

Larger-scale operations are often subject to greater variability in input materials and environmental conditions, which can significantly affect process stability and output quality. Hybrid models address these operational variabilities by incorporating flexibility in their structure, allowing them to adapt to new data as it becomes available. The main feature of hybrid models that enables this is the ability to handle temporal parameter variability caused by changes in operation. This adaptability is crucial for processes involving complex chemical reactions or biological systems, where even slight changes in conditions can lead to significantly different outcomes. By dynamically updating model parameters in real-time or near-real-time based on incoming data, hybrid models maintain high accuracy in predictions related to yield, quality, and throughput, which is essential for operational optimization and decision-making in industrial environments (Bellos et al., 2005).

Changes in scale can profoundly alter the fundamental behavior of industrial processes, particularly in terms of geometric and kinetic nonlinearities. As processes scale up, factors such as heat transfer and fluid dynamics do not increase proportionally, leading to significant impacts on reaction rates, mixing efficiency, and overall process performance. For instance, in larger systems, the surface area-to-volume ratio decreases, which can reduce the efficiency of heat dissipation and alter flow patterns, thereby affecting the rates of chemical reactions and the uniformity of mixing. These changes occur in a spatiotemporal domain, and the reaction rates and mixing parameters vary a lot in different regions of a system. Hybrid models are particularly effective in addressing these scale-related challenges by combining FPMs with data-driven techniques. These models identify and correct discrepancies that arise when scaling up from laboratory or pilot-scale systems to commercial-scale operations. By integrating data from the commercial

environment, hybrid models recalibrate critical parameters such as reaction kinetics and flow dynamics, ensuring that predictions remain accurate and reliable despite the complexities introduced by scale. In the fluid catalytic cracking (FCC) process, for example, hybrid models are employed to maintain accuracy across different scales. As the process scales up, the nonlinear effects of catalyst aging, feedstock variability, and changes in flow dynamics become more pronounced. By integrating mechanistic simulations of cracking reactions with NNs, hybrid models can dynamically adjust to these scale-induced changes, ensuring that the process remains efficient and stable even at commercial scales (Bollas et al., 2003; Michalopoulos et al., 2001; Acosta-López and de Lasa, 2023).

As processes scale up, the complexity of material and energy balances increases significantly, primarily due to factors like imperfect mixing and the spatially distributed behavior of large-scale systems. In commercial-scale operations, simple assumptions about uniformity in temperature, concentration, and flow patterns that hold at smaller scales no longer apply. Instead, these systems often exhibit complex behaviors governed by PDEs and computational fluid dynamics (CFD), which account for the spatial distribution of variables and their interactions (Nadal-Rey et al., 2021; Enfors et al., 2001; Baldyga et al., 1997). Hybrid models are particularly well-suited to address these complexities. By integrating fundamental principles, such as mass and energy conservation laws, with empirical data from large-scale operations, hybrid models can more accurately predict and optimize material and energy balances. These models consider the non-uniformity inherent in large systems, enabling more precise adjustments to manipulated inputs like energy requirements, material flow rates, and reaction conditions. The main feature that allows the hybrid model to handle these complexities is its ability to capture the latent system behaviors and understand how changes in scale can affect the structure of the FPM and adapt accordingly. Hybrid models can be used to predict spatiotemporally varying parameters even in complex PDEs with Courant–Friedrichs–Lewy (CFL) boundary conditions. For example, in large-scale reactors or distillation columns, where imperfect mixing can lead to hot spots or concentration gradients, hybrid models can predict these deviations and adjust operating conditions accordingly to ensure uniformity and optimal performance. By accurately modeling these spatial variations, hybrid models help in forecasting key performance indicators (KPIs) such as yield, purity, energy efficiency, and waste production, even in the face of scaling challenges. Moreover, hybrid models enable engineers to account for the additional constraints and complexities that emerge at larger scales, such as the increased difficulty of heat transfer, the need for more precise control over reaction kinetics, and the potential for scaling-induced inefficiencies. This allows for more informed decisions about process adjustments, material inputs, and operational conditions, ultimately leading to reduced costs and enhanced efficiency in commercial-scale operations (Simon et al., 2006; Te Braake et al., 1998). This ability to continuously adjust and optimize processes makes hybrid modeling a vital tool for scale-up efforts across various manufacturing and processing industries, ensuring efficient and cost-effective transitions from pilot to full-scale operations.

4.2. Case studies

As discussed throughout this review paper, hybrid modeling has emerged as a powerful tool for addressing complex systems characterized by the integration of DDMs with FPMs. However, its application across different fields encounters distinct challenges, which can be broadly categorized into four main areas: (1) capturing temporal variations in system dynamics, as demonstrated in crystallization processes, where time-series-transformer (TST) models accurately predict growth and nucleation kinetics; (2) managing uncertainties in kinetic parameters, as seen in fermentation processes, where hybrid models effectively predict microorganism growth rates under varying conditions; (3) addressing the spatiotemporal variability in parameters, illustrated by

reaction–diffusion systems, where hybrid models handle complex diffusion processes that vary over time and space; and (4) mitigating the effects of parameter uncertainties in PDEs, such as those in hydraulic fracturing, where hybrid models enable accurate predictions of fracture propagation by learning temporal profiles of uncertain parameters. Each of these applications highlights the capacity of hybrid models to enhance prediction accuracy and system interpretability by effectively resolving these challenges. Next, we will dive into each of these applications in detail and discuss how hybrid models can be used to address these four key challenges.

4.2.1. Application of hybrid models to crystallization

A critical example of applying the hybrid modeling methodology to a system of ODEs is highlighted in the work by Sitapure and Kwon (2023c). In this work, to tackle the challenges associated with capturing complex parametric variation by various NNs in the hybrid modeling framework, the NN is replaced with an advanced sequence-to-sequence modeling architecture, which is called TST (Wen et al., 2022). The developed framework was utilized to capture the temporal variation in the crystallization kinetic parameters (i.e., growth and nucleation rates). In this study, the authors also performed a rigorous analysis of the parallel and series approach for a hybrid model for applications to the case of crystallization. For demonstration purposes, the following equations were utilized:

$$\frac{\partial n(L, t)}{\partial t} + \frac{\partial G(T, C_s)n(L, t)}{\partial L} = B(T, C_s) \quad (35)$$

where $n(L, t)$ is the number of crystals of size L at time t , $B(T, C_s)$ is the total nucleation rate, $G(T, C_s)$ represents the crystal growth rate, and C_s is the solute concentration at temperature T . The following equations highlight the temporal variation in the concentration and temperature of the system:

$$\frac{dC_s}{dt} = -3\rho_c k_v G \mu_2 \quad (36)$$

$$mC_p \frac{dT}{dt} = -UA(T - T_j) - 3\Delta H \rho_c k_v G \mu_2 \quad (37)$$

where μ_2 is the second moment of crystallization, k_v is the shape factor, ρ_c is the crystal density, C_p is the heat capacity of the crystallization slurry, m is the total mass of slurry, A is the total area between the jacket and reactor, U is the heat transfer coefficient, and ΔH is the heat of crystallization. In the equations above, uncertainties associated with G and B (i.e., the crystal growth and nucleation rates) lead to unavoidable plant-model mismatch. To address this mismatch, the parallel hybrid model is used, which aims to eliminate the error by utilizing a TST model to predict the temporal variation of the error between model predictions and experimental results (Fig. 8). This method is preferred when the FPM alone cannot capture all the dynamic variations in crystallization, offering a more flexible way to improve accuracy in real-time. In the case of the series hybrid model, a TST is utilized to predict the uncertain parameters, i.e., G and B , and then these parameters are utilized in the FPM (Fig. 9). This approach is essential when maintaining the crystallization process's interpretability is essential, as it directly incorporates data-driven insights into the FPM (Sitapure and Kwon, 2023c). Utilizing TSTs in the hybrid modeling framework allows it to handle long-term dependencies in data, which is a challenge in process dynamics. When conventional DDMs like RNNs are utilized, they suffer from the challenges associated with vanishing gradients. In this case, utilizing the multiheaded attention mechanism to capture the temporal variation of the growth and nucleation rates alleviates this issue (Sitapure and Kwon, 2023b,a). The training data in this work was obtained by randomly manipulating operating conditions under a known case of nucleation and growth rates. The performance metrics include normalized MSE (NMSE) and the coefficient of determination (R^2), with results showing values within the range of $[10, 50] \times 10^{-4}$ for NMSE and over 0.99 for R^2 , indicating high accuracy and reliability. It is important to note that two architectures of TST models (i.e., Base and Large) were considered, which vary based on several

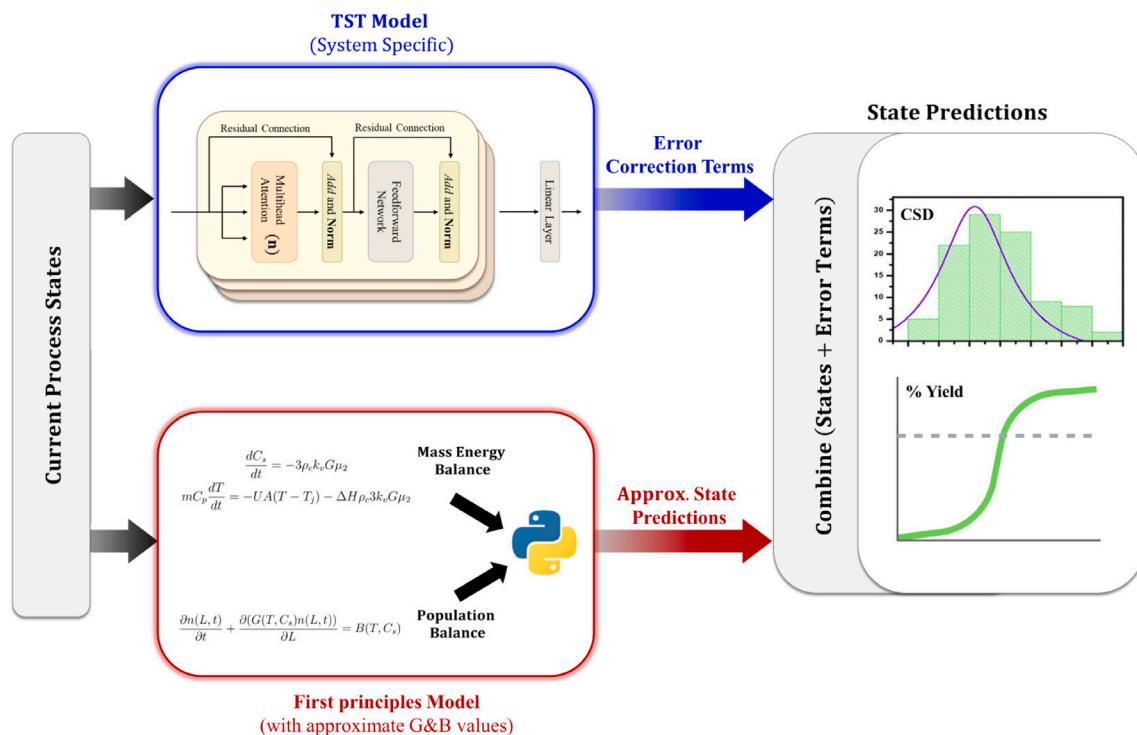


Fig. 8. A schematic illustration of TST-based hybrid model in a parallel configuration.

Source: Reprinted (adapted) with permission from Sitapure and Kwon (2023c).

© 2023 American Chemical Society.

hyperparameters such as the number of encoders/decoders, the internal dimensions (d_{model}), the size of NN, and other aspects. For context, the Base architecture has 2.5M parameters, 4 attention heads, and 2 encoder and decoder blocks, while the Large architecture has 6.5M parameters, 8 attention heads, and 6 encoder and decoder blocks. The training performance of the hybrid model is shown in Fig. 10.

The corresponding evolution of the CSD with time is shown in Fig. 11. It is also clear from Fig. 11 that owing to the numerical instabilities of the parallel hybrid model, the CSD predicted by the parallel hybrid model can have significant fluctuations as opposed to the smooth nature of the predictions from the series hybrid model. This is primarily because the parallel hybrid model tends to capture the discrepancy between the FPM prediction and the actual data by capturing the error values. In contrast, the series hybrid model allows us to predict accurate values of the growth and nucleation kinetics, ultimately leading to smoother CSD predictions.

Model predictions under a varying set of operating conditions (i.e., varying jacket temperature) were analyzed to observe the interpretability preserved by the series hybrid model. It was seen that the variation in the crystal growth and nucleation kinetics predicted by the hybrid model aligns closely with the actual values of the crystal growth kinetics, as is shown in Fig. 12. Overall, the integration of TSTs with traditional physics-based models represents a significant advancement in the field of process modeling, particularly for complex systems like batch crystallization. The hybrid models thus improved prediction accuracy and enhanced the interpretability of the results.

4.2.2. Fermentation process

In the study by Shah et al. (2022), the application of hybrid modeling to a complex fermentation process was shown. In this study, the primary motivation for utilizing a hybrid model was to capture the uncertainty associated with the kinetic parameters of the microorganism growth. The proposed hybrid model was able to address the longstanding challenge associated with accurately capturing the kinetics of microorganism growth in a 100,000-gallon industrial-scale fermentation process.

