

Article

Solution and Parameter Identification of a Fixed-Bed Reactor Model for Catalytic CO₂ Methanation Using Physics-Informed Neural Networks

Son Ich Ngo  and Young-Il Lim * 

Center of Sustainable Process Engineering (CoSPE), Department of Chemical Engineering, Hankyong National University, Jungang-ro 327, Anseong-si 17579, Korea; ngoichson@hknu.ac.kr

* Correspondence: limyi@hknu.ac.kr; Tel.: +82-31-670-5207

Abstract: In this study, we develop physics-informed neural networks (PINNs) to solve an isothermal fixed-bed (IFB) model for catalytic CO₂ methanation. The PINN includes a feed-forward artificial neural network (FF-ANN) and physics-informed constraints, such as governing equations, boundary conditions, and reaction kinetics. The most effective PINN structure consists of 5–7 hidden layers, 256 neurons per layer, and a hyperbolic tangent (*tanh*) activation function. The forward PINN model solves the plug-flow reactor model of the IFB, whereas the inverse PINN model reveals an unknown effectiveness factor involved in the reaction kinetics. The forward PINN shows excellent extrapolation performance with an accuracy of 88.1% when concentrations outside the training domain are predicted using only one-sixth of the entire domain. The inverse PINN model identifies an unknown effectiveness factor with an error of 0.3%, even for a small number of observation datasets (e.g., 20 sets). These results suggest that forward and inverse PINNs can be used in the solution and system identification of fixed-bed models with chemical reaction kinetics.



Citation: Ngo, S.I.; Lim, Y.-I. Solution and Parameter Identification of a Fixed-Bed Reactor Model for Catalytic CO₂ Methanation Using Physics-Informed Neural Networks. *Catalysts* **2021**, *11*, 1304. <https://doi.org/10.3390/catal11111304>

Academic Editor: Anton Naydenov

Received: 10 September 2021

Accepted: 27 October 2021

Published: 28 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: catalytic CO₂ methanation; fixed-bed reactor; reaction kinetics; system identification; machine learning; physics-informed neural network

1. Introduction

Power-to-gas technology using intermittent surplus renewable electricity has gained attention for mitigating CO₂ emissions to the atmosphere [1,2]. The intermittency of renewable energy sources is a major hurdle in their seamless integration with existing energy systems [3]. CO₂ methanation [4,5] combining captured CO₂ with H₂, produced via water electrolysis [6], is an alternative to existing energy systems that could be integrated with renewable electricity sources. CO₂ methanation technologies could considerably reduce carbon emissions by encouraging industrial symbiosis from industries with large CO₂ footprints [7], such as thermal power plants [8]. Because CH₄ is easier to store and transport than H₂ [1], the synergistic integration of renewable electricity with a natural gas grid is expected via CO₂ methanation [4].

Among the various reactor types, fixed-bed reactors (FBs) are the most commonly used types for CO₂ methanation. As the CO₂ methanation reaction is thermodynamically favored at low temperatures and high pressures [9], an isothermal fixed-bed reactor (IFB) without a hot spot produces high methane selectivity, exhibits stable operation, and prevents deactivation of catalyst particles through processes such as thermal degradation (i.e., nickel sintering [2]). However, the IFB usually requires high recycling and dilution ratios, and adiabatic reactors to maintain suitable productivity [10–12]. The first commercial CO/CO₂ methanation process developed by Lurgi and Sasol (USA, North Dakota) that produces 1.53 billion Nm³/year is composed of an IFB and two adiabatic FB reactors with recycling [10]. In Germany, Linde designed an IFB reactor with an indirect heat exchanger to generate steam from the exothermic heat of reaction [11].

To develop advanced CO₂ methanation technologies, researchers have explored the use of modeling and simulations in the optimization of reactor designs. In particular, computational fluid dynamics (CFD) involving process modeling has been used to understand the hydro- and thermo-dynamics of a reactor following geometrical and operational modifications [13]. CFD studies of CO₂ methanation have been reported for a single fixed-bed [14], multi-stage fixed-beds [14–17], fixed-beds with a plate-type heat exchanger [18,19], monolith reactors [20,21], gas–solid fluidized beds [1,5,22], three-phase slurry bubble columns [23], and microchannel reactors [24,25]. However, there are several limitations of traditional modeling and CFD, such as (1) high computational cost for three-dimensional (3D) multiscale and multiphase CFD [13,18], (2) difficulties in the efficient discretization of complex geometries [13,21], (3) numerical diffusion and round-off errors stemming from numerical differentiation [26], and (4) difficulties in the identification of physical model parameters [1,17,19,27].

Despite advances in first principles and empirical elucidation, artificial neural network (ANN) models in the category of data-driven models, black-box models, or surrogate models (SMs), have become an alternative functional mapping between input and output data because of their prompt predictions, automated knowledge extraction, and high inference accuracy [28–30]. The structure of ANNs relies on the availability of experimental data, observation data [28], or data generated by first-principle models such as CFD [31,32]. The automatic differentiation (AD) technique, which calculates both functions and their derivative values implementing the chain rule, is used across the ANN layers to efficiently estimate gradients [33]. AD has a lower computational cost than symbolic differentiation and a higher accuracy than numerical differentiation [26].

Recently, ANNs and conservation equations coupled with AD that solve ordinary differential equations (ODEs) and partial differential equations (PDEs) called physics-informed neural networks (PINNs) have been reported [32,34,35]. As PINNs are constrained to respect any symmetries, invariances, or first-principle laws [34], they present great potential for solving chemical engineering problems, which usually deal with complex geometries and physics phenomena. In contrast to common ANNs, PINNs do not depend on empirical data because the initial and boundary conditions are directly used to adjust the network parameters, such as weights and biases [34]. In addition, the extrapolation capability of PINNs is enhanced owing to physical constraints [36]. Nevertheless, there are few applications of PINNs in process modeling and chemical reactor design.

PINNs can be used to solve two types of problems: (1) forward and (2) inverse problems [34]. In the forward problem, the PINN solves ODEs/PDEs, like other numerical solvers of ODEs/PDEs, revealing its inference capability. In the inverse problem, unknown physical model parameters are identified using both the well-trained forward PINN and external input/output datasets. Previous PINN studies focused on a general solution of ODEs/PDEs [32,34,35] and elementary reaction rates [29]. Few researchers have addressed PINNs for complex reaction rate models showing high nonlinearity.

In this study, forward and inverse PINNs coupled with AD were developed for the solution and parameter identification of a highly nonlinear reaction rate model for catalytic CO₂ methanation in an IFB reactor. The results obtained from the PINNs were compared with those obtained using a common numerical solver of ODEs (ode15s in MATLAB). The hyper-parameters of the ANN used in the forward PINN, such as (1) the number of hidden layers, (2) number of neurons per hidden layer, (3) activation functions, and (4) number of collocation training points, were systematically determined. In the forward PINN problem, the extrapolation capability was analyzed by narrowing the collocation-training domain and detaching the collocation-training domain from the boundary. In the inverse PINN problem, a reaction effectiveness factor was identified using observation datasets containing 5% and 20% random noises in exact results. The influence of the observation data range on the prediction accuracy of the inverse PINN model was also investigated. This study demonstrates that the forward and inverse PINNs can solve fixed-bed models with highly nonlinear chemical reaction kinetics and identify unknown model parameters.

2. Isothermal Fixed-Bed Reactor for CO₂ Methanation

A single-tube IFB, which is a type of industrial-scale multi-tubular fixed-bed reactor [37,38], with a length (L) of 3 m and a tube diameter (D_{tube}) of 0.01 m was filled with spherical catalyst particles with a particle diameter (d_p) of 1 mm. The single-tube IFB was assumed to be equipped with a heat exchanger that was able to transfer immediately the heat generated in the exothermic reactions to the coolant. The catalytic CO₂ methanation reaction, known as the Sabatier reaction, is [4,14]



Figure 1 shows a tube in the IFB reactor at the reactor and particle scales. The IFB is composed of a single-path tubular reactor at the center and a heat exchanger on the outer wall, where the reactor is externally cooled. The bed voidage (ϵ) between the catalyst particles was assumed to be 0.4 [14]. The operating conditions were set as a temperature (T) of 450 °C, a pressure (P) of 5 bar, and a volumetric flow rate (Q) of 10 Nm³/s. The pure gas reactants were fed to the inlet at a CO₂/H₂ molar ratio of 1/4.

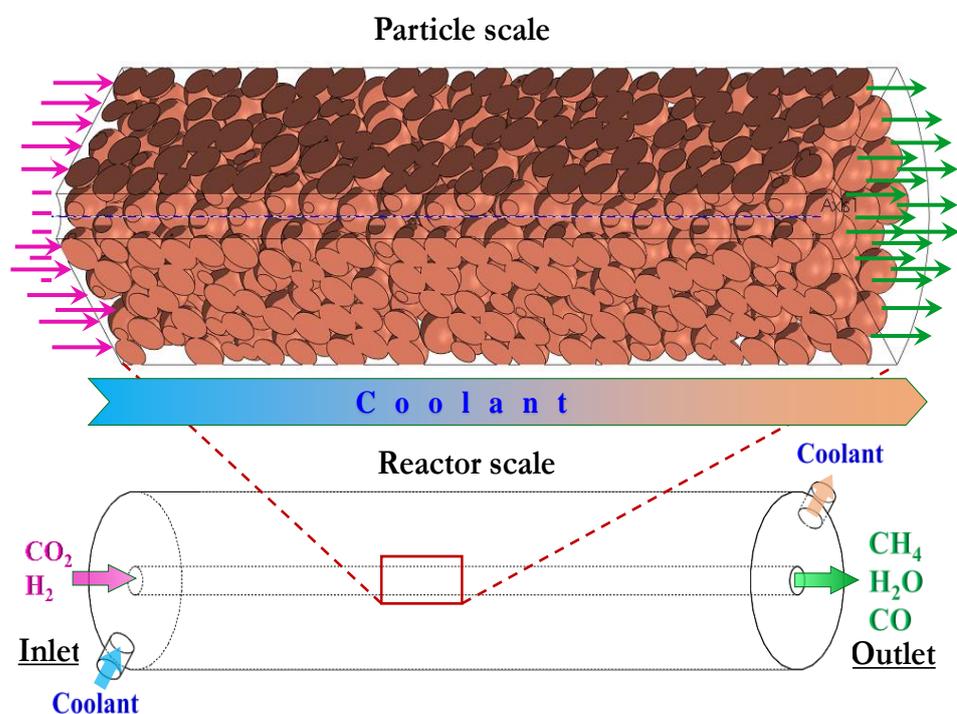


Figure 1. Single tube of an isothermal fixed-bed (IFB) reactor for catalytic CO₂ methanation in the reactor and particle scales.