To have a better understanding of the work done here, we first highlight some details of the kinetic model that was developed in this work:

$$\mu = (\mu_{S_1} + \mu_{S_2} + \mu_I) \cdot \left(\frac{X_2}{K_{X_2} + X_2} \right) \quad (38)$$

$$\mu_{S_1} = \frac{\mu_{max,S_2} \cdot S_1}{K_{S,S_1} + S_1 + a_{S_1,S_2} \cdot S_2 + a_{S_1,I} \cdot I} \quad (39)$$

$$\mu_{S_2} = \frac{\mu_{max,S_2} \cdot S_2}{K_{S,S_2} + S_2 + a_{S_2,S_1} \cdot S_1 + a_{S_2,I} \cdot I} \quad (40)$$

$$\mu_I = \frac{\mu_{max,I} \cdot I}{K_{S,I} + I + a_I \cdot S_1 + a_{I,S_2} \cdot S_2} \quad (41)$$

$$\frac{dX_2}{dt} = k_L a \cdot (X_{2max} - X_2) - q_{X_2} \cdot B \quad (42)$$

where μ , μ_i , and $\mu_{max,i}$ refer to the overall growth rate, the growth rates associated with each component (i.e., Substrate 1 (S_1), Substrate 2 (S_2), and Intermediate (I)), and the maximum specific growth rate of the microorganism associated with each component respectively. $K_{s,i}$ and $a_{i,j}$ refer to the half-velocity constant associated with each component and the inhibitory effect of component i on the utilization of component j by the microorganisms. It is important to note that X_2 is introduced in the equation for the overall growth rate to incorporate the effect of DO. Here, X_{2max} is the maximum value of X_2 , K_{X_2} is the half-velocity constant associated with X_2 , $k_L a$ is the mass transfer coefficient, and q_{X_2} is the uptake rate of X_2 .

Subsequently, the fermentation model was built to have phases, with each phase indicating a different mode of operation and each of these phases having a different nutrition source. Phase 1 involves the following dynamic equations:

$$\frac{dB}{dt} = (\mu_{S_1} + \mu_{S_2} + \mu_I) \cdot B - mp_1 \cdot \frac{F_{in}}{V} \cdot B \quad (43)$$

$$\frac{dS_1}{dt} = -\frac{\mu_{S_1} \cdot B}{Y_B/S_1} - \frac{F_{in}}{V} \cdot S_1 \quad (44)$$

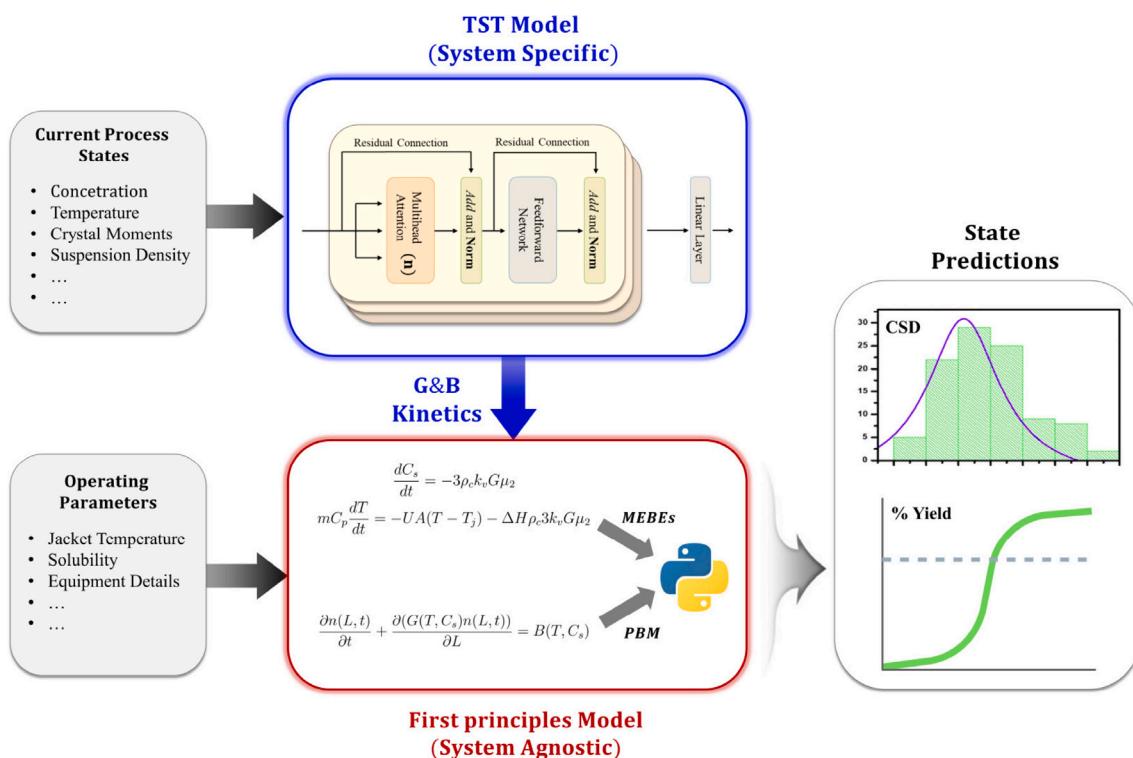


Fig. 9. A schematic illustration of TST-based hybrid model in a series configuration.

Source: Reprinted (adapted) with permission from Sitapure and Kwon (2023c).

© 2023 American Chemical Society.

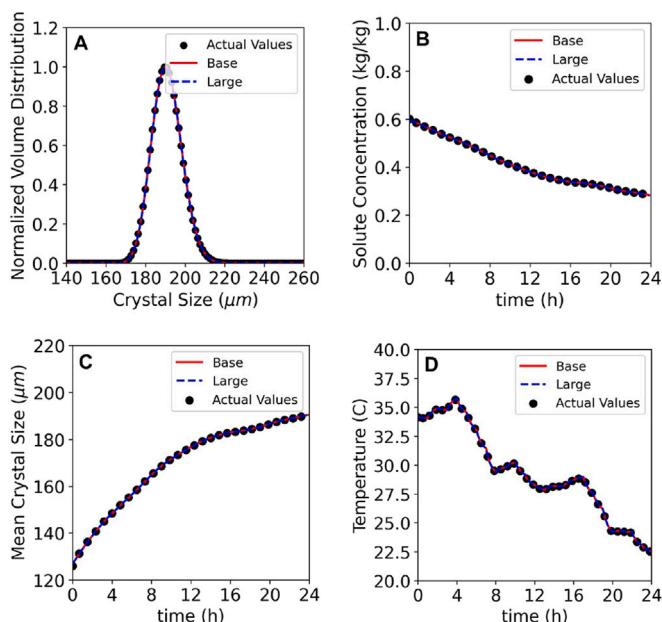


Fig. 10. Comparison of predicting (a) final CSD, (b) solute concentration, (c) mean crystal size, and (d) crystallization temperature by the series hybrid model and true system dynamics for an arbitrarily chosen operating condition. The R^2 value for all the predictions is over 0.99.

Source: Reprinted (adapted) with permission from Sitapure and Kwon (2023c). © 2023 American Chemical Society.

where B refers to biomass, and Y_{B/S_1} refers to the yield coefficient of Biomass associated with Substrate 1. During phase 2, the reactor model was described by the following equations:

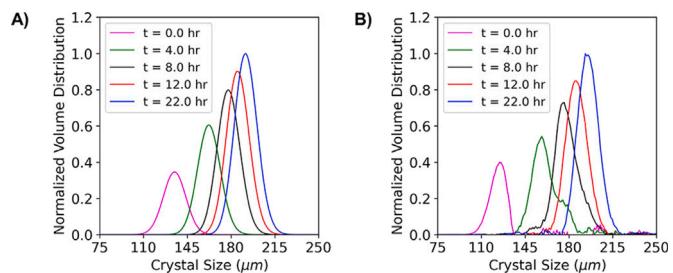


Fig. 11. CSD evolution as predicted by (a) series hybrid model and (b) parallel hybrid model.

Source: Reprinted (adapted) with permission from Sitapure and Kwon (2023c). © 2023 American Chemical Society.

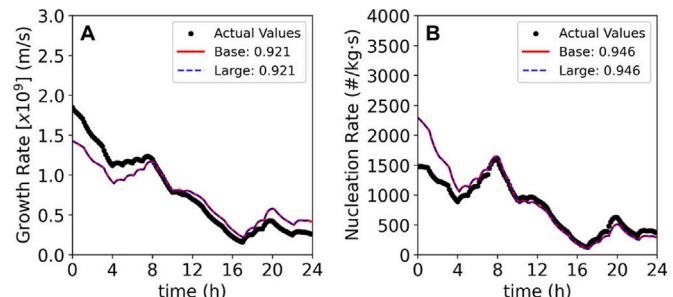


Fig. 12. Comparison of predicted G&B kinetics and true kinetics for the series hybrid model.

Source: Reprinted (adapted) with permission from Sitapure and Kwon (2023c). © 2023 American Chemical Society.

$$\frac{dB}{dt} = (\mu_{S_1} + \mu_{S_2} + \mu_I) \cdot B \quad (45)$$

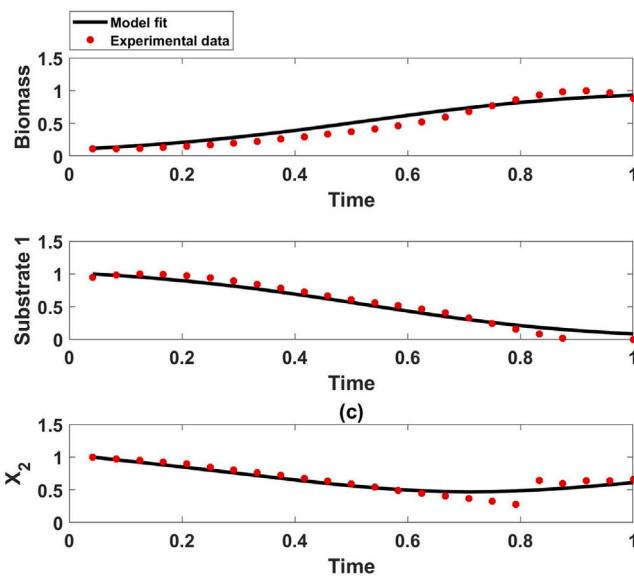


Fig. 13. Hybrid model training: comparison of model prediction with experimental data during phase 1 of the fermentation process (Copyright 2022 Elsevier).

Source: Reprinted (adapted) with permission from Shah et al. (2022).

$$\frac{dS_1}{dt} = -\frac{\mu_{S_1} \cdot B}{Y_{B/S_1}} \quad (46)$$

$$\frac{dS_2}{dt} = k_1 \cdot \mu_{S_1} \cdot B - \frac{\mu_{S_2} \cdot B}{Y_{B/S_2}} - \frac{F_{in}}{V} (S_2 - S_{2,initial}) - p_1 \cdot X_1 \quad (47)$$

$$\frac{dI}{dt} = (k_2 \cdot \mu_{S_1} + k_3 \cdot \mu_{S_2}) \cdot B - \frac{\mu_I \cdot B}{Y_{B/I}} - \frac{F_{in}}{V} \cdot I \quad (48)$$

$$\frac{dP}{dt} = (\alpha_1 \cdot \mu_{S_1} + \alpha_2 \cdot \mu_{S_2} + \alpha_3 \cdot \mu_I) \cdot B + \beta \cdot B - \frac{F_{in}}{V} \cdot P + p_2 \cdot X_1 \quad (49)$$

$$\frac{dV}{dt} = F_{in} \quad (50)$$

where P is the product concentration, V is the reactor volume, α_i is the coefficient linked to the growth rate responsible for the increase in Substrate 2 and Intermediate, β is the coefficient linked to the non-growth associated term responsible for the increase in product, $S_{2,initial}$ is the initial feed concentration of Substrate 2, and F_{in} refers to the feed flow rate of Substrate 2. The parameters Y_{B/S_1} , Y_{B/S_2} , and $Y_{B/I}$ refer to the yield coefficient of biomass associated with each component. p_1 and p_2 are empirical coefficients that allow X_1 to be incorporated as consumption and production terms in the equation for Substrate 2 and the product, respectively.

A GSA is required to determine the most critical parameters. These parameters can be predicted by the NN component of the hybrid model. GSA highlights that $S_{2,initial}$, Y_{B/S_2} , α_2 , and μ_{max,S_2} are most critical. These parameters were then estimated using the DNN component of the hybrid model. The results for the hybrid model training for Phase 1 and Phase 2 are highlighted in Figs. 13 and 14, respectively. Subsequently, the evolution of the model parameters is shown in Fig. 15. The results of the hybrid model highlight its excellent capability in capturing the temporal uncertainty with crucial model parameters that significantly improve the prediction accuracy of the FPM.

4.2.3. Application to hydraulic fracturing

Hybrid models are also applied to the case of hydraulic fracturing by Bangi and Kwon (2020). This application showcases the unique ability of the hybrid model to capture temporally varying parameters in the system of PDEs. Hydraulic fracturing is a process in which slick water with proppant is pumped inside a reservoir with very low permeability at very high pressure to create high permeability fractures through which hydrocarbons can flow from the reservoir into the

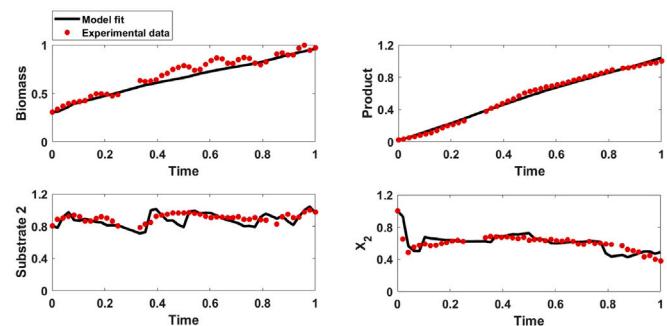


Fig. 14. Hybrid model training: comparison of model prediction and experimental data during phase 2 of the fermentation process (Copyright 2022 Elsevier).

Source: Reprinted (adapted) with permission from Shah et al. (2022).

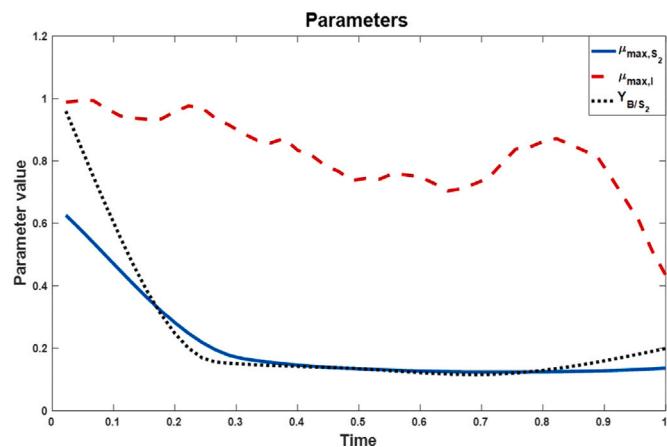


Fig. 15. Temporal variation of the parameters predicted by the hybrid model (Copyright 2022 Elsevier).

Source: Reprinted (adapted) with permission from Shah et al. (2022).

wellbore, thereby enhancing the amount of recoverable hydrocarbons from the reservoir. Specifically, fracture propagation in this study was modeled using the Perkins–Kern–Nordgren (PKN) model described by the following equation:

$$\frac{\pi H}{4} \frac{\partial W}{\partial t} - \frac{\pi E}{128\mu(1-\nu^2)} \left(3W^2 \left(\frac{\partial W}{\partial z} \right)^2 + W^3 \frac{\partial^2 W}{\partial z^2} \right) + HU = 0 \quad (51)$$

where H is the fracture height, W is the fracture width, ν is the Poisson's ratio, U is the leak off rate per unit height, t is the time elapsed, and E is the Young's modulus of the rock formation. The boundary condition and the initial conditions needed to solve the fracture propagation equation is mentioned as follows:

$$q_z(0, t) = Q_0 \quad W(L(t), t) = 0 \quad (52)$$

$$W(z, 0) = 0 \quad (53)$$

where Q_0 is the fracturing fluid injection rate at the wellbore, and $L(t)$ is the fracture length varying with time. In solving the equation mentioned above, uncertainty associated with the leak-off rate, U , leads to significant limitations in the predictive capability of the first-principles PKN model (Siddhamshetty et al., 2018). Understanding the physics associated with leak-off rate, it was proposed by Carter to be varying with time with the following empirical form:

$$U = \frac{2C_{leak}}{\sqrt{t - \tau(z)}} \quad (54)$$

where U is the leak-off rate, t is the associated time, and C_{leak} is the leak-off rate. The objective of the hybrid model was to learn the temporal variation of the uncertain parameter from the given data of

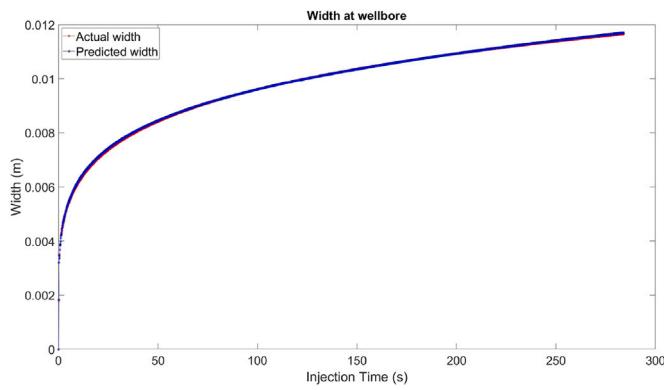


Fig. 16. Hybrid model training: Comparison of the actual and predicted fracture width (Copyright 2020 Elsevier).

Source: Reprinted (adapted) with permission from Bangi and Kwon (2020).

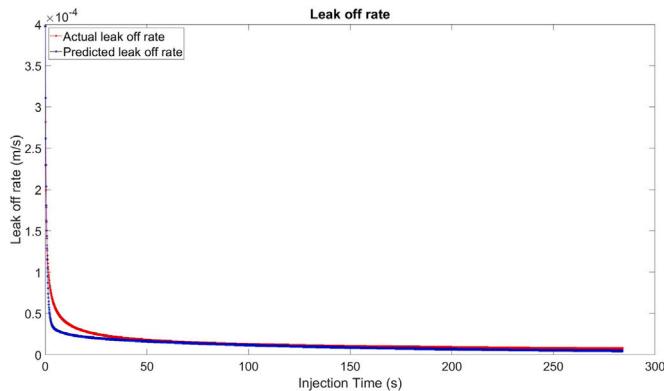


Fig. 17. Comparison of leak-off rates predicted from the DNN and actual values (Copyright 2020 Elsevier).

Source: Reprinted (adapted) with permission from Bangi and Kwon (2020).

fracture propagation that could be observed from field operations. To observe the effectiveness of the hybrid model, the PKN model with the leak-off rate given by Eq. (54) was considered as the virtual experiment, and it was also utilized for generating training data, with a value of $Q_0 = 0.03 \text{ m}^3/\text{s}$.

The fracture propagation results obtained from training the hybrid model are presented in Fig. 16, with an RMSE of 0.0019 on the training dataset. As depicted in Fig. 17, the hybrid model accurately captures the leak-off rate profile, closely aligning with the theoretical values given by Carter's equation (Eq. (54)). To evaluate the model's performance under new injection flow rates of $Q_0 = 0.02 \text{ m}^3/\text{s}$, $Q_0 = 0.04 \text{ m}^3/\text{s}$, and $Q_0 = 0.05 \text{ m}^3/\text{s}$, the test RMSEs were found to be 0.1097, 0.0711, and 0.1225, respectively. To assess the model's extrapolation capability, the fracture propagation predictions were compared with values from virtual experiments and a purely DDM trained on similar data, as shown in Fig. 18. Despite being tested on previously unknown operating conditions, the hybrid model demonstrated superior performance, effectively handling discrepancies due to its integration of first-principles. This study underscores the hybrid model's robust extrapolation capability and effectiveness in directly learning temporal profiles of key parameters from observed data, providing a strong foundation for predictive modeling in complex systems.

4.2.4. Application to reaction-diffusion system

In the last example, the application of the hybrid modeling methodology to capture temporal variation in uncertain parameters was analyzed. In the recent study by Pahari et al. (2024b,a), the application

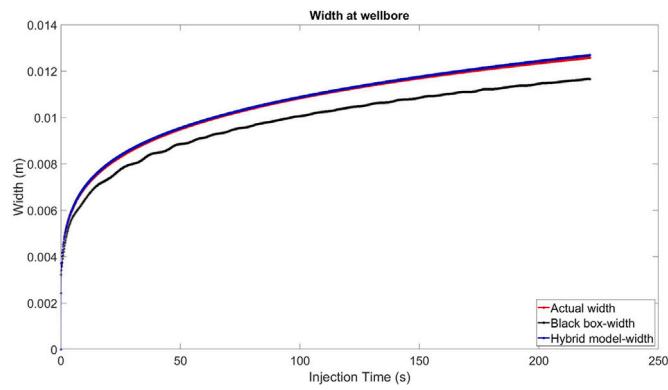


Fig. 18. Comparison of wellbore widths obtained from the hybrid model, DDM, and actual values (Copyright 2020 Elsevier).

Source: Reprinted (adapted) with permission from Bangi and Kwon (2020).

of a hybrid modeling framework to the PDE governing the reaction-diffusion of cells in a growth medium was analyzed. The primary challenge addressed in this study involved capturing uncertainty in model parameters that vary spatiotemporally. The equation covering the reaction-diffusion of cells is called the Fisher-KPP model and is given by the following equation:

$$\frac{\partial C}{\partial t} = \nabla(D\nabla C) + \lambda C \left(1 - \frac{C}{K}\right) \quad (55)$$

where $C(x, y, t)$ is the cell density that varies with space (x, y) and with time t , D is the diffusivity of the system, λ is the cell proliferation rate, and K is the cell capacity density. The boundary conditions needed to solve Eq. (55) are presented as follows:

$$\frac{\partial C}{\partial x} = 0 \quad x = 0 \quad (56)$$

$$\frac{\partial C}{\partial x} = 0 \quad x = L \quad (57)$$

$$\frac{\partial C}{\partial y} = 0 \quad y = 0 \quad (58)$$

$$\frac{\partial C}{\partial y} = 0 \quad y = L \quad (59)$$

where L is the length of the simulation domain. In the case of modeling the cell colony invasion and tumor growth, this work implemented a branched architecture of the hybrid model. This kind of Fisher-KPP model with spatiotemporally varying diffusivity is often called Porous-Fisher model. Due to spatiotemporally varying parameters like diffusivity, the nature of the solution for the PDEs changes. Furthermore, parameters like K and λ have only temporal variation and no spatial variation. To address the challenges associated with simultaneously capturing both spatiotemporally varying parameters and temporally varying parameters, this work implemented a branched architecture of the hybrid model. The branched architecture has CNNs and CNNs/DNNs to extract spatial features and subsequently relate them to spatiotemporally and temporally varying parameters. A detailed architecture for the branched hybrid model architecture is highlighted in Fig. 19.

Since the machine-learning component of the hybrid model predicts spatially varying parameters in every time-step, numerical stability in the solution to the reaction-diffusion system arises. To avoid this, the CFL stability criterion had to be satisfied while training the hybrid model. A comparison of the prediction of the hybrid model and FPM prediction is highlighted in Fig. 20. The remarkable difference between the predictions of the FPM and the hybrid model is the presence of a clear wavefront for cell density growth. This is primarily because the FPM (i.e., the Fisher-KPP model) considers a constant diffusivity while the hybrid model captures spatiotemporal variation in the diffusivity.

The comparison of the actual cell density and the cell density predicted by the hybrid model aligns very closely. Additionally, the

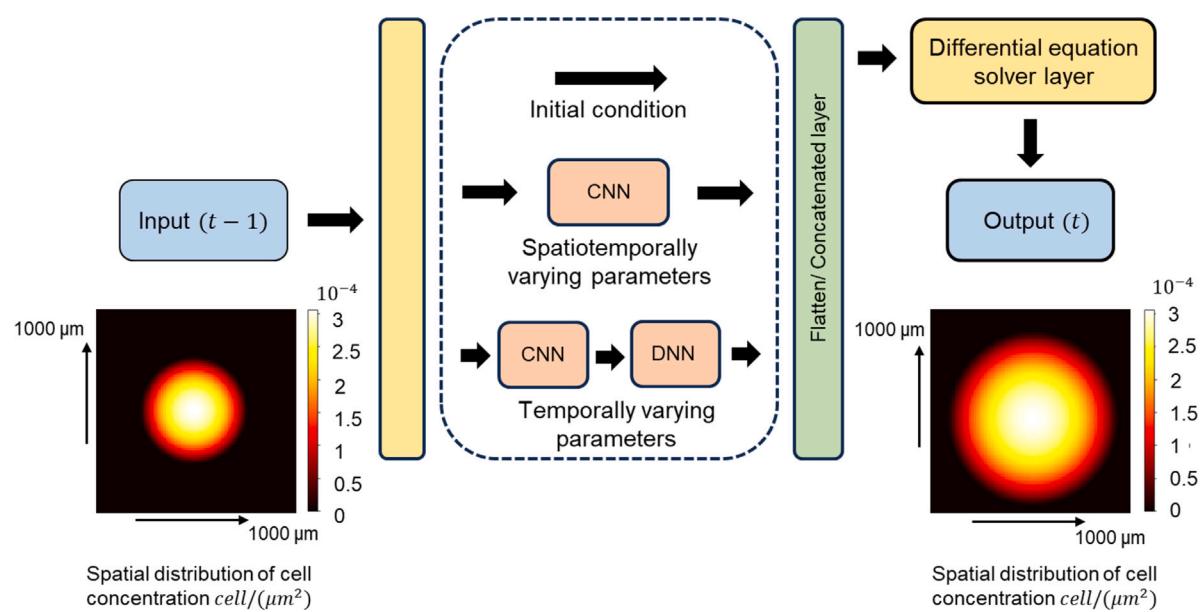


Fig. 19. Hybrid modeling architecture for predicting the temporally varying parameters and spatiotemporally varying parameters.

Source: Reprinted (adapted) with permission from Pahari et al. (2024b).

© 2024 American Chemical Society.

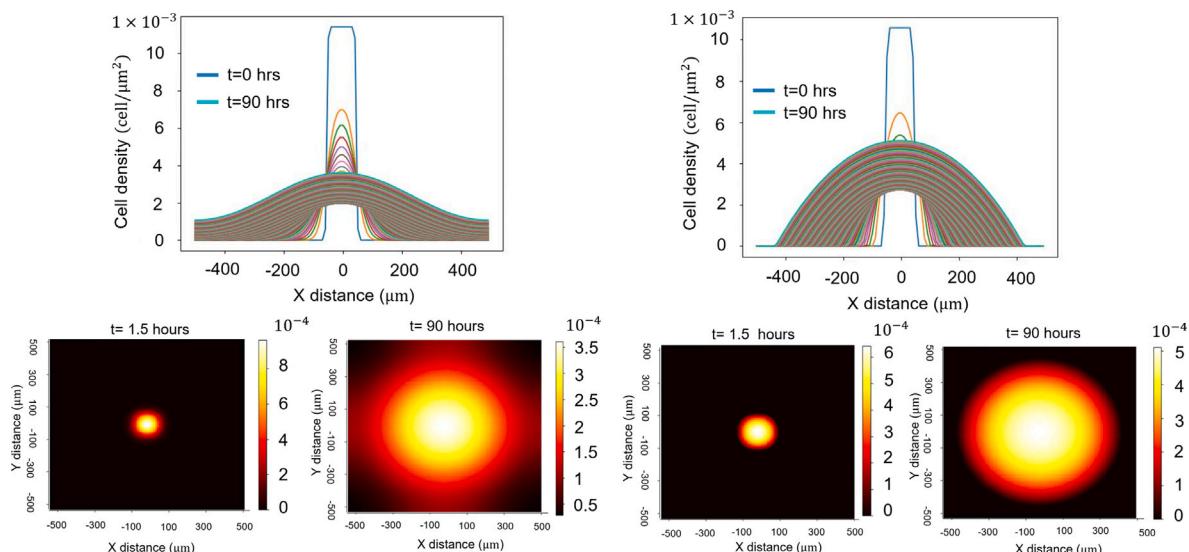


Fig. 20. Spatiotemporal variation of the cell density obtained from (a) the FPM, and (b) hybrid model predictions.