2.1. Governing Equations for the Isothermal Fixed-Bed Reactor

The IFB was modeled as a one-dimensional (1D) plug-flow reactor at a steady state [14,38]. The momentum and energy balances were neglected because of the low pressure drop and isothermal conditions, respectively. The mass balances for the i th species ($i = \text{CO}_2, \text{H}_2, \text{CH}_4,$ and H_2O) participating in the CO₂ methanation reaction in Equation (1) are formulated as follows:

$$\frac{1}{A_t} \frac{dF_i}{dz} = \eta v_i r \quad (2)$$

where z (m) is the reactor tube axial position, F_i (mol/s) is the molar flow rate of a species i at position z , A_t (m²) is the tube cross-sectional area, v_i is the stoichiometric coefficient of species i , and r (mol/m³/s) is the volumetric reaction rate. η is the effectiveness factor of the chemical reaction, which is defined as the volume-averaged reaction rate with diffusion

within catalyst particles divided by the area-averaged reaction rate at the catalyst particle surface [14]. For the sake of simplicity, the value of η was assumed as one in this study.

The boundary conditions for the molar flow rate (F_i) of the species at the inlet ($z = 0$) are as follows:

$$F_i|_{z=0} = x_{i,0}F_0 \quad (3)$$

where $x_{i,0}$ and F_0 (mol/s) are the inlet mole fraction of gas species i and the total molar flow rate of the inlet gas mixture, respectively. The IFB reactor model expressing the species material balance includes four spatial ODEs in Equation (2) with the boundary condition in Equation (3).

2.2. Reaction Kinetics Model

A reaction kinetics model proposed by Koschany et al. (2016) [39] for catalytic CO₂ methanation, which was tested within a wide range of Ni contents and industrial operating conditions, was adopted in this study.

$$r = \rho_{cat}(1 - \varepsilon)k \cdot \frac{p_{H_2}^{0.31} p_{CO_2}^{0.16}}{1 + K_{ad} \frac{p_{H_2O}}{p_{H_2}^{0.5}}} \left(1 - \frac{p_{CH_4} p_{H_2O}^2}{p_{H_2}^4 p_{CO_2} K_{eq}} \right) \quad (4)$$

$$k = 6.41 \times 10^{-5} \exp\left(\frac{93.6}{R} \left(\frac{1}{555} - \frac{1}{T}\right)\right) \quad (5)$$

$$K_{ad} = 0.62 \times 10^{-5} \exp\left(\frac{64.3}{R} \left(\frac{1}{555} - \frac{1}{T}\right)\right) \quad (6)$$

$$K_{eq} = 137 \cdot T^{-3.998} \exp\left(\frac{158.7}{RT}\right) \quad (7)$$

where R ($=8.314 \times 10^{-3}$ kJ/mol/K) is the gas constant, T (K) is the temperature, p_i (bar) is the partial pressure of species i , k (mol/g_{cat}/s) is the reaction rate constant, K_{ad} (1/bar^{0.5}) is the adsorption constant, and K_{eq} is the thermodynamic equilibrium constant. The catalyst density (ρ_{cat}) was set to 2300×10^3 g_{cat}/m³_{cat} [39]. The reaction rate in Equation (4) including inhibition by adsorbed water (K_{ad}), equilibrium constant (K_{eq}), and non-stoichiometric reaction orders is far from the elementary reaction rate.

3. Physics-Informed Neural Networks (PINN) Model

The 1D IFB reactor model coupled with the reaction kinetics in Equations (2)–(7) is typically solved using a stiff ODE solver. In this study, the solution of the 1D IFB reactor model obtained from a stiff ODE solver was compared with that obtained with the PINNs.

The PINN solving the system of ODEs in the IFB reactor model was composed of an FF-ANN, AD, and a governing equation. A strategy for adjusting the hyper-parameters of the FF-ANN was presented in the forward PINN problem. An unknown model parameter (i.e., η) was identified in the inverse PINN problem. The two PINN structures were the same, while the forward PINN exploited training data self-generated for the initial condition and the inverse PINN used observation data from an external source as the training data.

3.1. PINN Structure in the Forward Problem

The architecture of the forward PINN problem is shown in Figure 2. The objective of the forward PINN problem is to solve the given governing equation with initial, boundary,

and operating conditions. The initial conditions were the target values ($F_{i,0}$) over the reactor length ($0 < z \leq L$) except $z = 0$ at the beginning of the reaction, which are given as follows:

$$F_{i,0}(z) = \begin{cases} 94.74 \frac{\text{mol}}{\text{s}} & \text{for CO}_2 \\ 378.9 \frac{\text{mol}}{\text{s}} & \text{for H}_2 \\ 0 & \text{for CH}_4 \\ 0 & \text{for H}_2\text{O} \end{cases}, 0 < z \leq L \quad (8)$$

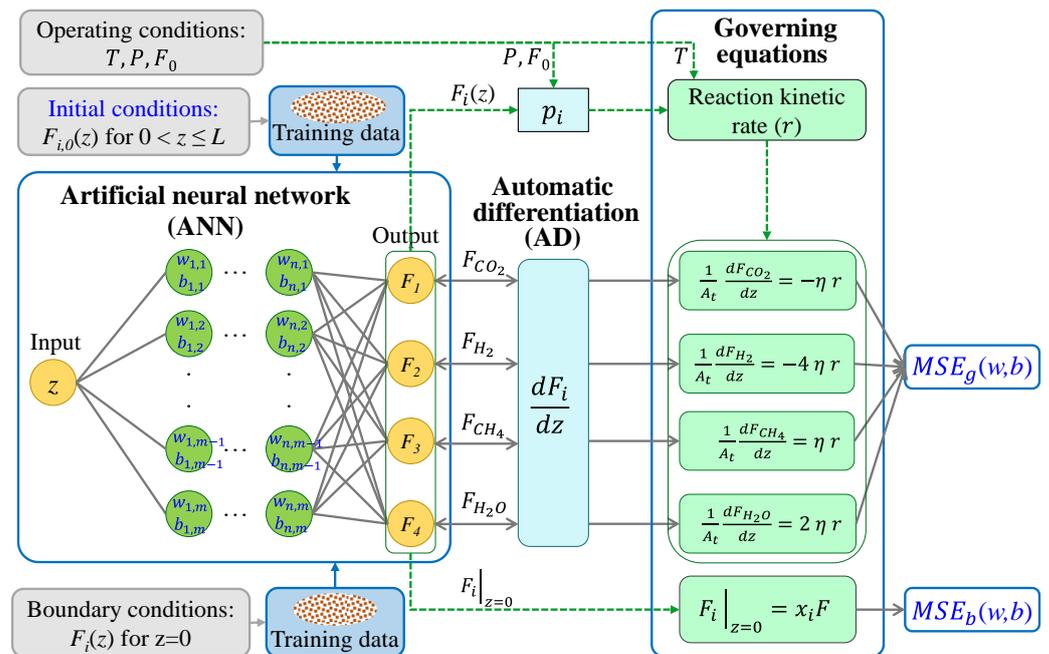


Figure 2. Architecture of the physics-informed neural network (PINN) forward problem for CO₂ methanation in an isothermal fixed-bed (IFB) reactor.

However, any initial condition must be acceptable in theory to adjust w and b because the forward PINN converges to a solution ($F_{i,PINN}$) satisfying the physics-informed constraints. The Dirichlet boundary conditions at $z = 0$ were implemented separately in the PINN:

$$F_i(0) = x_{i,0}F_0 = \begin{cases} 94.74 \frac{\text{mol}}{\text{s}} & \text{for CO}_2 \\ 378.9 \frac{\text{mol}}{\text{s}} & \text{for H}_2 \\ 0 & \text{for CH}_4 \\ 0 & \text{for H}_2\text{O} \end{cases} \quad (9)$$

The operating conditions of the IFB, such as T , P , and F_0 , were used to calculate the partial pressure (p_i) and reaction kinetic rate (r). The FF-ANN structure contained one input (z), four outputs (F_i), n hidden layers, and m neurons for each layer. The input and output datasets of the FF-ANN were randomly sampled from the initial and boundary conditions, Equations (8) and (9), respectively, in the training stage. The activation function (f_a), such as the *sigmoid* and hyperbolic tangent (*tanh*), was applied for each neuron. The weights ($w_{j,k}$) and biases ($b_{j,k}$) for the j th hidden layer and the k th neuron must be adjusted to minimize the loss function ($Loss$). The AD for spatial derivatives ($\frac{dF_i}{dz}$) was calculated via the reverse accumulation mode, which propagates derivatives backward from a given output [26]. The governing equations as the physics-informed part of the ANN included the reaction kinetic rate (r) in Equation (4), the four ODEs in Equation (2), and the boundary conditions in Equation (3).