Source: Reprinted (adapted) with permission from Pahari et al. (2024b).

© 2024 American Chemical Society.

diffusivity estimated by the hybrid model aligns closely with the actual diffusivity, as demonstrated in Fig. 21. The values of the parameters like K and λ predicted by the hybrid model align very closely with the actual values, as shown in Fig. 22. Although, in practice, these values are expected to be constant, the hybrid model keeps them to be time-varying. However, the temporal variation of these values is seen to be very low. This example highlights the capability of the hybrid model to capture temporal and spatiotemporal variation in parameters.

5. Outlook

The future of hybrid modeling is poised to address several critical challenges and opportunities, particularly in the context of rapidly evolving computational capabilities and the increasing complexity of

systems being modeled. As hybrid models continue to integrate diverse sources of knowledge and data, several key research directions emerge that could significantly impact their development and application.

5.1. Adaptive hybrid models

One of the most promising avenues for future research in hybrid modeling is the development of advanced methodologies that enable the dynamic adaptation and recalibration of sub-models within a hybrid framework in response to changing process conditions. In complex and evolving industrial environments, maintaining model accuracy and robustness is paramount, particularly for applications requiring real-time control and optimization. Traditional approaches often rely on

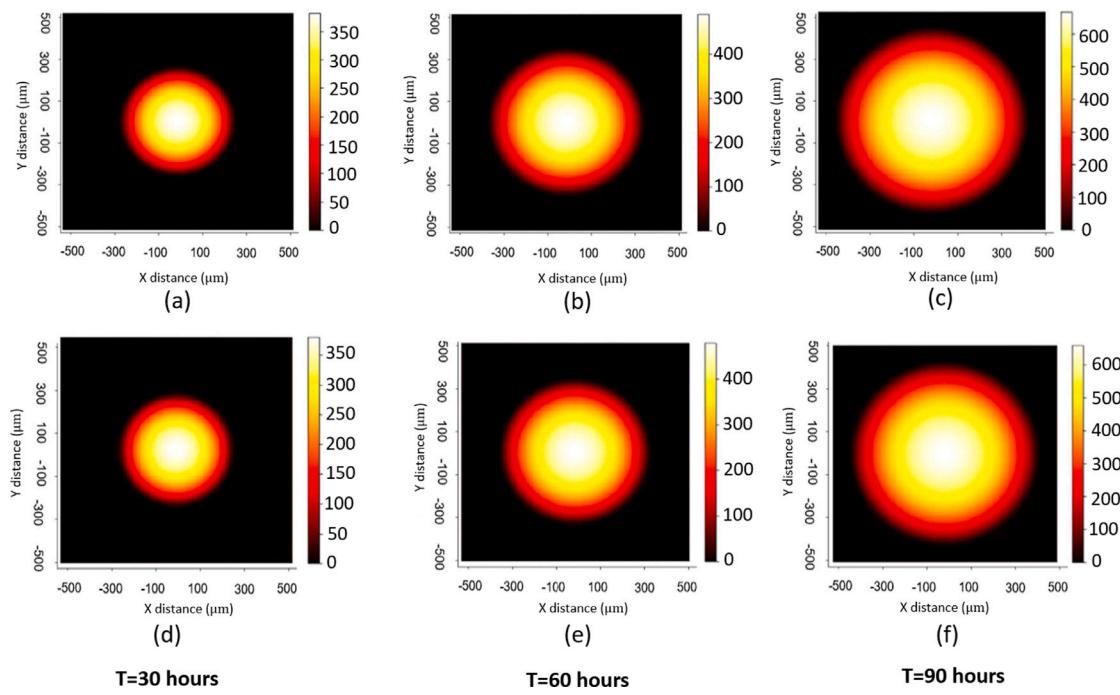


Fig. 21. Comparison of the spatiotemporal evolution of the cell diffusivity ($\mu\text{m}^2/\text{h}$) at different times: 30, 60, and 90 h.
Source: Reprinted (adapted) with permission from Pahari et al. (2024b).
© 2024 American Chemical Society.

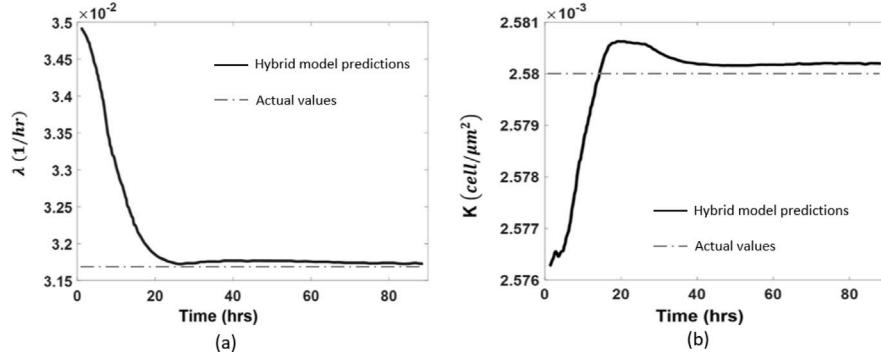


Fig. 22. Temporal evolution of parameters K and λ obtained while training: (a) temporal evolution of λ and (b) temporal evolution of K .
Source: Reprinted (adapted) with permission from Pahari et al. (2024b).
© 2024 American Chemical Society.

static models that struggle to accommodate shifts in operating conditions, leading to potential inaccuracies and suboptimal performance. To address this, future research could focus on creating automated DDM identification and updating strategies that continuously adjust sub-model parameters or dynamically switch between different modeling paradigms as conditions evolve. Such methodologies would likely involve the integration of online learning algorithms, real-time data assimilation, and adaptive filtering techniques to ensure that hybrid models remain accurate and reliable across varying process states. For example, in flexible pharmaceutical manufacturing, where process adaptability is crucial, the ability to implement self-adjusting models that react to shifts in process parameters could significantly enhance production efficiency and product quality (Gernaey and Gani, 2010; Glassey et al., 2011). By advancing these adaptive hybrid modeling techniques, researchers can improve the reliability and applicability of hybrid models in real-time industrial applications, thereby pushing the boundaries of what can be achieved in process optimization and control.

5.2. Multiscale hybrid modeling

Hybrid models offer a powerful and flexible framework for addressing the unique challenges associated with processes that operate on multiple timescales, a common occurrence in many industrial and biochemical systems. These systems often exhibit both fast and slow dynamics, making them difficult to model accurately and efficiently using traditional methods. By decoupling these dynamics and modeling them separately, hybrid models can significantly enhance both the efficiency and accuracy of simulations. In systems where fast and slow timescales coexist, stiff equations often arise. These equations are challenging to solve because they require very small time steps to accurately capture the fast dynamics, leading to high computational costs. However, by leveraging the strengths of hybrid modeling, it is possible to separate the fast and slow dynamics into different modeling frameworks. The FPM can be used to describe the slower, steady-state dynamics of the system, such as thermal diffusion or long-term chemical equilibria. These models are based on well-established physical laws and can provide a solid foundation for understanding the underlying slow processes. On the other hand, DDMs are more suitable for

capturing fast, transient dynamics that mechanistic equations may not fully understand or easily represent. These might include rapid reaction kinetics, sudden fluctuations in process variables, or fast responses to control inputs. By using machine learning algorithms, particularly those designed for time-series data, such as RNNs or transformers, DDMs can effectively learn and predict these fast dynamics from historical and real-time data. By isolating the fast dynamics in a DDM, the need for excessively small time steps in the FPM is reduced, thereby lowering computational demands. This separation also helps to mitigate the stiffness of the overall system, leading to more stable numerical simulations. The hybrid model can then couple the outputs of the DDM (representing fast processes) with the FPM (representing slow processes), ensuring that the interactions between fast and slow dynamics are accurately captured without compromising computational efficiency. Moreover, hybrid models can provide real-time insights into both aspects of biopharmaceutical manufacturing, where cell culture processes involve fast metabolic reactions and slower growth phases. This enables more precise control over feeding strategies, enhancing product yield and quality.

5.3. Integration with state-of-the-art ML models

Hybrid models encounter significant challenges when applied to computationally intensive systems such as CFD, molecular dynamics (MD), dissipative particle dynamics (DPD), and kinetic monte carlo (kMC) simulations (Golshan et al., 2020; Pahari et al., 2021a). This complexity primarily stems from the difficulties in training these models directly from experimental data, which is often sparse or noisy. However, recent advances in molecular simulations have demonstrated the substantial benefits of NN-based force fields, which have shown superior accuracy in predicting interaction parameters between molecules compared to traditional parametric force fields. The integration of advanced deep learning architectures, such as transformers and models with attention and memory-based architectures (MAMBA), into these hybrid models, represents a significant leap forward in process modeling (Vaswani, 2017; Gu and Dao, 2023). These sophisticated models excel at capturing long-range dependencies and complex interactions within large datasets, making them particularly advantageous for chemical processes where traditional models may fall short. Traditional hybrid models are prone to overfitting, particularly when dealing with high-dimensional data, limiting their ability to generalize to new, unseen scenarios. They find it challenging to integrate data from multiple sources with varying fidelity levels, which complicates the training process and reduces model accuracy. Ensuring that these models converge to a stable and accurate solution during training is difficult, especially as the model complexity increases. By leveraging these advanced DDMs, hybrid models can achieve more accurate and detailed predictions of complex properties, such as reaction kinetics and material behaviors, under varying conditions (Ross et al., 2022; Chithrananda et al., 2020). This not only improves the precision of the models but also enhances their generalizability and robustness across different operating conditions. To fully realize these benefits, future research should focus on developing more generalizable and computationally efficient models by leveraging state-of-the-art ML techniques. These efforts will position hybrid models as essential tools for the future of process modeling, enabling more accurate, adaptable, and robust simulations in computationally intensive applications.

5.4. Automatic determination of the hybrid model structure

One of the critical challenges in hybrid modeling is the automatic determination of the most appropriate model structure, especially in complex systems with intricate dynamics (Schweidtmann et al., 2023). Traditionally, this process has relied heavily on expert knowledge and trial-and-error, making it time-consuming and resource-intensive. However, recent advancements in optimization algorithms and machine

learning are paving the way for more efficient and automated approaches. Techniques like mixed-integer programming (MIP) allow for the systematic exploration of potential model configurations by framing the selection of components as an optimization problem. This method efficiently identifies the optimal combination of mechanistic and data-driven elements that best represent the system. These advancements significantly reduce the time and expertise required to develop robust hybrid models, facilitating their broader adoption across various industries. By automating the structure determination process, hybrid models become more accessible and easier to implement, particularly in fields like pharmaceuticals, chemical engineering, and energy systems, where complexity and data richness are prevalent.

5.5. Integration with uncertainty quantification techniques

Uncertainty quantification is crucial for the reliable application of hybrid models, especially in complex and high-stakes industries like pharmaceuticals, chemical engineering, and energy systems. As hybrid models combine mechanistic and data-driven components, they must address both epistemic uncertainty (stemming from incomplete knowledge or model structure) and aleatoric uncertainty (arising from inherent variability in the system). Techniques such as Bayesian inference and Monte Carlo methods offer robust frameworks for managing these uncertainties. Bayesian inference allows for the probabilistic updating of model parameters as new data becomes available, integrating prior knowledge with real-time information. Monte Carlo methods, on the other hand, provide a comprehensive understanding of potential outcomes by simulating a range of scenarios based on probability distributions of input parameters, capturing the inherent variability in complex systems. The integration of these uncertainty quantification techniques into hybrid models is essential for enhancing their accuracy and reliability, particularly in real-time applications like digital twins and smart manufacturing systems. Future research will likely focus on improving the scalability and precision of these methods, leveraging advancements in machine learning, such as Bayesian NNs and variational inference, to handle the high-dimensional and nonlinear nature of hybrid models. By refining these approaches, hybrid models can provide more actionable insights and maintain robust performance even in dynamic environments, ultimately driving their broader adoption across various industries where precise risk management and decision-making are critical.

5.6. Development of software tools

The development of software tools that enable the flexible integration of multiple knowledge sources into hybrid models is another critical area for future research. As hybrid models grow in complexity, the ability to seamlessly combine different types of data — ranging from empirical measurements to theoretical models — will be essential for their practical application. These tools should be designed to support the integration of diverse modeling approaches, such as combining mechanistic models with advanced data-driven techniques like transformers or deep reinforcement learning. Such tools would not only facilitate the development of more sophisticated hybrid models but also make these models more accessible to practitioners in various industries.

5.7. Real-world implementation

Hybrid modeling has shown great promise in academic and lab settings, but real-world implementations bring unique challenges that require specific techniques to ensure models work efficiently in dynamic, data-intensive environments. One major hurdle is dealing with incomplete, noisy, or sparse datasets, which can significantly affect model performance. Techniques like data augmentation, filtering, and

regularization are crucial for making data usable and preventing overfitting, especially when data is sparse. Scaling hybrid models for real-time applications or large datasets also presents computational challenges. Running complex simulations in real-time can demand significant processing power, but leveraging distributed computing, GPU/TPU acceleration, and model pruning can optimize performance without sacrificing accuracy. As data dimensionality increases, managing memory and resources becomes tricky in frameworks like TensorFlow or PyTorch. Techniques such as batching, model compression, and simplifying neural network architectures help reduce strain on memory and improve scalability without compromising the model's predictive power.

Optimization strategies are critical for hybrid models in real-life applications. Hyperparameter tuning, through methods like grid search or Bayesian optimization, helps balance the FPM and DDM components, avoiding overfitting while ensuring the FPM accurately captures the physical processes. For dynamic processes, solver performance is another challenge; using custom solvers or integrating more efficient libraries like SciPy in TensorFlow can overcome issues like solver stiffness and poor convergence. Additionally, transfer learning can be a powerful tool, allowing pre-trained models from similar systems to be fine-tuned for new applications, reducing training time, and improving performance when data is limited. Together, these strategies can help ensure that hybrid models remain efficient and accurate across diverse real-world implementations.