The optimized weights and biases (w^* and b^*) were obtained from the following optimization problem:

$$\{w^*, b^*\} = \underset{w, b}{\operatorname{argmin}} \{Loss = MSE_g(w, b) + MSE_b(w, b)\} \quad (10)$$

$$MSE_g(w, b) = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} \sum_{i=1}^{N_{comp}} \left| \frac{1}{A_t} \left(\frac{dF_i}{dz} \right)_j - \eta v_i r_j \right|^2 \quad (11)$$

$$MSE_b(w, b) = \frac{1}{N_{bnd}} \sum_{k=1}^{N_{bnd}} \sum_{i=1}^{N_{comp}} |F_{i,k}|_{z=0} - x_{i,0} F_0|^2 \quad (12)$$

where MSE_g and MSE_b are the mean squared errors for the governing equation and boundary condition, respectively. N_{train} , N_{comp} , and N_{bnd} are the number of training data sets, species (or components), and boundary condition sampling points, respectively. The loss function ($Loss$) sums MSE_g and MSE_b .

In Equation (8), for the initial condition, 1000–10,000 training data were randomly and uniformly sampled for the adjustment of the ANN parameters (w and b) and determination of the hyper-parameters (n , m , f_a , and N_{train}). An Adam optimizer [40] was used to solve Equation (10), which combines a stochastic gradient descent with adaptive momentum, because of its good convergence speed [41], as confirmed in several PINN models [29,35,42,43]. An initial learning rate of 0.001 and decay rate of 0.005 were chosen for the Adam optimizer.

Negative intermediate outputs (F_i) appeared frequently when the stochastic gradient optimizer was used in the PINN. However, a negative F_i was unfavorable for solving the IFB reaction model with the reaction kinetics. In addition, as the ODE system of the reactor model with chemical reaction rates was stiff, it was desirable to avoid negative F_i and improve the convergence of the PINN. An exponential mapping of the output values from each hidden layer [29] was used:

$$a_{j,l} = \exp \left(f_a \sum_{k=1}^m [w_{j,k} a_{j-1,k} + b_{j,k}] \right) \quad (13)$$

where $a_{j,l}$ is the value exiting the l th neuron of the j th hidden layer.

Gusmao et al. (2020) [29] presented forward and inverse PINNs to create an SM for the solution of chemical reaction kinetics and to acquire kinetic parameters from experimental data. In our study, the PINN concept was applied to a complex and stiff reaction kinetic problem for CO₂ methanation.

3.2. PINN Structure in the Inverse Problem

The weights (w) and biases (b) of the FF-ANN are the optimized variables in the forward PINN problem, whereas unknown model parameters are identified in the inverse PINN using the optimized weights (w^*) and biases (b^*) obtained from the forward PINN. The inverse PINN included the governing equations, boundary, and operating conditions that were used in the forward PINN, as shown in Figure 3. Rather than using the initial condition as the training data, the inverse PINN problem uses observation data from an external source, such as experimental data. Therefore, the values of z and $F_i(z)$ are different from those in Figure 2.

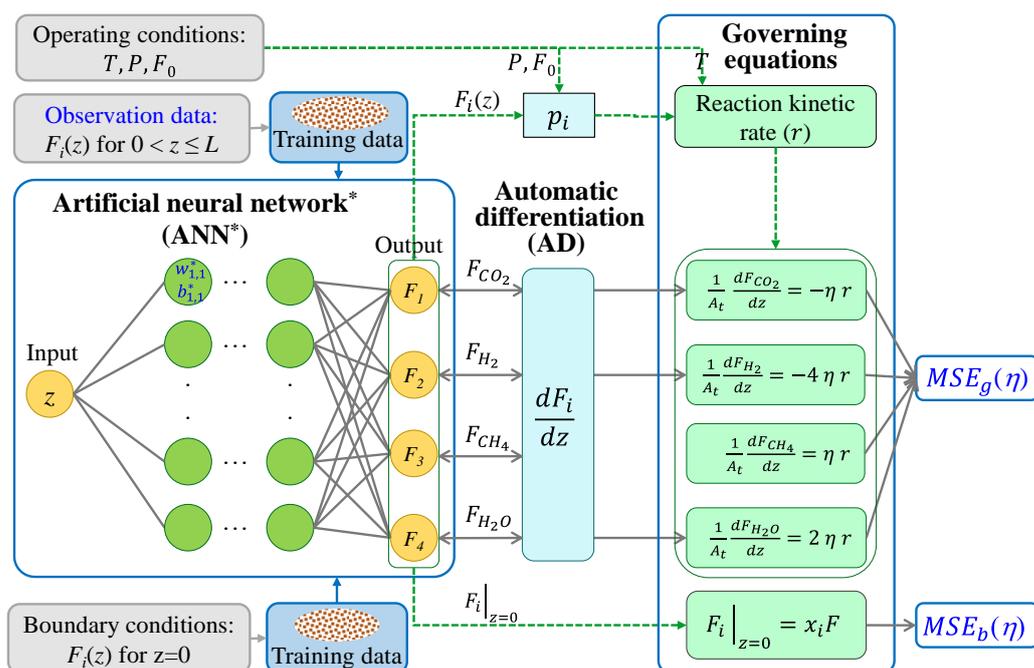


Figure 3. Architecture of the physics-informed neural networks (PINN) inverse problem for CO₂ methanation in an isothermal fixed-bed (IFB) reactor.

In the inverse PINN, the effectiveness factor (η) as an unknown model parameter was identified using the following optimization with a loss function:

$$\{\eta^*\} = \underset{\eta}{\operatorname{argmin}} \{ \text{Loss} = \text{MSE}_g(\eta) + \text{MSE}_b(\eta) \} \quad (14)$$

$$\text{MSE}_g(\eta) = \frac{1}{N_{obs}} \sum_{j=1}^{N_{obs}} \sum_{i=1}^{N_{comp}} \left| \frac{1}{A_t} \left(\frac{dF_i}{dz} \right)_j - \eta v_i r_j \right|^2 \quad (15)$$

$$\text{MSE}_b(\eta) = \frac{1}{N_{bnd}} \sum_{k=1}^{N_{bnd}} \sum_{i=1}^{N_{comp}} |F_{i,k}|_{z=0} - x_{i,0} F_0|^2 \quad (16)$$

where N_{obs} is the number of observation data points (or the experimental data). MSE_g was evaluated for the observation data. The MSE_b in the inverse PINN used the same training data for the Dirichlet boundary condition that was used in the forward problem. The loss function (Loss) sums MSE_g and MSE_b as functions of η .

3.3. Hyper-Parameters Setting and Accuracy of PINN Solution

For the FF-ANN, a mini-batch size of 128, which had a minor effect on the PINN training results, was used. The number of training epochs was set to 1000. The number of the hidden layers (n) ranged from 2 to 11. The number of neurons (m) for each layer was 64–256. The *sigmoid* and *tanh* functions were considered the activation functions (f_a). The number of training data points (N_{train}) varied from 1000 to 10,000. The training data were used to adjust w and b , and to determine the hyper-parameters (m , n , f_a , and N_{train}), because the validation data for determining hyper-parameters were not necessary in the PINN, which provides a solution for physical models.

In the FF-ANN, the biases (b) were initialized to zeros and the weights (w) were initialized by the following commonly used heuristic called Xavier's method [44]:

$$w = U \left[-\sqrt{\frac{6}{N_{in} + N_{out}}}, \sqrt{\frac{6}{N_{in} + N_{out}}} \right] \quad (17)$$

where U is the uniform distribution in the interval of $\pm\sqrt{\frac{6}{N_{in}+N_{out}}}$. N_{in} and N_{out} are the neuron numbers of the previous and present layers, respectively. A pseudo-random generator “*philox*” [45] with 10 rounds and a seed value of “1234” was used to provide the same initial weights for all trainings according to different hyper-parameters. The training data were sampled using Sobol’s quasi-random sequence generator [46], which filled the z space in a highly uniform manner.

The PINNs were implemented using the deep learning toolbox of MATLAB (Mathworks, R2021a, Natick, MA, USA, 2021). The PINNs were executed on a single NVIDIA Quadro RTX 6000 GPU device. The computational time was 5 min to 3 h to train each forward PINN according to the number of hidden layers and neurons. The inverse PINN model required less than 20 s.

The governing equation in Equation (2) with the boundary conditions in Equation (3) was also solved using a stiff ODE numerical solver, *ode15s* in MATLAB, with a strict relative and absolute tolerance of 1×10^{-8} . The solution was considered as an exact solution. The accuracy of the PINN model was measured using an L_2 relative error norm ($L_{2,rel}$) [30] between the PINN solution ($F_{i,PINN}$) and stiff ODE solution ($F_{i,ODE}$):

$$L_{2,rel} = \sqrt{\frac{\sum_{i=1}^{N_{comp}} \sum_{j=1}^{N_{test}} (F_{i,PINN}^j - F_{i,ODE}^j)^2}{\sum_{i=1}^{N_{comp}} \sum_{j=1}^{N_{test}} (F_{i,ODE}^j)^2}} \quad (18)$$

N_{test} (=1000) is the number of test data in the forward PINN, whereas N_{obs} is the number of observation data points (N_{obs}) in the inverse PINN. The test and observation data were generated uniformly over a given range of reactor lengths (z).