6. Conclusion

The future of hybrid modeling is exceptionally promising, with abundant opportunities to advance the state of the art. Researchers can fully unlock the potential of hybrid models by concentrating on the interpretation and adaptation of sub-models, refining methodologies for computationally intensive systems, and developing flexible tools for integrating diverse knowledge sources. These developments will be essential for creating more accurate, adaptable, and computationally efficient models across various applications, from pharmaceutical manufacturing to petrochemical industries to pulp and paper processes. Moreover, advancements in machine learning, particularly in real-time learning and online updating algorithms, will significantly enhance hybrid models' ability to adapt to changing process conditions. The continued integration of hybrid models into digital twins and smart manufacturing systems is poised to drive the next generation of industrial automation, where real-time optimization and control are crucial. The rapid growth in computing power over the coming years will play a pivotal role in this evolution, enabling more sophisticated and responsive hybrid models that meet the demands of modern industry.

CRediT authorship contribution statement

Parth Shah: Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Silabrata Pahari:** Writing – original draft, Software, Project administration, Methodology, Investigation, Data curation, Conceptualization. **Raj Bhavsar:** Writing – original draft, Investigation, Data curation, Conceptualization. **Joseph Sang-II Kwon:** Writing – review & editing, Writing – original draft, Visualization, Supervision, Resources, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, 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.

Acknowledgments

The authors gratefully acknowledge financial support from the Texas A&M Energy Institute, United States and the Artie McFerrin Department of Chemical Engineering.

Data availability

Data will be made available on request.

References

- Acosta-López, J.G., de Lasa, H., 2023. Artificial intelligence for hybrid modeling in fluid catalytic cracking (FCC). *Processes* 12 (1), 61.
- Adib, H., Sharifi, F., Mehranbod, N., Kazerooni, N.M., Koolivand, M., 2013. Support Vector Machine based modeling of an industrial natural gas sweetening plant. *J. Nat. Gas Sci. Eng.* 14, 121–131.
- Aehle, M., Simutis, R., Lübbert, A., 2010. Comparison of viable cell concentration estimation methods for a mammalian cell cultivation process. *Cytotechnology* 62, 413–422.
- Aguiar, H.C., Maciel Filho, R., 2001. Neural network and hybrid model: a discussion about different modeling techniques to predict pulping degree with industrial data. *Chem. Eng. Sci.* 56 (2), 565–570.
- Alauddin, M., Khan, F., Imtiaz, S., Ahmed, S., Amyotte, P., 2023. Integrating process dynamics in data-driven models of chemical processing systems. *Process Saf. Environ. Prot.* 174, 158–168.
- Alhajeri, M.S., Abdullah, F., Wu, Z., Christofides, P.D., 2022a. Physics-informed machine learning modeling for predictive control using noisy data. *Chem. Eng. Res. Des.* 186, 34–49.
- Alhajeri, M.S., Luo, J., Wu, Z., Albalawi, F., Christofides, P.D., 2022b. Process structure-based recurrent neural network modeling for predictive control: A comparative study. *Chem. Eng. Res. Des.* 179, 77–89.
- Ali, M.M., Adewumi, A.O., Blamah, N., Falowo, O., et al., 2015. Mathematical modeling and optimization of industrial problems. *J. Appl. Math.*
- Alotaibi, S., Amooie, M.A., Ahmadi, M.H., Nabipour, N., Chau, K.-w., 2020. Modeling thermal conductivity of ethylene glycol-based nanofluids using multivariate adaptive regression splines and group method of data handling artificial neural network. *Eng. Appl. Comput. Fluid Mech.* 14 (1), 379–390.
- Anderson, J., McAvoy, T., Hao, O., 2000. Use of hybrid models in wastewater systems. *Ind. Eng. Chem. Res.* 39 (6), 1694–1704.
- Aoyama, A., Venkatasubramanian, V., 1995. Internal model control framework using neural networks for the modeling and control of a bioreactor. *Eng. Appl. Artif. Intell.* 8 (6), 689–701.
- Asgari, S., MirhoseiniNejad, S., Moazamigoodarzi, H., Gupta, R., Zheng, R., Puri, I.K., 2021. A gray-box model for real-time transient temperature predictions in data centers. *Appl. Therm. Eng.* 185, 116319.
- Asprion, N., Böttcher, R., Pack, R., Stavrou, M.-E., Höller, J., Schwientek, J., Bortz, M., 2019. Gray-box modeling for the optimization of chemical processes. *Chem. Ing. Tech.* 91 (3), 305–313.
- Åström, K.J., Elmqvist, H., Mattsson, S.E., et al., 1998. Evolution of continuous-time modeling and simulation. In: ESM. Citeseer, pp. 9–18.
- Azarpour, A., Borhani, T.N., Alwi, S.R.W., Manan, Z.A., Mutualib, M.I.A., 2017. A generic hybrid model development for process analysis of industrial fixed-bed catalytic reactors. *Chem. Eng. Res. Des.* 117, 149–167.
- Aziz, I., Khan, I., 2018. Numerical solution of diffusion and reaction-diffusion partial integro-differential equations. *Int. J. Comput. Methods* 15 (06), 1850047.
- Azwar, Hussain, M., Ramachandran, K., 2006. The study of neural network-based controller for controlling dissolved oxygen concentration in a sequencing batch reactor. *Bioprocess Biosyst. Eng.* 28, 251–265.
- Bai, Y., Yan, B., Zhou, C., Su, T., Jin, X., 2023. State of art on state estimation: Kalman filter driven by machine learning. *Annu. Rev. Control* 56, 100909.
- Baldyga, J., Bourne, J., Hearn, S., 1997. Interaction between chemical reactions and mixing on various scales. *Chem. Eng. Sci.* 52 (4), 457–466.
- Bangi, M.S.F., Kao, K., Kwon, J.S.-I., 2022. Physics-informed neural networks for hybrid modeling of lab-scale batch fermentation for β -carotene production using *Saccharomyces cerevisiae*. *Chem. Eng. Res. Des.* 179, 415–423.
- Bangi, M.S.F., Kwon, J.S.-I., 2020. Deep hybrid modeling of chemical process: Application to hydraulic fracturing. *Comput. Chem. Eng.* 134, 106696.
- Bangi, M.S.F., Kwon, J.S.-I., 2023. Deep hybrid model-based predictive control with guarantees on domain of applicability. *AIChE J.* 69 (5), e18012.
- Bansal, S., Roy, S., Larachi, F., 2012. Support vector regression models for trickle bed reactors. *Chem. Eng. J.* 207, 822–831.
- Bao, A., Gildin, E., Narasingam, A., Kwon, J.S., 2019. Data-driven model reduction for coupled flow and geomechanics based on DMD methods. *Fluids* 4 (3), 138.
- Basore, P.A., 1990. Numerical modeling of textured silicon solar cells using PC-1D. *IEEE Trans. Electron Devices* 37 (2), 337–343.

- Battaile, C.C., Srolovitz, D.J., 2002. Kinetic Monte Carlo simulation of chemical vapor deposition. *Annu. Rev. Mater. Res.* 32 (1), 297–319.
- Bayer, B., Duerkop, M., Striedner, G., Sissolak, B., 2021. Model transferability and reduced experimental burden in cell culture process development facilitated by hybrid modeling and intensified design of experiments. *Front. Bioeng. Biotechnol.* 9, 740215.
- Bayer, B., Striedner, G., Duerkop, M., 2020. Hybrid modeling and intensified DoE: an approach to accelerate upstream process characterization. *Biotechnol. J.* 15 (9), 2000121.
- Bazaei, A., Majd, V.J., 2003. Feedback linearization of discrete-time nonlinear uncertain plants via first-principles-based serial neuro-gray-box models. *J. Process Control* 13 (8), 819–830.
- Bellos, G., Kallinikos, L., Gounaris, C., Papayannakos, N., 2005. Modelling of the performance of industrial HDS reactors using a hybrid neural network approach. *Chem. Eng. Process.: Process Intensif.* 44 (5), 505–515.
- Bennett, J.C., Wang, Q., Li, M., Robertson, D.E., Schepen, A., 2016. Reliable long-range ensemble streamflow forecasts: Combining calibrated climate forecasts with a conceptual runoff model and a staged error model. *Water Resour. Res.* 52 (10), 8238–8259.
- Bhadriraju, B., Bangi, M.S.F., Narasingam, A., Kwon, J.S.-I., 2020. Operable adaptive sparse identification of systems: Application to chemical processes. *AIChE J.* 66 (11), e16980.
- Bhadriraju, B., Kwon, J.S.-I., Khan, F., 2021a. OASIS-P: Operable adaptive sparse identification of systems for fault prognosis of chemical processes. *J. Process Control* 107, 114–126.
- Bhadriraju, B., Kwon, J.S.-I., Khan, F., 2021b. Risk-based fault prediction of chemical processes using operable adaptive sparse identification of systems (OASIS). *Comput. Chem. Eng.* 152, 107378.
- Bhadriraju, B., Narasingam, A., Kwon, J.S.-I., 2019. Machine learning-based adaptive model identification of systems: Application to a chemical process. *Chem. Eng. Res. Des.* 152, 372–383.
- Bhutani, N., Rangaiah, G., Ray, A., 2006. First-principles, data-based, and hybrid modeling and optimization of an industrial hydrocracking unit. *Ind. Eng. Chem. Res.* 45 (23), 7807–7816.
- Bikmukhametov, T., Jäschke, J., 2020. Combining machine learning and process engineering physics towards enhanced accuracy and explainability of data-driven models. *Comput. Chem. Eng.* 138, 106834.
- Bishara, D., Li, S., 2023. A machine-learning aided multiscale homogenization model for crystal plasticity: Application for face-centered cubic single crystals. *Comput. Mech.* 72 (1), 77–93.
- Bishop, C.M., Nasrabadi, N.M., 2006. Pattern Recognition and Machine Learning, vol. 4, Springer.
- Bogaerts, A., Bultinck, E., Eckert, M., Georgieva, V., Mao, M., Neyts, E., Schwaederlé, L., 2009. Computer modeling of plasmas and plasma-surface interactions. *Plasma Process. Polym.* 6 (5), 295–307.
- Bohlin, T.P., 2006. Practical Grey-Box Process Identification: Theory and Applications. Springer Science & Business Media.
- Bohlin, T., 2013. Interactive System Identification: Prospects and Pitfalls. Springer Science & Business Media.
- Bohlin, T., Graebe, S.F., 1995. Issues in nonlinear stochastic grey box identification. *Internat. J. Adapt. Control Signal Process.* 9 (6), 465–490.
- Bollas, G., Papadokonstadakis, S., Michalopoulos, J., Arampatzis, G., Lappas, A., Vasalos, I., Lygeros, A., 2003. Using hybrid neural networks in scaling up an FCC model from a pilot plant to an industrial unit. *Chem. Eng. Process.: Process Intensif.* 42 (8–9), 697–713.
- Bradley, W., Kim, J., Kilwein, Z., Blakely, L., Eydenberg, M., Jalvin, J., Laird, C., Boukouvala, F., 2022. Perspectives on the integration between first-principles and data-driven modeling. *Comput. Chem. Eng.* 166, 107898.
- Brandolini, A., Lacunza, M., Ugrin, P., Capiati, N., 1996. High pressure polymerization of ethylene. An improved mathematical model for industrial tubular reactors. *Polym. React. Eng.* 4 (4), 193–241.
- Brunton, S.L., Proctor, J.L., Kutz, J.N., 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proc. Natl. Acad. Sci.* 113 (15), 3932–3937.
- Bui, L., Joswiak, M., Castillo, I., Phillips, A., Yang, J., Hickman, D., 2021. A hybrid modeling approach for catalyst monitoring and lifetime prediction. *ACS Eng. Au* 2 (1), 17–26.
- Cameron, I.T., Hangos, K., 2001. Process Modelling and Model Analysis. Elsevier.
- Carranza-Abaíd, A., González-García, R., 2020. A Petlyuk distillation column dynamic analysis: Hysteresis and bifurcations. *Chem. Eng. Process.-Process Intensif.* 149, 107843.
- Chen, Y., Ierapetritou, M., 2020. A framework of hybrid model development with identification of plant-model mismatch. *AIChE J.* 66 (10), e16996.
- Chen, C., Zhou, L., Ji, X., He, G., Dai, Y., Dang, Y., 2020. Adaptive modeling strategy integrating feature selection and random forest for fluid catalytic cracking processes. *Ind. Eng. Chem. Res.* 59 (24), 11265–11274.
- Chithrananda, S., Grand, G., Ramsundar, B., 2020. ChemBERTa: large-scale self-supervised pretraining for molecular property prediction. arXiv preprint arXiv: 2010.09885.
- Choi, Y., Bhadriaju, B., Cho, H., Lim, J., Han, I.-S., Moon, I., Kwon, J.S.-I., Kim, J., 2023. Data-driven modeling of multimode chemical process: Validation with a real-world distillation column. *Chem. Eng. J.* 457, 141025.
- Choi, H.-K., Kwon, J.S.-I., 2019. Modeling and control of cell wall thickness in batch delignification. *Comput. Chem. Eng.* 128, 512–523.
- Chorin, A.J., 1968. Numerical solution of the Navier-Stokes equations. *Math. Comp.* 22 (104), 745–762.
- Chu, Y., Hahn, J., 2007. Parameter set selection for estimation of nonlinear dynamic systems. *AIChE J.* 53 (11), 2858–2870.
- Côte, M., Grandjean, B.P., Lessard, P., Thibault, J., 1995. Dynamic modelling of the activated sludge process: improving prediction using neural networks. *Water Res.* 29 (4), 995–1004.
- Crose, M., Kwon, J.S.-I., Nayhouse, M., Ni, D., Christofides, P.D., 2015. Multiscale modeling and operation of PECVD of thin film solar cells. *Chem. Eng. Sci.* 136, 50–61.
- Cubillo, F., Callejas, H., Lima, E., Vega, M., 2001. Adaptive control using a hybrid-neural model: application to a polymerisation reactor. *Braz. J. Chem. Eng.* 18, 113–120.
- Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F., 2022. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J. Sci. Comput.* 92 (3), 88.
- De Falco, C., Porro, M., Sacco, R., Verri, M., 2012. Multiscale modeling and simulation of organic solar cells. *Comput. Methods Appl. Mech. Engrg.* 245, 102–116.
- Di Caprio, U., Degrève, J., Hellinckx, P., Waldherr, S., Leblebici, M.E., 2023. HybridGamma: A thermodynamically consistent framework for hybrid modelling of activity coefficients. *Chem. Eng. J.* 475, 146104.
- Dors, M., Simutis, R., Lübbert, A., 1995. Advanced supervision of mammalian cell cultures using hybrid process models. In: Computer Applications in Biotechnology. Elsevier, pp. 72–77.
- Dotson, N.A., Galvan, R., Laurence, R.L., Tirrell, M., 1996. Polymerization Process Modeling. John Wiley & Sons.
- Doyle III, F.J., Harrison, C.A., Crowley, T.J., 2003. Hybrid model-based approach to batch-to-batch control of particle size distribution in emulsion polymerization. *Comput. Chem. Eng.* 27 (8–9), 1153–1163.
- Dubé, M.A., Soares, J.B., Penlidis, A., Hamielec, A.E., 1997. Mathematical modeling of multicomponent chain-growth polymerizations in batch, semibatch, and continuous reactors: a review. *Ind. Eng. Chem. Res.* 36 (4), 966–1015.
- Elmqvist, H., 1978. A Structured Model Language for Large Continuous Systems. Department of Automatic Control, Lund Institute of Technology (LTH).
- Enfors, S.-O., Jahic, M., Rozkov, A., Xu, B., Hecker, M., Jürgen, B., Krüger, E., Schweder, T., Hamer, G., O'beirne, D., et al., 2001. Physiological responses to mixing in large scale bioreactors. *J. Biotech.* 85 (2), 175–185.
- Essenfelder, A.H., Larosa, F., Mazzoli, P., Bagli, S., Broccoli, D., Luzzi, V., Mysiak, J., Mercogliano, P., dalla Valle, F., 2020. Smart climate hydropower tool: a machine-learning seasonal forecasting climate service to support cost-benefit analysis of reservoir management. *Atmosphere* 11 (12), 1305.
- Eugene, E.A., Gao, X., Dowling, A.W., 2020. Learning and optimization with Bayesian hybrid models. In: 2020 American Control Conference. ACC, IEEE, pp. 3997–4002.
- Fearon, O., Kuitunen, S., Ruuttunen, K., Alopaeus, V., Vuorinen, T., 2020. Detailed modeling of kraft pulping chemistry. Delignification. *Ind. Eng. Chem. Res.* 59 (29), 12977–12985.
- Ferreira, A., Dias, J., Von Stosch, M., Clemente, J., Cunha, A., Oliveira, R., 2014. Fast development of *Pichia pastoris* GS115 Mut+ cultures employing batch-to-batch control and hybrid semi-parametric modeling. *Bioprocess Biosyst. Eng.* 37, 629–639.
- Fiedler, B., Schuppert, A., 2008. Local identification of scalar hybrid models with tree structure. *IMA J. Appl. Math.* 73 (3), 449–476.
- Galagali, N., Marzouk, Y.M., 2015. Bayesian inference of chemical kinetic models from proposed reactions. *Chem. Eng. Sci.* 123, 170–190.
- Garg, A., Mhaskar, P., 2017. Subspace identification-based modeling and control of batch particulate processes. *Ind. Eng. Chem. Res.* 56 (26), 7491–7502.
- Gawehn, E., Hiss, J.A., Schneider, G., 2016. Deep learning in drug discovery. *Mol. Inform.* 35 (1), 3–14.
- Georgieva, P., Meireles, M., De Azevedo, S.F., 2003. Knowledge-based hybrid modelling of a batch crystallisation when accounting for nucleation, growth and agglomeration phenomena. *Chem. Eng. Sci.* 58 (16), 3699–3713.
- Germaey, K.V., Gani, R., 2010. A model-based systems approach to pharmaceutical product-process design and analysis. *Chem. Eng. Sci.* 65 (21), 5757–5769.
- Ghiasi, M.M., Mohammadi, A.H., 2017. Application of decision tree learning in modelling CO₂ equilibrium absorption in ionic liquids. *J. Mol. Liq.* 242, 594–605.
- Ghosh, D., Hermonat, E., Mhaskar, P., Snowling, S., Goel, R., 2019. Hybrid modeling approach integrating first-principles models with subspace identification. *Ind. Eng. Chem. Res.* 58 (30), 13533–13543.
- Ghosh, D., Moreira, J., Mhaskar, P., 2021. Model predictive control embedding a parallel hybrid modeling strategy. *Ind. Eng. Chem. Res.* 60 (6), 2547–2562.
- Ghosh, D., Moreira, J., Mhaskar, P., 2022. Application of data-driven modeling approaches to industrial hydroprocessing units. *Chem. Eng. Res. Des.* 177, 123–135.
- Glassey, J., Germaey, K., Clemens, C., Schulz, T.W., Oliveira, R., Striedner, G., Mandenius, C.-F., 2011. Process analytical technology (PAT) for biopharmaceuticals. *Biotechnol. J.* 6 (4), 369–377.