4. Results and Discussion

The hyper-parameters (m , n , f_a , and N_{train}) of the forward PINN were first determined. Then, the extrapolation performance of the forward PINN was investigated using 10,000 training data points sampled at a limited reactor length (z) and 1000 test data points sampled at the full reactor length ($0 \leq z \leq 3$). Using the optimized structure of the forward PINN, the effects of the number and range of the observation data on unknown model parameters (e.g., η) were examined in the inverse PINN problem.

4.1. Determination of Hyper-Parameters

Activation functions such as the *sigmoid* and *tanh* functions were first tested in an FF-ANN structure with four hidden layers and 64 neurons per layer. Then the number of layers and neurons were determined for the FF-ANN using an activation function selected previously. The *sigmoid* activation function showed good performance for a specific FF-ANN structure (e.g., four hidden layers and 64 neurons), while the *tanh* activation function exhibited good results for almost all network structures.

Figure 4 shows the comparison between *sigmoid* and *tanh* in terms of (a) the loss function and (b) F_i along the reactor length (z) for 1000 test data points evenly distributed in $0 \leq z \leq 3$. The 10,000 training data points used to adjust w and b were distributed by Sobol’s quasi-random sequence generator, as mentioned earlier. The loss function (*Loss*) obtained using *tanh* was lower by three orders of magnitude than that obtained using the *sigmoid* function (Figure 4a). Here, the iteration number is the number of mini-batches multiplied by the number of epochs. As a result, the mole flow rates ($F_{i,PINN}$) obtained from the PINN with *tanh* were closer to the ODE solution ($F_{i,ODE}$) than those obtained from the PINN with the *sigmoid* function (Figure 4b). The $L_{2,rel}$ of the *sigmoid* and *tanh* functions were 0.05251 and 0.02372, respectively. Therefore, the *tanh* activation function for all the hidden layers was chosen for further investigation. The mixed activation functions and other activation functions, such as the rectified linear unit (ReLU), were outside the scope of this study.

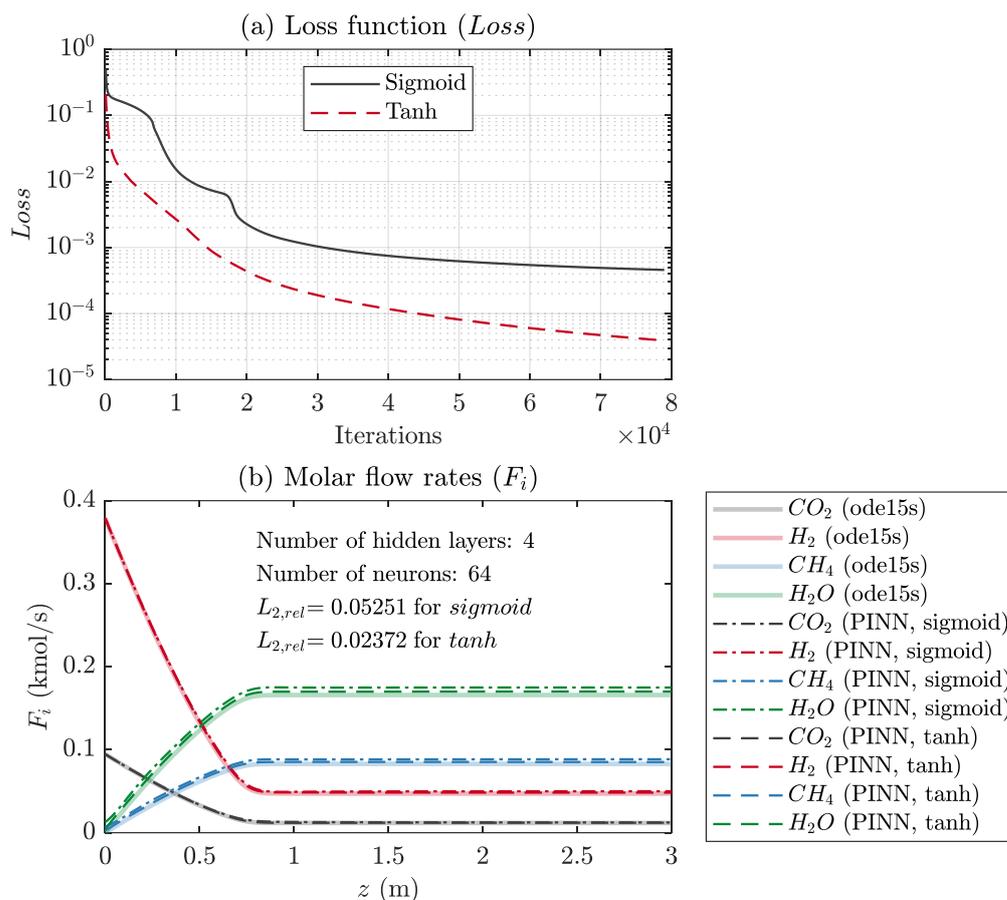


Figure 4. (a) History of the loss function ($Loss$) and (b) comparison of the mole flow rates (F_i) of the exact ODE solutions ($F_{i,ODE}$) and PINN solutions ($F_{i,PINN}$) with the *sigmoid* and *tanh* activation functions.

Figure 5 shows the influence of the number of hidden layers and neurons in each layer on the loss function ($Loss$), L_2 relative error ($L_{2,rel}$), and training time (t). The loss function and training time were obtained from the PINN with *tanh* for 10,000 training data points. $L_{2,rel}$ was measured for 1000 test data compared to the ODE solution. The red spot indicates the forward PINN structure with $n = 7$ layers and $m = 256$ neurons achieved a minimum loss.

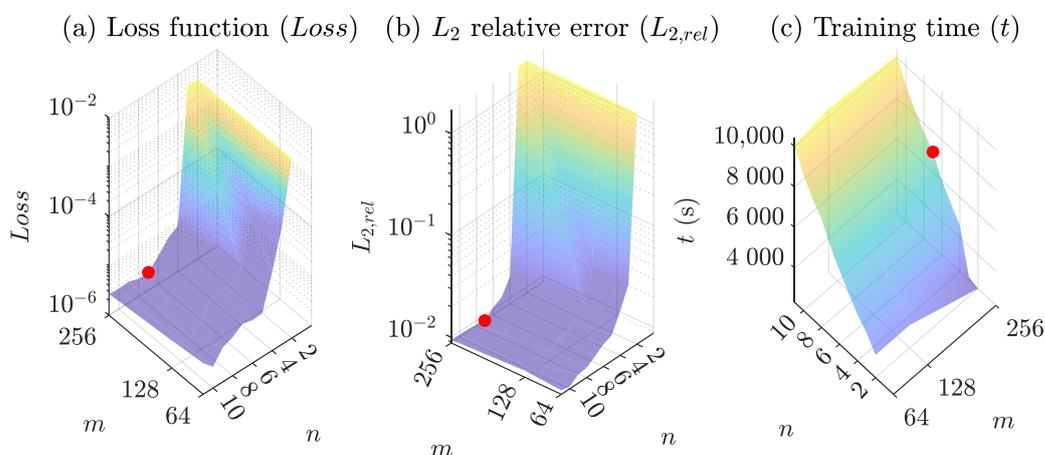


Figure 5. (a) Loss function ($Loss$), (b) L_2 relative error ($L_{2,rel}$), and (c) training time (t) with respect to the number of layers (n) and neurons (m) for each layer.

$Loss$ and $L_{2,rel}$ sharply decrease for $2 \leq n \leq 5$ and slowly converge to a certain value for all the investigated numbers of neurons (see Figure 5a,b). The number of neurons (m) weakly influenced $Loss$ and $L_{2,rel}$, which were the lowest for the forward PINN structure with 256 neurons. Although the minimum values of $Loss = 2.3 \times 10^{-6}$ and $L_{2,rel} = 0.00881$ were obtained with the forward PINN structure with seven layers and 256 neurons, the variation in $L_{2,rel}$ was negligible for forward PINN structures with more than four layers, at less than 0.69%. The computational time (t) for the training increased almost linearly with the number of hidden layers (m). However, the training time (t) did not increase as the number of neurons in each layer increased. This may be attributed to the fast convergence achieved with a high number of neurons. The FF-ANN structure with seven hidden layers and 256 neurons was chosen for the CO₂ methanation IFB reactor model.

Figure 6 compares the performance of the PINN with two, five, and seven hidden layers, using 256 neurons for each layer. The $L_{2,rel}$ decreases with an increase in the number of hidden layers, as shown in Figure 5. The solution obtained from the PINN with two layers and 256 neurons deviated significantly from the ODE solution (Figure 6a). The solution obtained from the PINN with seven layers showed excellent agreement with the ODE solution at high computational cost (Figure 6c).

Using the PINN with seven layers and 256 neurons, the influences of the number of training data points (N_{train}) on the loss function ($Loss$) and training time (t) are depicted in Figure 7. As N_{train} increases, the $Loss$ converges, and t increases proportionally. For the error between the PINN and ODE solutions to be sufficiently small, it is desirable that N_{train} for the 1D reactor model be over 5000.

4.2. Computational Efficiency of the PINN

A single training time (t) was approximately two hours for the PINN with $m = 256$, $n = 7$, $tanh$, and $N_{train} = 10,000$, as shown in Figure 7. A substantial computational time was required to determine all hyper-parameters of PINN, whereas the ODE numerical solver (e.g., ode15s) showed fast calculation due to no training stage. However, once the hyper-parameters were determined, and weights (w) and biases (b) were optimized, the PINN surrogate model against the ODE numerical solver had the advantage in computational time.

In Figure 8, the computational times of the PINN surrogate model and the ODE numerical solver are compared with respect to the number of spatial points from 1000 to 200,000 with an interval of 1000. The PINN with $n = 7$, $m = 256$ and $tanh$ activation function was used. The calculation time (t) increases with the increase in the number of spatial points or N_{test} . The calculation time of PINN is less than 0.1 s at $N_{test} = 200,000$, while that of ODE solver with 200,000 spatial points is approximately 0.95 s. The PINN surrogate model has computational efficiency over the ODE solver and it will be useful for optimization problems repeating the calculation of ODEs/PDEs.

4.3. Extrapolation Capability of the PINN

If a sufficient amount of training data is provided (see Figure 7), the forward PINN guarantees a solution ($F_{i,PINN}$) that satisfies the governing equation, as mentioned earlier. The PINN has an extrapolation capability when applied for the range out of training data, which is similar to solving first-principle laws in a computational domain.

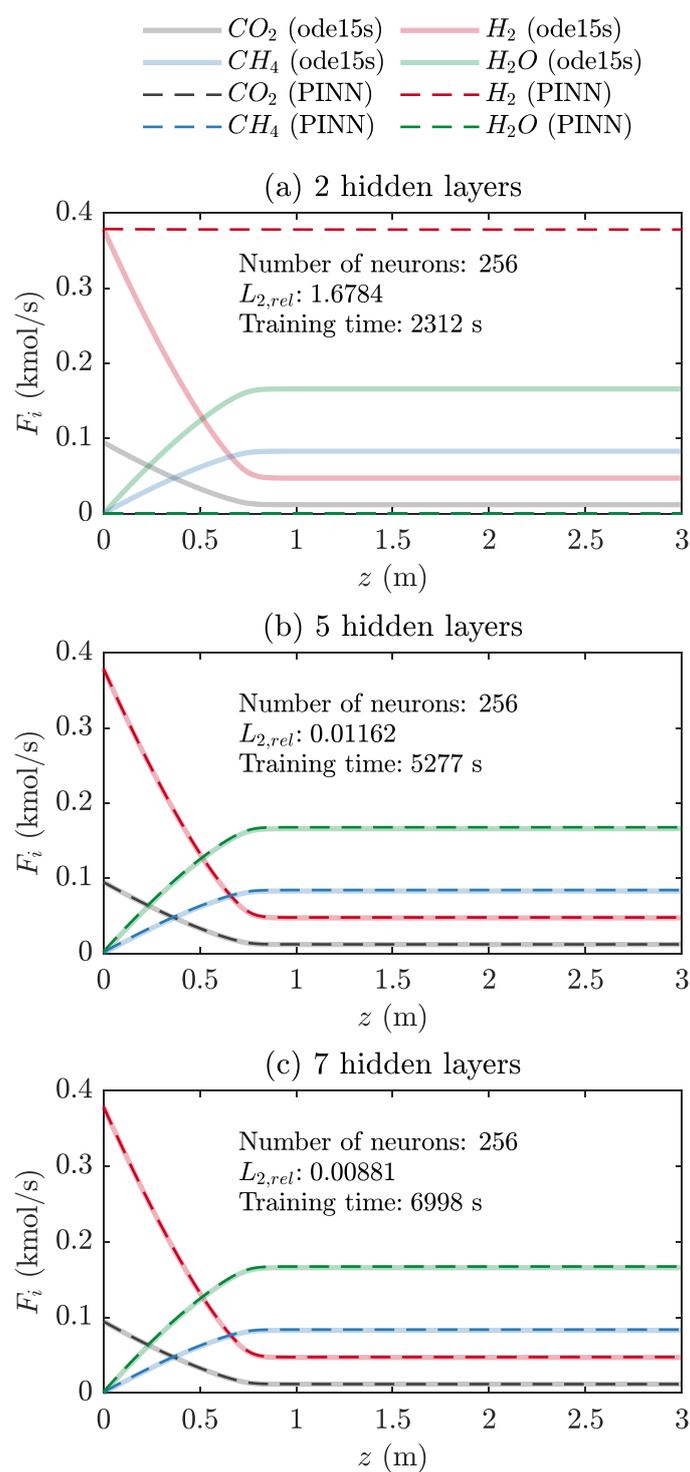


Figure 6. Performance of PINN models with different numbers of hidden layers: (a) 2 hidden layers, (b) 5 hidden layers, and (c) 7 hidden layers.

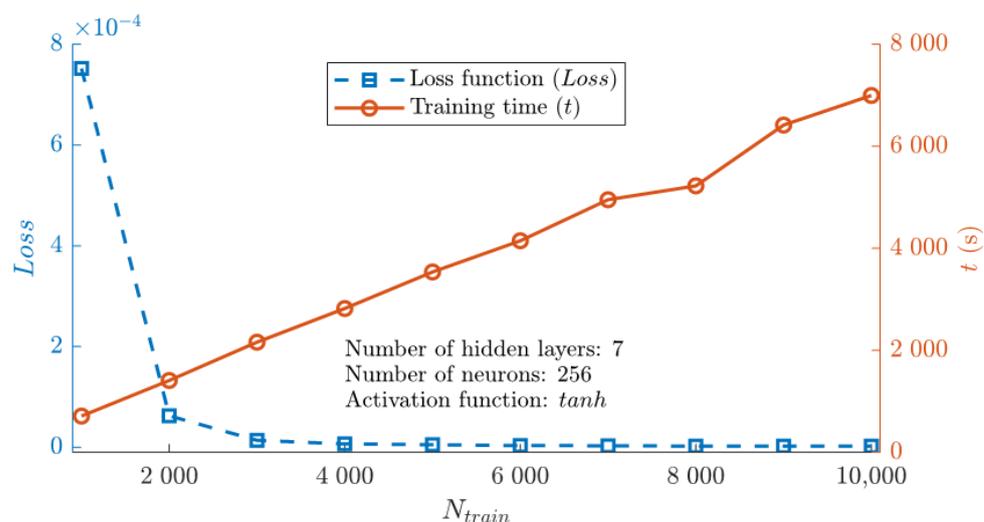


Figure 7. Influence of the number of training data points (N_{train}) on the loss function ($Loss$) and training time (t).

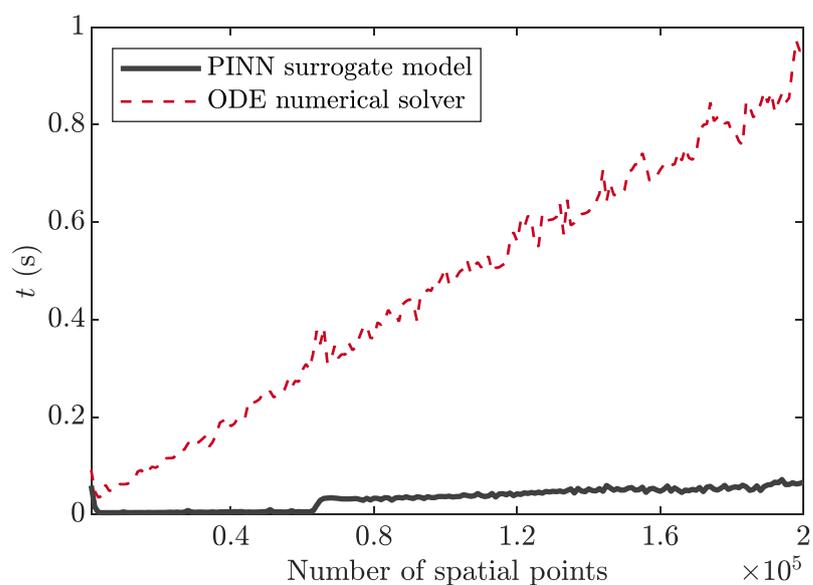


Figure 8. Comparison of calculation time between PINN surrogate model and ODE numerical solver with respect to the number of spatial points.

Figure 9 shows the performance of the forward PINN for 10,000 training data points in a limited range of z and 1000 test data points in a full range of z ($0 \leq z \leq 2$), using five hidden layers, 256 neurons per layer, and the \tanh activation function. The collocation range of the training data starts from $z = 0$ and ends at $z = 0.5$ – 1.0 , with an interval of 0.1 in Figure 9a–f. Even though the PINN was trained within one sixth ($0 \leq z \leq 0.5$) of the full range, the PINN output ($F_{i,PINN}$) for the test data of the full range ($0 \leq z \leq 2$) agrees well the ODE solution ($F_{i,ODE}$) outside the training range (Figure 9a). $L_{2,rel}$ decreases, and the training time tends to increase as the training range increases.