- Golshan, S., Sotudeh-Gharebagh, R., Zarghami, R., Mostoufi, N., Blais, B., Kuipers, J., 2020. Review and implementation of CFD-DEM applied to chemical process systems. *Chem. Eng. Sci.* 221, 115646.
- Grabow, L.C., Gokhale, A.A., Evans, S.T., Dumesic, J.A., Mavrikakis, M., 2008. Mechanism of the water gas shift reaction on Pt: First principles, experiments, and microkinetic modeling. *J. Phys. Chem. C* 112 (12), 4608–4617.
- Gray, F.M., Schmidt, M., 2018. A hybrid approach to thermal building modelling using a combination of Gaussian processes and grey-box models. *Energy Build.* 165, 56–63.
- Greeley, J., Mavrikakis, M., 2004. Alloy catalysts designed from first principles. *Nature Mater.* 3 (11), 810–815.
- Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint [arXiv:2312.00752](https://arxiv.org/abs/2312.00752).
- Guo, B., Li, D., Cheng, C., Lü, Z.-a., Shen, Y., 2001. Simulation of biomass gasification with a hybrid neural network model. *Bioresour. Technol.* 76 (2), 77–83.
- Hansen, E.W., Neurock, M., 1999. Modeling surface kinetics with first-principles-based molecular simulation. *Chem. Eng. Sci.* 54 (15–16), 3411–3421.
- Hassanpour, H., Mhaskar, P., House, J.M., Salsbury, T.I., 2020. A hybrid modeling approach integrating first-principles knowledge with statistical methods for fault detection in HVAC systems. *Comput. Chem. Eng.* 142, 107022.
- Hassanpour, H., Mhaskar, P., Risbeck, M.J., 2022. A hybrid machine learning approach integrating recurrent neural networks with subspace identification for modelling HVAC systems. *Can. J. Chem. Eng.* 100 (12), 3620–3634.
- He, G., Zhou, C., Luo, T., Zhou, L., Dai, Y., Dang, Y., Ji, X., 2020. Online optimization of Fluid Catalytic Cracking process via a Hybrid model based on Simplified structure-Oriented Lumping and case-based reasoning. *Ind. Eng. Chem. Res.* 60 (1), 412–424.
- Himmelblau, D.M., 2008. Accounts of experiences in the application of artificial neural networks in chemical engineering. *Ind. Eng. Chem. Res.* 47 (16), 5782–5796.
- Hoja, J., Reilly, A.M., Tkatchenko, A., 2017. First-principles modeling of molecular crystals: structures and stabilities, temperature and pressure. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* 7 (1), e1294.
- Hong, F., Qiu, K., 2008. An alternative carbon source from konjac powder for enhancing production of bacterial cellulose in static cultures by a model strain acetobacter aceti subsp. *Xylinus ATCC 23770*. *Carbohydr. Polymers* 72 (3), 545–549.
- Hosen, M.A., Hussain, M.A., Mjallal, F.S., 2011. Control of polystyrene batch reactors using neural network based model predictive control (NNMPC): An experimental investigation. *Control Eng. Pract.* 19 (5), 454–467.
- Iglesias-Silva, G.A., Bonilla-Petriciolet, A., Eubank, P.T., Holste, J.C., Hall, K.R., 2003. An algebraic method that includes Gibbs minimization for performing phase equilibrium calculations for any number of components or phases. *Fluid Phase Equilib.* 210 (2), 229–245.
- Jana, A.K., 2018. Chemical Process Modelling and Computer Simulation. PHI Learning Pvt. Ltd..
- Jiang, S.-L., Shen, X., Zheng, Z., 2019. Gaussian process-based hybrid model for predicting oxygen consumption in the converter steelmaking process. *Processes* 7 (6), 352.
- Jørgensen, S.B., Hangos, K.M., 1995. Grey box modelling for control: Qualitative models as a unifying framework. *Internat. J. Adapt. Control Signal Process.* 9 (6), 547–562.
- Kahrs, O., Brendel, M., Michalik, C., Marquardt, W., 2009. Incremental identification of hybrid models of dynamic process systems. In: Model-Based Control: Bridging Rigorous Theory and Advanced Technology. Springer, pp. 185–202.
- Kahrs, O., Marquardt, W., 2007. The validity domain of hybrid models and its application in process optimization. *Chem. Eng. Process.: Process Intensif.* 46 (11), 1054–1066.
- Katrašnik, T., Mele, I., Zelič, K., 2021. Multi-scale modelling of lithium-ion batteries: From transport phenomena to the outbreak of thermal runaway. *Energy Convers. Manage.* 236, 114036.
- Khataei, A., Vahid, B., Behjati, B., Safarpour, M., Joo, S.W., 2014. Kinetic modeling of a triarylmethane dye decolorization by photoelectro-Fenton process in a recirculating system: nonlinear regression analysis. *Chem. Eng. Res. Des.* 92 (2), 362–367.
- Kiparissides, C., Verros, G., Macgregor, J.F., 1993. Mathematical modeling, optimization, and quality control of high-pressure ethylene polymerization reactors. *J. Macromol. Sci. Part C: Polym. Rev.* 33 (4), 437–527.
- Kirilova, E.G., 2022. Artificial neural networks: Applications in chemical engineering. In: Modeling and Simulation in Chemical Engineering: Project Reports on Process Simulation. Springer, pp. 127–146.
- Krippel, M., Dürauer, A., Duerkop, M., 2020. Hybrid modeling of cross-flow filtration: Predicting the flux evolution and duration of ultrafiltration processes. *Sep. Purif. Technol.* 248, 117064.
- Kumari, P., Bhadriraju, B., Wang, Q., Kwon, J.S.-I., 2022. A modified Bayesian network to handle cyclic loops in root cause diagnosis of process faults in the chemical process industry. *J. Process Control* 110, 84–98.
- Kuno, M., Fromm, D., Johnson, S., Gallagher, A., Nesbitt, D.J., 2003. Modeling distributed kinetics in isolated semiconductor quantum dots. *Phys. Rev. B* 67 (12), 125304.
- Kwon, J.S.-I., 2024. Adding big data into the equation. *Nat. Chem. Eng.* (in press).
- Kwon, J.S.-I., Nayhouse, M., Christofides, P.D., Orkoulas, G., 2013. Modeling and control of shape distribution of protein crystal aggregates. *Chem. Eng. Sci.* 104, 484–497.
- Lagergren, J.H., Nardini, J.T., Michael Lavigne, G., Rutter, E.M., Flores, K.B., 2020. Learning partial differential equations for biological transport models from noisy spatio-temporal data. *Proc. R. Soc. A* 476 (2234), 20190800.
- Lauret, P., Boyer, H., Gatina, J.-C., 2000. Hybrid modelling of a sugar boiling process. *Control Eng. Pract.* 8 (3), 299–310.
- Laursen, S.Ö., Webb, D., Ramirez, W.F., 2007. Dynamic hybrid neural network model of an industrial fed-batch fermentation process to produce foreign protein. *Comput. Chem. Eng.* 31 (3), 163–170.
- Leal, J.R., Romanenko, A., Santos, L.O., 2017. Daedalus modeling framework: building first-principle dynamic models. *Ind. Eng. Chem. Res.* 56 (12), 3332–3346.
- Lee, D., Jayaraman, A., Kwon, J.S.-I., 2019. Identification of a time-varying intracellular signalling model through data clustering and parameter selection: application to NF-B signalling pathway induced by LPS in the presence of BFA. *IET Syst. Biol.* 13 (4), 169–179.
- Lee, D., Jayaraman, A., Kwon, J.S., 2020a. Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling. *PLoS Comput. Biol.* 16 (12), e1008472.
- Lee, D., Jayaraman, A., Kwon, J.S.-I., 2020b. Identification of cell-to-cell heterogeneity through systems engineering approaches. *AIChE J.* 66 (5), e16925.
- Lee, D.S., Jeon, C.O., Park, J.M., Chang, K.S., 2002. Hybrid neural network modeling of a full-scale industrial wastewater treatment process. *Biotechnol. Bioeng.* 78 (6), 670–682.
- Lee, C.H., Pahari, S., Sitapure, N., Bartea, M.A., Kwon, J.S.-I., 2023. Investigating high-performance non-precious transition metal oxide catalysts for nitrogen reduction reaction: a multifaceted DFT-kMC-LSTM approach. *ACS Catal.* 13 (13), 8336–8346.
- Lee, D., Singla, A., Wu, H.-J., Kwon, J.S.-I., 2018. An integrated numerical and experimental framework for modeling of CTB and GD1b ganglioside binding kinetics. *AIChE J.* 64 (11), 3882–3893.
- Lee, H., Sitapure, N., Hwang, S., Kwon, J.S.-I., 2021. Multiscale modeling of dendrite formation in lithium-ion batteries. *Comput. Chem. Eng.* 153, 107415.
- Levenberg, K., 1944. A method for the solution of certain non-linear problems in least squares. *Quart. Appl. Math.* 2 (2), 164–168.
- Li, X., Wang, Z., Zhang, L., 2019. Co-estimation of capacity and state-of-charge for lithium-ion batteries in electric vehicles. *Energy* 174, 33–44.
- Li, Y., Yang, Z., Deng, X., Li, N., Li, S., Lei, Z., Eslamimanesh, A., Jin, S., Shen, W., 2023. A mechanism-data hybrid-driven framework for identifying dynamic characteristic of actual chemical processes. *Chem. Eng. Res. Des.* 199, 115–129.
- Li, C., Zhu, Geng, 2007. Multi-objective particle swarm optimization hybrid algorithm: An application on industrial cracking furnace. *Ind. Eng. Chem. Res.* 46 (11), 3602–3609.
- Lima, F.A.R., Rebello, C.M., Costa, E.A., Santana, V.V., de Moares, M.G., Barreto, Jr., A.G., Secchi, A.R., de Souza, Jr., M.B., Nogueira, I.B., 2023. Improved modeling of crystallization processes by Universal Differential Equations. *Chem. Eng. Res. Des.* 200, 538–549.
- Lin, T., Horne, B.G., Tino, P., Giles, C.L., 1996. Learning long-term dependencies in NARX recurrent neural networks. *IEEE Trans. Neural Netw.* 7 (6), 1329–1338.
- Liu, Z.-K., 2009. First-principles calculations and CALPHAD modeling of thermodynamics. *J. Phase Equilib. Diffus.* 30, 517–534.
- Liu, J., Gnanasekar, A., Zhang, Y., Bo, S., Liu, J., Hu, J., Zou, T., 2021. Simultaneous state and parameter estimation: the role of sensitivity analysis. *Ind. Eng. Chem. Res.* 60 (7), 2971–2982.
- Luedeking, R., Piret, E.L., 2000. A kinetic study of the lactic acid fermentation. Batch process at controlled pH. *Biotechnol. Bioeng.* 67 (6), 636–644.
- Luna, M.F., Ochsner, A.M., Amstutz, V., von Blarer, D., Sokolov, M., Arosio, P., Zinn, M., 2021. Modeling of continuous PHA production by a hybrid approach based on first principles and machine learning. *Processes* 9 (9), 1560.
- Mahalec, V., Sanchez, Y., 2012. Inferential monitoring and optimization of crude separation units via hybrid models. *Comput. Chem. Eng.* 45, 15–26.
- Marquardt, D., 1963. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.* 11 (2), 431–441.
- Maton, M., Bogaerts, P., Vande Wouwer, A., 2022. Hybrid dynamic models of bioprocesses based on elementary flux modes and multilayer perceptrons. *Processes* 10 (10), 2084.
- McKay, A., Ghosh, D., Zhu, L., Xi, L., Mhaskar, P., 2022. A novel linear hybrid model predictive control design: application to a fed batch crystallization process. *Digit. Chem. Eng.* 3, 100033.
- Merkelbach, K., Schweidtmann, A.M., Müller, Y., Schwöbel, P., Mhamdi, A., Mitsos, A., Schuppert, A., Mrziglod, T., Schnecke, S., 2022. HybridML: open source platform for hybrid modeling. *Comput. Chem. Eng.* 160, 107736.
- Michalopoulos, J., Papadokonstadakis, S., Arampatzis, G., Lygeros, A., 2001. Modelling of an industrial fluid catalytic cracking unit using neural networks. *Chem. Eng. Res. Des.* 79 (2), 137–142.
- Moayedi, F., Chandrasekar, A., Rasmussen, S., Sarna, S., Corbett, B., Mhaskar, P., 2024. Physics-informed neural networks for process systems: Handling plant-model mismatch. *Ind. Eng. Chem. Res.* 63 (31), 13650–13659.
- Moon, J., Gbadago, D.Q., Hwang, G., Lee, D., Hwang, S., 2022. Software platform for high-fidelity-data-based artificial neural network modeling and process optimization in chemical engineering. *Comput. Chem. Eng.* 158, 107637.
- Mueller, I., Kenig, E.Y., 2007. Reactive distillation in a dividing wall column: rate-based modeling and simulation. *Ind. Eng. Chem. Res.* 46 (11), 3709–3719.