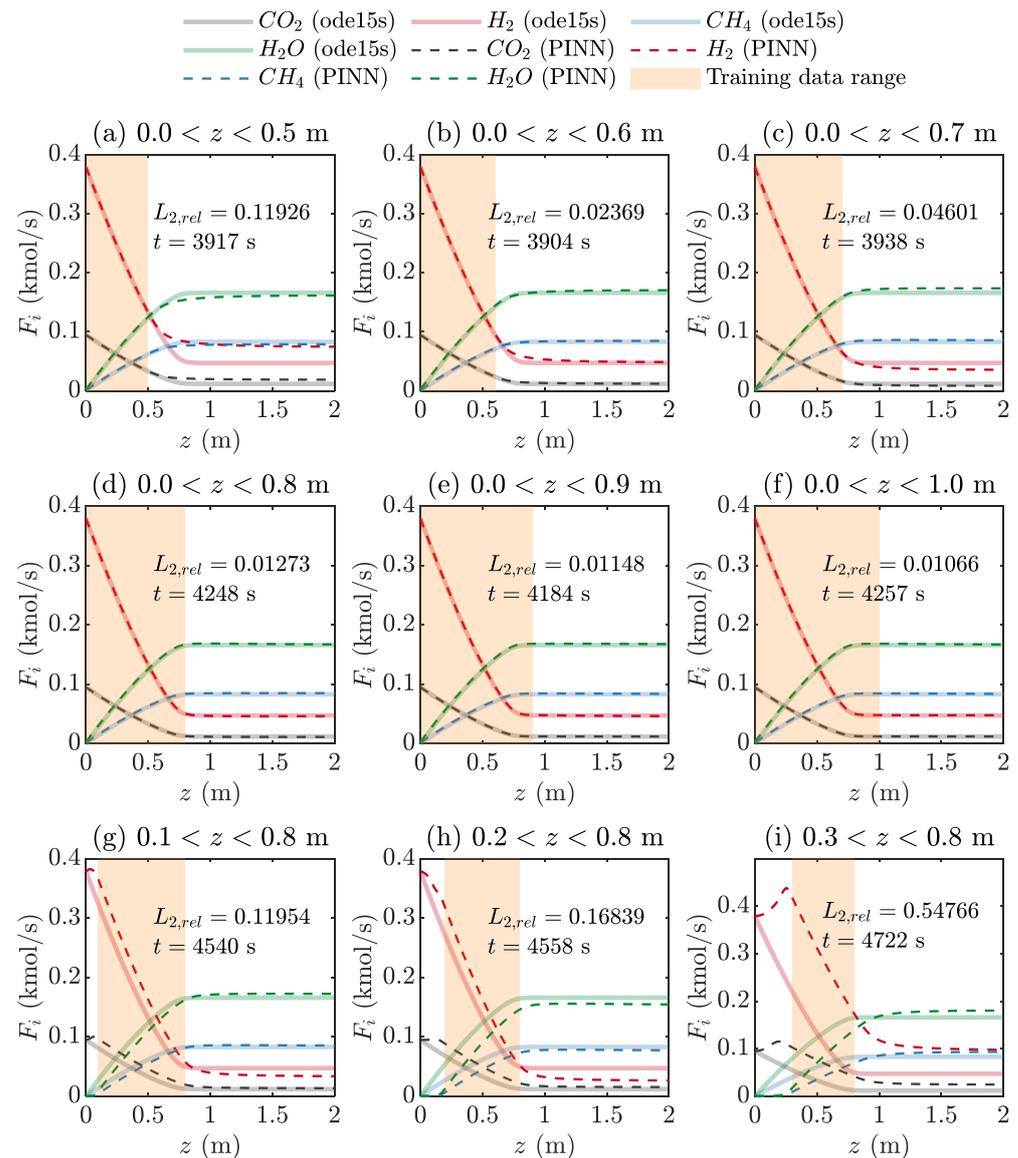


Figure 9. Performance of the forward PINN model using five hidden layers, 256 neurons, and the *tanh* activation function for 10,000 training data points in a limited range of the reactor length (z) and 1000 test data points in the full range of z : (a) $0 < z < 0.5$ m, (b) $0 < z < 0.6$ m, (c) $0 < z < 0.7$ m, (d) $0 < z < 0.8$ m, (e) $0 < z < 0.9$ m, (f) $0 < z < 1.0$ m, (g) $0.1 < z < 0.8$ m, (h) $0.2 < z < 0.8$ m, and (i) $0.3 < z < 0.8$ m.

Figure 9g–i show the performance of the PINN that was trained for three data ranges detached from $z = 0$, reducing the width of the range of z . The value of $F_i|_{z=0}$ is correct for all data ranges detached from $z = 0$ owing to the boundary condition in Equation (3). However, $F_{i,PINN}$ between $z = 0$ and the beginning point of the training data is far from $F_{i,ODE}$, and the errors between $F_{i,PINN}$ and $F_{i,ODE}$ are persistent in the other ranges. As the governing equation used in this study is a type of initial value problem, sufficient training data close to $z = 0$ must be provided to obtain reliable PINN solutions.

The extrapolation capability of the PINN is remarkable, unlike that of common ANNs [28,30]. The accuracy of the PINN solution is closely related to the range and distribution of the training data [35]. The present forward PINN model for solving governing equations can overcome the drawbacks of traditional CFD modeling and simulation methods, as mentioned in the introduction. Once the PINN parameters (hyper-parameters, weights, and biases) are optimized, the PINN instantly predicts outputs (F_i) corresponding

to inputs (z), which can be used as an excellent SM of the governing equation. Because training data at spatial collocation points (z) are generated independently against the specific spatial domain, the forward PINN model is appropriate for solving governing equations with complex geometries or moving boundary conditions [47]. In addition, numerical diffusion and round-off errors are minimized in PINNs with the aid of AD [48].

4.4. Identification of Unknown Model Parameters with the Inverse PINN

The trained PINN with seven layers, 256 neurons per layer, and the *tanh* activation function was used in the inverse PINN problem to identify an unknown parameter (i.e., η). The number of epochs, initial learning rate, and decay rate of the inverse PINN model were set to 500, 0.03, and 0.005, respectively. The initial η value was assumed to be 1×10^{-6} .

Figure 10 shows the influence of the number of observation data points (N_{obs}) and noise ratios on the identification performance of the effectiveness factor (η). N_{obs} had a range of 20–1000. Initially, the ODE solution ($F_{i,ODE}$) for the governing equation with $\eta = 1$ and $0 \leq z \leq 2$ was obtained using a stiff ODE solver (i.e., ode15s in MATLAB), which is the mean value of $F_i(z)$. The observation data were randomly and uniformly generated, satisfying a normal distribution with a mean of $F_i(z)$ and standard deviation as the noise ratios of 5% and 20%. The $F_{i,PINN}$ displayed by the solid, dashed, and dotted lines in Figure 10 was acquired from the forward PINN with η^* obtained from the inverse PINN.

It is expected that the six inverse PINN problems in Figure 10 result in $\eta^* = 1$ because all observation data were generated for $\eta = 1$. N_{obs} influence the value of η^* more significantly than the noise ratio. If the number of observation data is the same, the noise ratio hardly affects η^* . Even though a small amount of observation data ($N_{obs} = 20$) were used, the error between the exact and predicted η values was only 0.3% (see Figure 10a,b). The inverse PINN model inherits the accuracy of the forward PINN model. Thus, the inverse PINN model with well-trained weights and biases can accurately identify model parameters, even for a small amount of observation data.

The effect of the collocation range of observation data on η^* was investigated for the inverse PINN model, as shown in Figure 11. The collocation range of the observation data significantly influences the identification of η . High accuracy was achieved when the collocation range of the observation data was close to the boundary ($z = 0$), as shown in Figure 11a,b. When observation data far from the boundary are provided, it was difficult for the inverse PINN model to identify the model parameter (η), as shown in Figure 11c. The inverse PINN problem took approximately 20 s for 1000 observations. If the forward PINN is well-trained and the observation data are properly provided, the inverse PINN model can identify unknown model parameters more efficiently than other computationally intensive methods such as CFD.

4.5. Extension of PINN to Different Effectiveness Factors

The current PINN can be extended to the ODE solutions for different process conditions, such as the effectiveness factor (η). One more neuron for $0 \leq \eta \leq 1$ was added into the input layer, which results in the input layer with z and η , and the identical output layer (F_i). The boundary condition with the combination of z and η was used as follows:

$$F_i(0, \eta_0) = x_{i,0}F_0 \text{ and } F_i(0, \eta_1) = x_{i,0}F_0 \quad (19)$$

where $\eta_0 = 0$ and $\eta_1 = 1$ are the bounds of η .

The previous optimized network structure with five hidden layers, 256 neurons per layer, and *tanh* activation function was used. The number of training points (N_{train}) was increased from 10,000 to 30,000 because the two input variables (z and η) were applied. The number of epochs was also increased from 1000 to 2000 to enhance the convergence during training. Figure 12 shows the PINN predictions $F_i(z)$ for $\eta = 0.6, 0.8$, and 1.2 , where $N_{test} = 1000$. The PINN captured the $F_i(z)$ at any η . Excellent prediction for $F_i(z)$ at $\eta = 1.2$, as an extrapolation was observed, as shown in Figure 12c. The PINN surrogate

model can be used for an optimization problem of the CO₂ methanation process with computational efficiency.

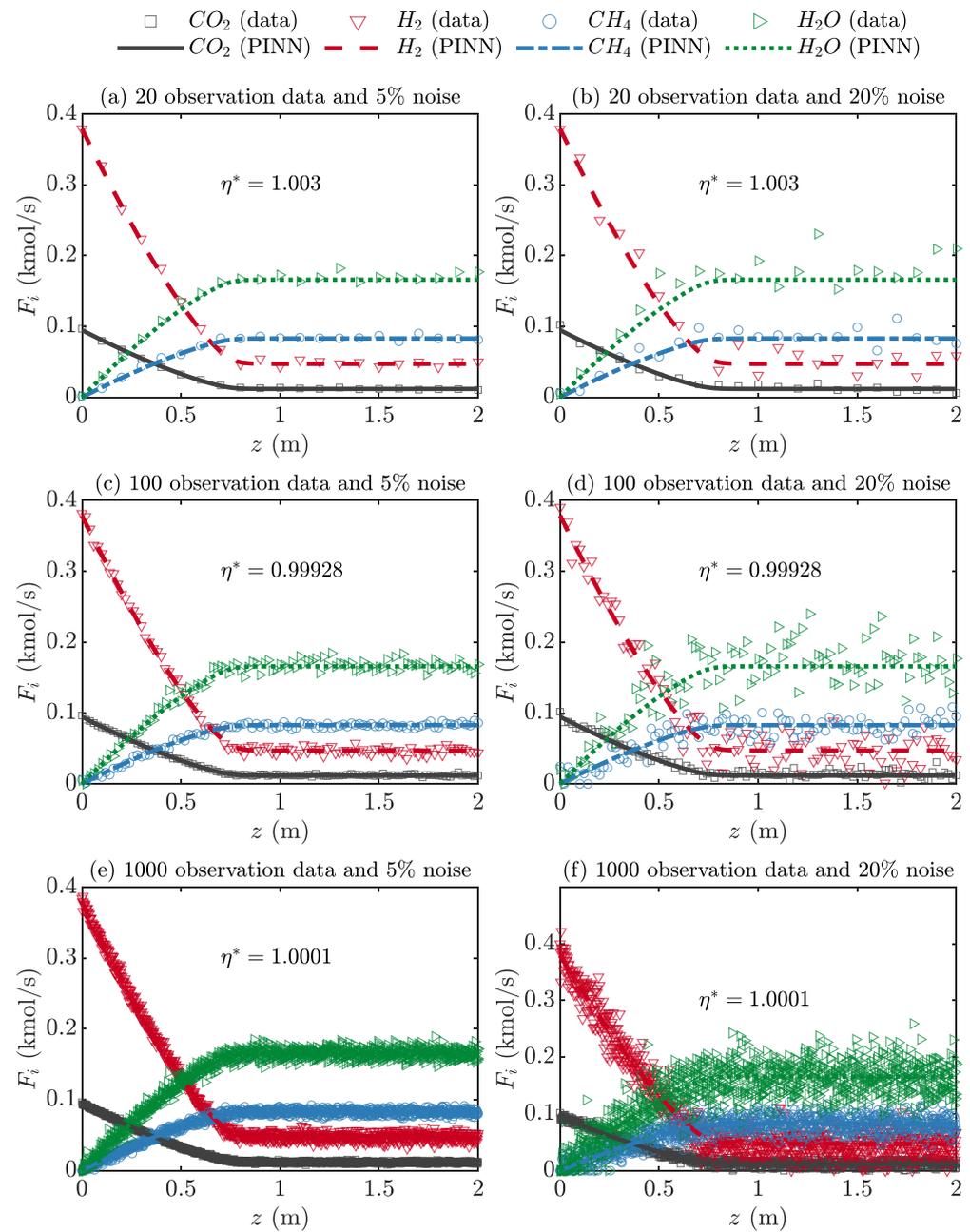


Figure 10. Influence of the number of observation data points (N_{obs}) and noise ratios on the identification performance of the effectiveness factor (η) of an inverse PINN with seven layers, 256 neurons per layer, and the \tanh activation function: (a) 20 data and 5% noise, (b) 20 data and 20% noise, (c) 100 data and 5% noise, (d) 100 data and 20% noise, (e) 1000 data and 5% noise, and (f) 1000 data and 20% noise.

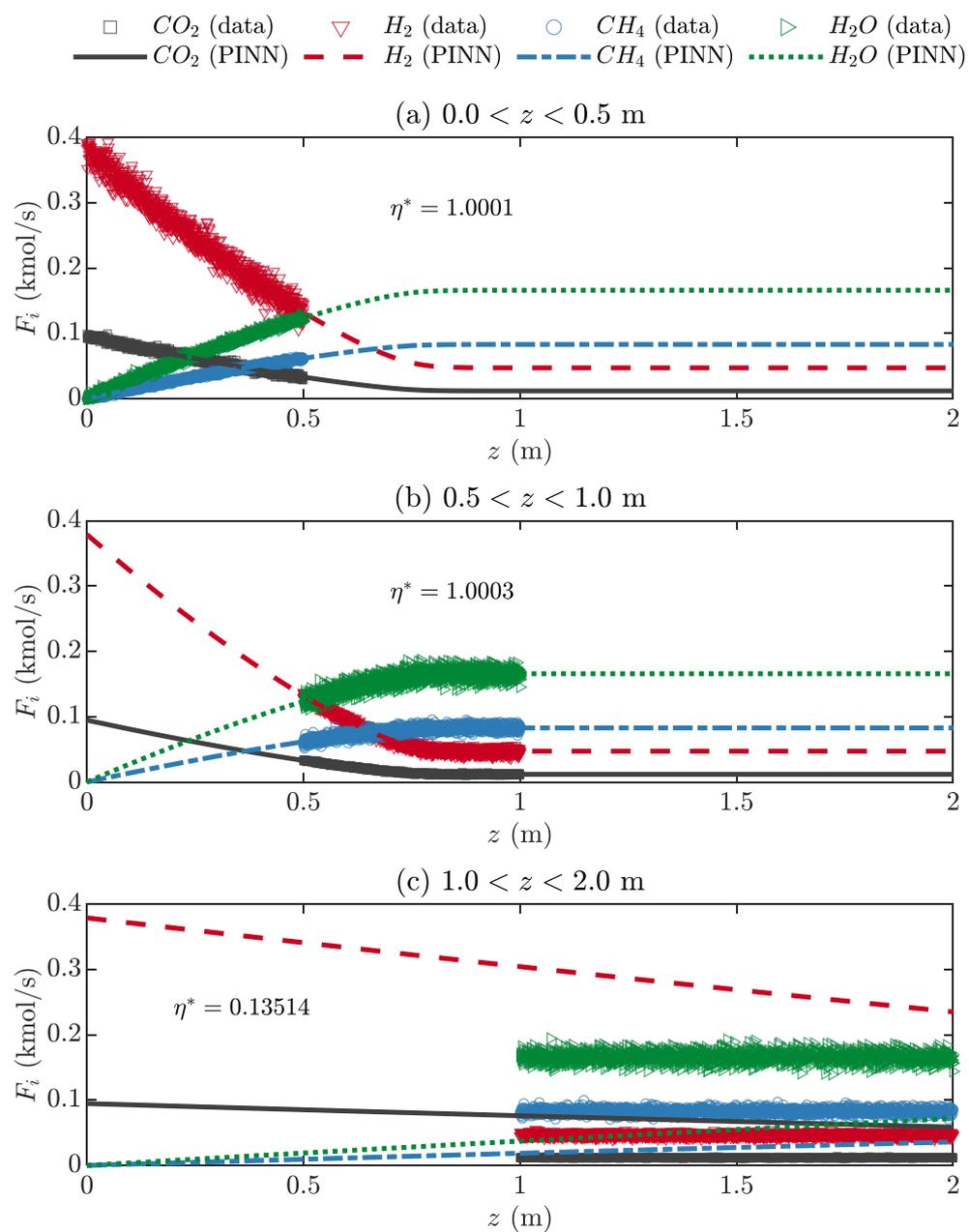


Figure 11. Performance of the inverse PINN model with seven hidden layers, 256 neurons, and the \tanh activation function for 1000 observation data points in a limited range of the reactor length (z): (a) $0 < z < 0.5$ m, (b) $0.5 < z < 1.0$ m, and (c) $1.0 < z < 2.0$ m.

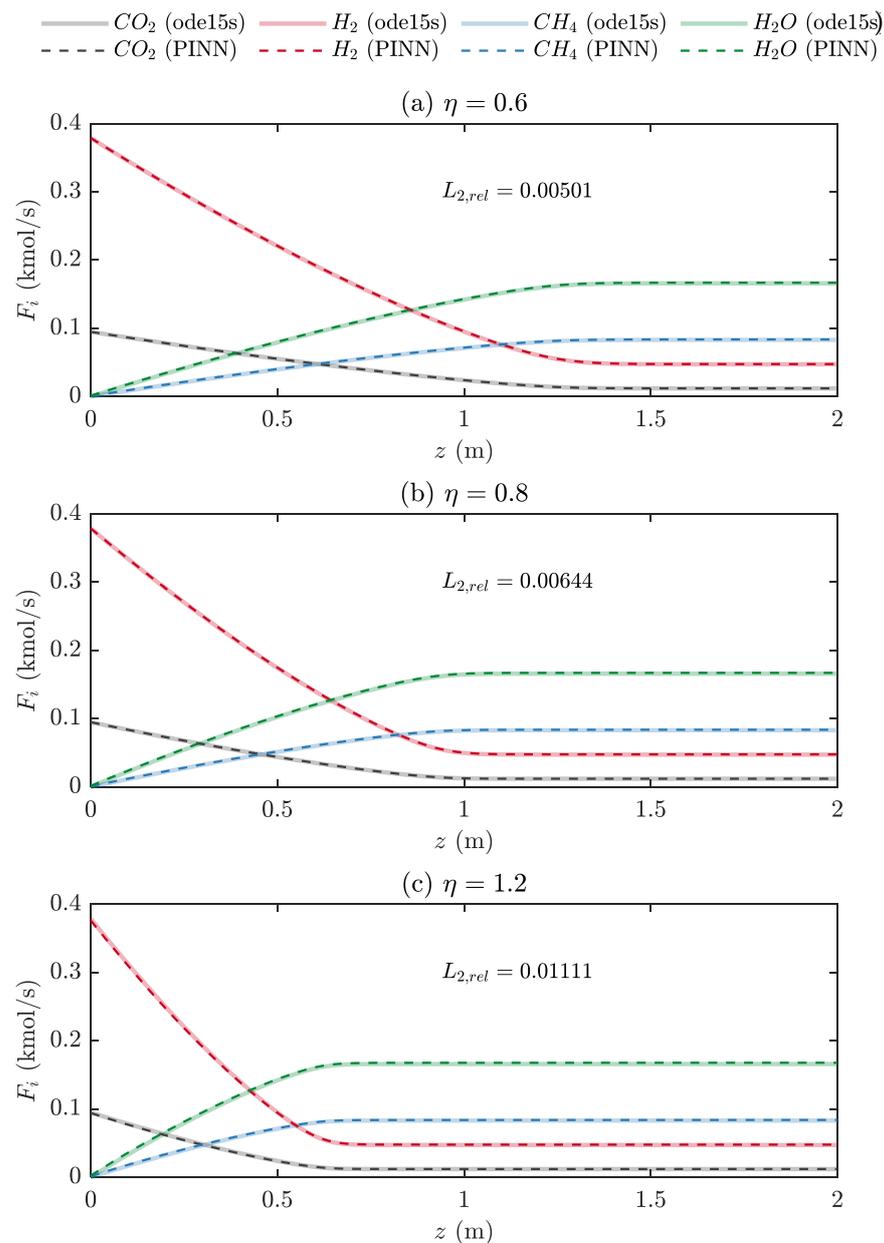


Figure 12. Performance of forward PINN models with two inputs (z and η) at $\eta = 0.6$ (a), 0.8 (b), and 1.2 (c).