- Nadal-Rey, G., McClure, D.D., Kavanagh, J.M., Cassells, B., Cornelissen, S., Fletcher, D.F., Gernaey, K.V., 2021. Development of dynamic compartment models for industrial aerobic fed-batch fermentation processes. *Chem. Eng. J.* 420, 130402.
- Nagpal, S., Sitapure, N., Gagnon, Z., Kwon, J.S.-I., 2024. Advancing crystal growth prediction: An adaptive kMC model spanning multiple regimes. *Chem. Eng. Sci.* 299, 120472.
- Nagrath, D., Messac, A., Bequette, B.W., Cramer, S., 2004. A hybrid model framework for the optimization of preparative chromatographic processes. *Biotechnol. Prog.* 20 (1), 162–178.
- Narasingam, A., Kwon, J.S.-I., 2017. Development of local dynamic mode decomposition with control: Application to model predictive control of hydraulic fracturing. *Comput. Chem. Eng.* 106, 501–511.
- Narasingam, A., Kwon, J.S.-I., 2018. Data-driven identification of interpretable reduced-order models using sparse regression. *Comput. Chem. Eng.* 119, 101–111.
- Narasingam, A., Kwon, J.S.-I., 2019. Koopman Lyapunov-based model predictive control of nonlinear chemical process systems. *AIChE J.* 65 (11), e16743.
- Narasingam, A., Son, S.H., Kwon, J.S.-I., 2023. Data-driven feedback stabilisation of nonlinear systems: Koopman-based model predictive control. *Internat. J. Control.* 96 (3), 770–781.
- Narayanan, H., Behle, L., Luna, M.F., Sokolov, M., Guillén-Gosálbez, G., Morbidelli, M., Butté, A., 2020. Hybrid-EKF: Hybrid model coupled with extended Kalman filter for real-time monitoring and control of mammalian cell culture. *Biotechnol. Bioeng.* 117 (9), 2703–2714.
- Nascimento, R.G., Fricke, K., Viana, F.A., 2020. A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network. *Eng. Appl. Artif. Intell.* 96, 103996.
- Nellis, G., Klein, S., 2008. Heat Transfer. Cambridge University Press.
- Neurock, M., Hansen, E.W., 1998. First-principles-based molecular simulation of heterogeneous catalytic surface chemistry. *Comput. Chem. Eng.* 22, S1045–S1060.
- Ng, C., Hussain, M.A., 2004. Hybrid neural network—prior knowledge model in temperature control of a semi-batch polymerization process. *Chem. Eng. Process.: Process Intensif.* 43 (4), 559–570.
- Ordonez, M.C., Raftery, J.P., Jaladi, T., Chen, X., Kao, K., Karim, M.N., 2016. Modeling of batch kinetics of aerobic carotenoid production using *Saccharomyces cerevisiae*. *Biochem. Eng. J.* 114, 226–236.
- Osborne, M.R., 1992. Fisher's method of scoring. *Int. Stat. Rev.* 86, 271–286.
- Oussar, Y., Dreyfus, G., 2001. How to be a gray box: dynamic semi-physical modeling. *Neural Netw.* 14 (9), 1161–1172.
- Özdemir, U., Özbay, B., Veli, S., Zor, S., 2011. Modeling adsorption of sodium dodecyl benzene sulfonate (SDBS) onto polyaniline (PANI) by using multi linear regression and artificial neural networks. *Chem. Eng. J.* 178, 183–190.
- Pahari, S., Bhadraju, B., Akbulut, M., Kwon, J.S.-I., 2021a. A slip-spring framework to study relaxation dynamics of entangled wormlike micelles with kinetic Monte Carlo algorithm. *J. Colloid Interface Sci.* 600, 550–560.
- Pahari, S., Moon, J., Akbulut, M., Hwang, S., Kwon, J.S.-I., 2021b. Estimation of microstructural properties of wormlike micelles via a multi-scale multi-recommendation batch bayesian optimization. *Ind. Eng. Chem. Res.* 60 (43), 15669–15678.
- Pahari, S., Shah, P., Kwon, J.S.-I., 2024a. Achieving robustness in hybrid models: A physics-informed regularization approach for spatiotemporal parameter estimation in PDEs. *Chem. Eng. Res. Des.* 204, 292–302.
- Pahari, S., Shah, P., Kwon, J.S.-I., 2024b. Unveiling latent chemical mechanisms: Hybrid modeling for estimating spatiotemporally varying parameters in moving boundary problems. *Ind. Eng. Chem. Res.* 63 (3), 1501–1514.
- Pannala, S., Turner, J.A., Allu, S., Elwasif, W.R., Kalnaus, S., Simunovic, S., Kumar, A., Billings, J.J., Wang, H., Nanda, J., 2015. Multiscale modeling and characterization for performance and safety of lithium-ion batteries. *J. Appl. Phys.* 118 (7).
- Patel, R.G., Manickam, I., Trask, N.A., Wood, M.A., Lee, M., Tomas, I., Cyr, E.C., 2022. Thermodynamically consistent physics-informed neural networks for hyperbolic systems. *J. Comput. Phys.* 449, 110754.
- Piela, P.C., Epperly, T.G., Westerberg, K.M., Westerberg, A.W., 1991. ASCEND: An object-oriented computer environment for modeling and analysis: The modeling language. *Comput. Chem. Eng.* 15 (1), 53–72.
- Pinto, J., Mestre, M., Ramos, J., Costa, R.S., Striedner, G., Oliveira, R., 2022. A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Comput. Chem. Eng.* 165, 107952.
- Pirdashti, M., Curteanu, S., Kamangar, M.H., Hassim, M.H., Khatami, M.A., 2013. Artificial neural networks: applications in chemical engineering. *Rev. Chem. Eng.* 29 (4), 205–239.
- Piron, E., Latrille, E., Rene, F., 1997. Application of artificial neural networks for cross-flow microfiltration modelling: "black-box" and semi-physical approaches. *Comput. Chem. Eng.* 21 (9), 1021–1030.
- Psychogios, D.C., Ungar, L.H., 1992. A hybrid neural network-first principles approach to process modeling. *AIChE J.* 38 (10), 1499–1511.
- Qi, H., Moore, J.B., 2002. Direct Kalman filtering approach for GPS/INS integration. *IEEE Trans. Aerosp. Electron. Syst.* 38 (2), 687–693.
- Quintana-Hernandez, P., Bolanos-Reynoso, E., Miranda-Castro, B., Salcedo-Estrada, L., 2004. Mathematical modeling and kinetic parameter estimation in batch crystallization. *AIChE J.* 50 (7), 1407–1417.
- Raiissi, M., Perdikaris, P., Karniadakis, G.E., 2017. Machine learning of linear differential equations using Gaussian processes. *J. Comput. Phys.* 348, 683–693.
- Raiissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707.
- Raupp, G.B., Alexiadis, A., Hossain, M.M., Changrani, R., 2001. First-principles modeling, scaling laws and design of structured photocatalytic oxidation reactors for air purification. *Catal. Today* 69 (1–4), 41–49.
- Ren, Q., Cheng, H., Han, H., 2017. Research on machine learning framework based on random forest algorithm. In: AIP Conference Proceedings. AIP Publishing.
- Richardson, D., Neal, R., Dankers, R., Mylne, K., Cowling, R., Clements, H., Millard, J., 2020. Linking weather patterns to regional extreme precipitation for highlighting potential flood events in medium-to long-range forecasts. *Meteorol. Appl.* 27 (4), e1931.
- Rodriguez, C., Mhaskar, P., Mahalec, V., 2023. Linear hybrid models of distillation towers. *Comput. Chem. Eng.* 171, 108160.
- Rogers, A.W., Cardenas, I.O.S., Del Rio-Chanona, E.A., Zhang, D., 2023. Investigating physics-informed neural networks for bioprocess hybrid model construction. In: Computer Aided Chemical Engineering, vol. 52, Elsevier, pp. 83–88.
- Romijn, R., Özkan, L., Weiland, S., Ludlage, J., Marquardt, W., 2008. A grey-box modeling approach for the reduction of nonlinear systems. *J. Process Control* 18 (9), 906–914.
- Ross, J., Belgodere, B., Chenthamarakshan, V., Padhi, I., Mroueh, Y., Das, P., 2022. Large-scale chemical language representations capture molecular structure and properties. *Nat. Mach. Intell.* 4 (12), 1256–1264.
- Rumelhart, D.E., Hinton, G.E., Williams, R.J., 1986. Learning representations by back-propagating errors. *Nature* 323, 533–536.
- Safavi, A., Noorai, A., Romagnoli, J., 1999. A hybrid model formulation for a distillation column and the on-line optimisation study. *J. Process Control* 9 (2), 125–134.
- Sansana, J., Joswiak, M.N., Castillo, I., Wang, Z., Rendall, R., Chiang, L.H., Reis, M.S., 2021. Recent trends on hybrid modeling for Industry 4.0. *Comput. Chem. Eng.* 151, 107365.
- Santos, J., Hidalgo, A., Oliveira, R., Velizarov, S., Crespo, J., 2007. Analysis of solvent flux through nanofiltration membranes by mechanistic, chemometric and hybrid modelling. *J. Membr. Sci.* 300 (1–2), 191–204.
- Schäfer, P., Caspari, A., Mhamdi, A., Mitsos, A., 2019. Economic nonlinear model predictive control using hybrid mechanistic data-driven models for optimal operation in real-time electricity markets: In-silico application to air separation processes. *J. Process Control* 84, 171–181.
- Schmidt, F., 2005. Optimization and scale up of industrial fermentation processes. *Appl. Microbiol. Biotechnol.* 68, 425–435.
- Schweidtmann, A.M., Zhang, D., von Stosch, M., 2023. A review and perspective on hybrid modelling methodologies. *Digit. Chem. Eng.* 100136.
- Shah, P., Choi, H.-K., Kwon, J.S.-I., 2023a. Achieving optimal paper properties: A layered multiscale kMC and LSTM-ANN-based control approach for kraft pulping. *Processes* 11 (3), 809.
- Shah, P., Sheriff, M.Z., Bangi, M.S.F., Kravaris, C., Kwon, J.S.-I., Botre, C., Hirota, J., 2022. Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters. *Chem. Eng. J.* 441, 135643.
- Shah, P., Sheriff, M.Z., Bangi, M.S.F., Kravaris, C., Kwon, J.S.-I., Botre, C., Hirota, J., 2023b. Multi-rate observer design and optimal control to maximize productivity of an industry-scale fermentation process. *AIChE J.* 69 (2), e17946.
- Sharma, N., Liu, Y., 2022. A hybrid science-guided machine learning approach for modeling chemical processes: A review. *AIChE J.* 68 (5), e17609.
- Siddhamshtety, P., Kwon, J.S.-I., Liu, S., Valkó, P.P., 2018. Feedback control of proppant bank heights during hydraulic fracturing for enhanced productivity in shale formations. *AIChE J.* 64 (5), 1638–1650.
- Silva, J.A., Costa Neto, E.H., Adriano, W.S., Ferreira, A.L., Gonçalves, L.R., 2008. Use of neural networks in the mathematical modelling of the enzymic synthesis of amoxicillin catalysed by penicillin G acylase immobilized in chitosan. *World J. Microbiol. Biotechnol.* 24, 1761–1767.
- da Silva Pereira, A., Pinheiro, Á.D.T., Rocha, M.V.P., Gonçalves, L.R.B., Cartaxo, S.J.M., 2021. Hybrid neural network modeling and particle swarm optimization for improved ethanol production from cashew apple juice. *Bioprocess Biosyst. Eng.* 44, 329–342.
- Simon, L.L., Fischer, U., Hungerbühler, K., 2006. Modeling of a three-phase industrial batch reactor using a hybrid first-principles neural-network model. *Ind. Eng. Chem. Res.* 45 (21), 7336–7343.
- Singh Sidhu, H., Siddhamshtety, P., Kwon, J.S.-I., 2018. Approximate dynamic programming based control of proppant concentration in hydraulic fracturing. *Mathematics* 6 (8), 132.
- Sitapure, N., Epps, R., Abolhasani, M., Kwon, J.S.-I., 2021a. Multiscale modeling and optimal operation of millifluidic synthesis of perovskite quantum dots: towards size-controlled continuous manufacturing. *Chem. Eng. J.* 413, 127905.
- Sitapure, N., Epps, R.W., Abolhasani, M., Kwon, J.S.-I., 2021b. CFD-based computational studies of quantum dot size control in slug flow crystallizers: Handling slug-to-slug variation. *Ind. Eng. Chem. Res.* 60 (13), 4930–4941.

- Sitapure, N., Kwon, J.S.-I., 2023a. CrystalGPT: Enhancing system-to-system transferability in crystallization prediction and control using time-series-transformers. *Comput. Chem. Eng.* 177, 108339.
- Sitapure, N., Kwon, J.S.-I., 2023b. Exploring the potential of time-series transformers for process modeling and control in chemical systems: an inevitable paradigm shift? *Chem. Eng. Res. Des.* 194, 461–477.
- Sitapure, N., Kwon, J.S.-I., 2023c. Introducing hybrid modeling with time-series-transformers: A comparative study of series and parallel approach in batch crystallization. *Ind. Eng. Chem. Res.* 62 (49), 21278–21291.
- Slater, L., Arnal, L., Boucher, M.-A., Chang, A.Y.-Y., Moulds, S., Murphy, C., Nearing, G., Shalev, G., Shen, C., Speight, L., et al., 2023. Hybrid forecasting: blending climate predictions with AI models. *Hydrol. Earth Syst. Sci.* 27 (9), 1865–1889.
- Sohlberg, B., 2007. Control of a pickling process using a grey box model. *IFAC Proc.* Vol. 40 (11), 249–254.
- Son, S.H., Choi, H.-K., Kwon, J.S.-I., 2020. Multiscale modeling and control of pulp digester under fiber-to-fiber heterogeneity. *Comput. Chem. Eng.* 143, 107117.
- Son, S.H., Choi, H.-K., Moon, J., Kwon, J.S.-I., 2022a. Hybrid Koopman model predictive control of nonlinear systems using multiple EDMD models: An application to a batch pulp digester with feed fluctuation. *Control Eng. Pract.* 118, 104956.
- Son, S.H., Narasingam, A., Kwon, J.S.-I., 2022b. Development of offset-free Koopman Lyapunov-based model predictive control and mathematical analysis for zero steady-state offset condition considering influence of Lyapunov constraints on equilibrium point. *J. Process Control* 118, 26–36.
- von Stosch, M., Davy, S., Francois, K., Galvanauskas, V., Hamelink, J.-M., Luebert, A., Mayer, M., Oliveira, R., O'Kennedy, R., Rice, P., et al., 2014. Hybrid modeling for quality by design and PAT-benefits and challenges of applications in biopharmaceutical industry. *Biotechnol. J.* 9 (6), 719–726.
- Su, H.-T., Bhat, N., Minderman, P., McAvoy, T., 1993. Integrating neural networks with first principles models for dynamic modeling. In: *Dynamics and Control of Chemical Reactors, Distillation Columns and Batch Processes*. Elsevier, pp. 327–332.
- Sun, B., Yang, C., Wang, Y., Gui, W., Craig, I., Olivier, L., 2020. A comprehensive hybrid first principles/machine learning modeling framework for complex industrial processes. *J. Process Control* 86, 30–43.
- Szegedy, C., Toshev, A., Erhan, D., 2013. Deep neural networks for object detection. *Adv. Neural Inf. Process. Syst.* 26.
- Talib, A., Park, S., Im, P., Joe, J., 2023. Grey-box and ANN-based building models for multistep-ahead prediction of indoor temperature to implement model predictive control. *Eng. Appl. Artif. Intell.* 126, 107115.
- Taylor, C.D., 2016. A first-principles surface reaction kinetic model for hydrogen evolution under cathodic and anodic conditions on magnesium. *J. Electrochem. Soc.* 163 (9), C602.
- Taylor, R., Krishna, R., 2000. Modelling reactive distillation. *Chem. Eng. Sci.* 55 (22), 5183–5229.
- Te Braake, H., Van Can, H., Verbruggen, H., 1998. Semi-mechanistic modeling of chemical processes with neural networks. *Eng. Appl. Artif. Intell.* 11 (4), 507–515.
- Teixeira, A.P., Clemente, J.J., Cunha, A.E., Carrondo, M.J., Oliveira, R., 2006. Bioprocess iterative batch-to-batch optimization based on hybrid parametric/non-parametric models. *Biotechnol. Prog.* 22 (1), 247–258.
- Thangavel, S., Subramanian, S., Lucia, S., Engell, S., 2018. Handling structural plant-model mismatch using a model-error model in the multi-stage NMPC framework. *IFAC-PapersOnLine* 51 (15), 1074–1079.
- Thompson, M.L., Kramer, M.A., 1994. Modeling chemical processes using prior knowledge and neural networks. *AIChE J.* 40 (8), 1328–1340.
- Tian, Y., Zhang, J., Morris, J., 2001. Modeling and optimal control of a batch polymerization reactor using a hybrid stacked recurrent neural network model. *Ind. Eng. Chem. Res.* 40 (21), 4525–4535.
- Tornøe, C.W., Jacobsen, J.L., Pedersen, O., Hansen, T., Madsen, H., 2004. Grey-box modelling of pharmacokinetic/pharmacodynamic systems. *J. Pharmacokinet. Pharmacodyn.* 31, 401–417.
- Tsen, A.Y.-D., Jang, S.S., Wong, D.S.H., Joseph, B., 1996. Predictive control of quality in batch polymerizing using hybrid ANN models. *AIChE J.* 42 (2), 455–465.
- Tsivintzelis, I., Kontogeorgis, G.M., Michelsen, M.L., Stenby, E.H., 2010. Modeling phase equilibria for acid gas mixtures using the CPA equation of state. I. Mixtures with H₂S. *AIChE J.* 56 (11), 2965–2982.
- Van Can, H., Te Braake, H., Bijman, A., Hellinga, C., Luyben, K.C.A., Heijnen, J., 1999. An efficient model development strategy for bioprocesses based on neural networks in macroscopic balances: Part II. *Biotechnol. Bioeng.* 62 (6), 666–680.
- Vaswani, A., 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.
- Vega, M., Lima, E., Pinto, J., 1997. Modeling and control of tubular solution polymerization reactors. *Comput. Chem. Eng.* 21, S1049–S1054.
- Van der Ven, A., Aydinol, M., Ceder, G., Kresse, G., Hafner, J., 1998. First-principles investigation of phase stability in Li x CoO 2. *Phys. Rev. B* 58 (6), 2975.
- Von Stosch, M., Oliveira, R., Peres, J., de Azevedo, S.F., 2014. Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Comput. Chem. Eng.* 60, 86–101.
- Wang, G., Briskot, T., Hahn, T., Baumann, P., Hubbuch, J., 2017. Root cause investigation of deviations in protein chromatography based on mechanistic models and artificial neural networks. *J. Chromatogr. A* 1515, 146–153.
- Wang, Y., Wu, Z., 2024. Optimal control of nonlinear systems with input and state constraints using Koopman operator. In: *2024 American Control Conference. ACC*, IEEE, pp. 4078–4083.
- Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., Sun, L., 2022. Transformers in time series: A survey. *arXiv preprint arXiv:2202.07125*.
- Werbos, P.J., 1988. Backpropagation: past and future. In: *Proceedings of the Second International Conference on Neural Network*. Vol. 1, pp. 343–353.
- Williams, M.O., Kevrekidis, I.G., Rowley, C.W., 2015. A data-driven approximation of the koopman operator: Extending dynamic mode decomposition. *J. Nonlinear Sci.* 25, 1307–1346.
- Wu, Z., Rincon, D., Christofides, P.D., 2020. Process structure-based recurrent neural network modeling for model predictive control of nonlinear processes. *J. Process Control* 89, 74–84.
- Wu, Z., Shi, X., 2007. Optimization for high-density cultivation of heterotrophic Chlorella based on a hybrid neural network model. *Lett. Appl. Microbiol.* 44 (1), 13–18.
- Xie, C., Yao, R., Zhu, L., Gong, H., Li, H., Chen, X., 2024. Hybrid dynamic modeling of an industrial reactor network with first-principles and data-driven approaches. *Chem. Eng. Sci.* 119852.
- Xiong, Q., Jutan, A., 2002. Grey-box modelling and control of chemical processes. *Chem. Eng. Sci.* 57 (6), 1027–1039.
- Xu, T., Valocchi, A.J., 2015. Data-driven methods to improve baseflow prediction of a regional groundwater model. *Comput. Geosci.* 85, 124–136.
- Yang, L., Liu, S., 2005. Kinetic model for kraft pulping process. *Ind. Eng. Chem. Res.* 44 (18), 7078–7085.
- Yang, C.-C., Prasher, S.O., Lacroix, R., Kim, S.H., 2003. A multivariate adaptive regression splines model for simulation of pesticide transport in soils. *Biosyst. Eng.* 86 (1), 9–15.
- Yarveicy, H., Saghafi, H., Ghiasi, M.M., Mohammadi, A.H., 2019. Decision tree-based modeling of CO₂ equilibrium absorption in different aqueous solutions of absorbents. *Environ. Prog. Sustain. Energy* 38 (s1), S441–S448.
- Ye, L., Shen, F., Guan, H., 2022. Design of hybrid batch-to-batch and within-batch self-optimizing control structures for batch processes. *J. Process Control* 113, 1–17.
- Yu, Z., Zhang, Z., Jiang, Q., Yan, X., 2024. Neural network-based hybrid modeling approach incorporating Bayesian optimization with industrial soft sensor application. *Knowl.-Based Syst.* 301, 112341.
- Zahedi, G., Lohi, A., Mahdi, K., 2011. Hybrid modeling of ethylene to ethylene oxide heterogeneous reactor. *Fuel Process. Technol.* 92 (9), 1725–1732.
- Zendejhoudi, S., Rezaei, N., Lohi, A., 2018. Applications of hybrid models in chemical, petroleum, and energy systems: A systematic review. *Appl. Energy* 228, 2539–2566.
- Zendejhoudi, S., Zahedi, G., Bahadori, A., Lohi, A., Elkamel, A., Chatzis, I., 2014. A dual approach for modeling and optimization of industrial urea reactor: Smart technique and grey box model. *Can. J. Chem. Eng.* 92 (3), 469–485.
- Zhang, X., Han, M., Yin, X., 2023. Reduced-order Koopman modeling and predictive control of nonlinear processes. *Comput. Chem. Eng.* 179, 108440.
- Zheng, Y., Wu, Z., 2023. Physics-informed online machine learning and predictive control of nonlinear processes with parameter uncertainty. *Ind. Eng. Chem. Res.* 62 (6), 2804–2818.
- Zhou, P., Song, H., Wang, H., Chai, T., 2016. Data-driven nonlinear subspace modeling for prediction and control of molten iron quality indices in blast furnace ironmaking. *IEEE Trans. Control Syst. Technol.* 25 (5), 1761–1774.
- Zhu, R., Pan, E., Chung, P., 2007. Fast multiscale kinetic Monte Carlo simulations of three-dimensional self-assembled quantum dot islands. *Phys. Rev. B—Condens. Matter Mater. Phys.* 75 (20), 205339.