5. Summary and Conclusions

A physics-informed neural network (PINN) was developed for an isothermal fixed-bed (IFB) reactor model for catalytic CO_2 methanation. The PINN was composed of a feed-forward artificial neural network (FF-ANN), automatic differentiation (AD) for derivatives, and governing equations with a stiff reaction kinetic rate. The loss function of the PINN included two mean squared errors (MSEs) for the governing equations and boundary conditions. The one-dimensional reactor was initialized at a molar flow rate that was the same as the boundary condition at the reactor inlet.

For the forward problem, the PINN solved the material balance expressed by ordinary differential equations (ODEs) for the IFB reactor model, where hyper-parameters, weights, and biases of FF-ANN were determined. For the inverse problem, the PINN used the weights and biases of the trained forward PINN model and identified unknown model parameters, such as the effectiveness factor of the CO_2 methanation reaction, using observation data. Future work is to implement the PINN for the solution of the fixed-bed reactor

model with a wide range of operating conditions (temperature, pressure, flow rate, and inlet composition), where the reactor model includes the heat and mass balances. The key conclusions drawn are as follows.

- The PINN with the *tanh* activation function, 5–7 hidden layers, and 256 neurons per hidden layer was found to have the most effective combination of hyper-parameters for the IFB reactor model.
- The reliability of the PINN depended on the number and range of the training data.
- When the molar flow rates of the reactor were predicted as out of the range of training data, the forward PINN model exhibited an excellent extrapolation performance because the PINN provides a solution satisfying physics-informed constraints.
- The inverse PINN model identified unknown model parameters when the observation data were properly provided to the inverse PINN.
- The training time of the forward PINN was almost proportional to the number of hidden layers and the number of training data points. The training time of the inverse PINN was relatively short, and the inverse PINN was more efficient at identifying unknown model parameters compared to other numerical methods such as computational fluid dynamics (CFD).
- The present PINN model was extended to different process conditions such as effectiveness factors.
- The current approach is useful for building a surrogate model for CO₂ methanation process design and optimization.

Author Contributions: Conceptualization, S.I.N. and Y.-I.L.; methodology, S.I.N. and Y.-I.L.; software, S.I.N.; validation, S.I.N. and Y.-I.L.; formal analysis, S.I.N.; investigation, S.I.N. and Y.-I.L.; resources, S.I.N. and Y.-I.L.; data curation, S.I.N.; writing—original draft preparation, S.I.N.; writing—review and editing, Y.-I.L.; visualization, S.I.N. and Y.-I.L.; supervision, Y.-I.L.; project administration, S.I.N. and Y.-I.L.; funding acquisition, S.I.N. and Y.-I.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (grant number: 2020R111A1A01074184). The APC was funded by NRF (grant number: 2020R111A1A01074184).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ngo, S.I.; Lim, Y.-I.; Lee, D.; Seo, M.W.; Kim, S. Experiment and numerical analysis of catalytic CO₂ methanation in bubbling fluidized bed reactor. *Energy Convers. Manag.* **2021**, *233*, 113863.
2. Rönsch, S.; Schneider, J.; Matthischke, S.; Schlüter, M.; Götz, M.; Lefebvre, J.; Prabhakaran, P.; Bajohr, S. Review on methanation—From fundamentals to current projects. *Fuel* **2016**, *166*, 276–296. [[CrossRef](#)]
3. Zantye, M.S.; Arora, A.; Hasan, M.M.F. Renewable-integrated flexible carbon capture: A synergistic path forward to clean energy future. *Energy Environ. Sci.* **2021**, *14*, 3986–4008. [[CrossRef](#)]
4. Miguel, C.V.; Mendes, A.; Madeira, L.M. Intrinsic kinetics of CO₂ methanation over an industrial nickel-based catalyst. *J. CO₂ Util.* **2018**, *25*, 128–136. [[CrossRef](#)]
5. Ngo, S.I.; Lim, Y.-I.; Lee, D.; Seo, M.W. Flow behavior and heat transfer in bubbling fluidized-bed with immersed heat exchange tubes for CO₂ methanation. *Powder Technol.* **2021**, *380*, 462–474. [[CrossRef](#)]
6. Kim, S.; Lim, Y.-I.; Lee, D.; Cho, W.; Seo, M.W.; Lee, J.G.; Ok, Y.S. Coal power plant equipped with CO₂ capture, utilization, and storage: Implications for carbon emissions and global surface temperature. *Energy Environ. Sci.* **2021**. in review.
7. Saeidi, S.; Najari, S.; Hessel, V.; Wilson, K.; Keil, F.J.; Concepción, P.; Suib, S.L.; Rodrigues, A.E. Recent advances in CO₂ hydrogenation to value-added products—Current challenges and future directions. *Prog. Energy Combust. Sci.* **2021**, *85*, 100905. [[CrossRef](#)]
8. Kim, S.; Lim, Y.-I.; Lee, D.; Seo, M.W.; Mun, T.-Y.; Lee, J.-G. Effects of flue gas recirculation on energy, exergy, environment, and economics in oxy-coal circulating fluidized-bed power plants with CO₂ capture. *Int. J. Energy Res.* **2021**, *45*, 5852–5865. [[CrossRef](#)]
9. Uebbing, J.; Rihko-Struckmann, L.K.; Sundmacher, K. Exergetic assessment of CO₂ methanation processes for the chemical storage of renewable energies. *Appl. Energy* **2019**, *233–234*, 271–282. [[CrossRef](#)]
10. Davis, C.R. *Methanation Plant Design for HTGR Process Heat*; OSTI: Oak Ridge, TN, USA, 1981; p. 218.
11. Lohmüller, R. Methanation Process. US Patent 4294932, 13 October 1981.

12. Porubova, J.; Bazbauers, G.; Markova, D. Modeling of the Adiabatic and Isothermal Methanation Process. *Environ. Clim. Technol.* **2012**, *6*, 79–84. [[CrossRef](#)]
13. Ngo, S.I.; Lim, Y.-I. Multiscale Eulerian CFD of chemical processes: A review. *ChemEngineering* **2020**, *4*, 23. [[CrossRef](#)]
14. Ngo, S.I.; Lim, Y.-I.; Lee, D.; Go, K.S.; Seo, M.W. Flow behaviors, reaction kinetics, and optimal design of fixed-and fluidized-beds for CO₂ methanation. *Fuel* **2020**, *275*, 117886. [[CrossRef](#)]
15. Kiewidt, L.; Thöming, J. Predicting optimal temperature profiles in single-stage fixed-bed reactors for CO₂-methanation. *Chem. Eng. Sci.* **2015**, *132*, 59–71. [[CrossRef](#)]
16. Alarcón, A.; Guilera, J.; Andreu, T. An insight into the heat-management for the CO₂ methanation based on free convection. *Fuel Process. Technol.* **2020**, *213*, 106666. [[CrossRef](#)]
17. Zimmermann, R.T.; Bremer, J.; Sundmacher, K. Optimal catalyst particle design for flexible fixed-bed CO₂ methanation reactors. *Chem. Eng. J.* **2020**, *387*, 123704. [[CrossRef](#)]
18. Vidal Vázquez, F.; Kihlman, J.; Mylvaganam, A.; Simell, P.; Koskinen-Soivi, M.L.; Alopaeus, V. Modeling of nickel-based hydrotalcite catalyst coated on heat exchanger reactors for CO₂ methanation. *Chem. Eng. J.* **2018**, *349*, 694–707. [[CrossRef](#)]
19. Hernandez Lalinde, J.A.; Roongruangsree, P.; Ilsemann, J.; Bäumer, M.; Kopyscinski, J. CO₂ methanation and reverse water gas shift reaction. Kinetic study based on in situ spatially-resolved measurements. *Chem. Eng. J.* **2020**, *390*, 124629. [[CrossRef](#)]
20. Kiewidt, L.; Thöming, J. Multiscale modeling of monolithic sponges as catalyst carrier for the methanation of carbon dioxide. *Chem. Eng. Sci. X* **2019**, *2*, 100016. [[CrossRef](#)]
21. Huynh, H.L.; Tucho, W.M.; Shen, Q.; Yu, Z. Bed packing configuration and hot-spot utilization for low-temperature CO₂ methanation on monolithic reactor. *Chem. Eng. J.* **2021**, *428*, 131106. [[CrossRef](#)]
22. Jia, C.; Dai, Y.; Yang, Y.; Chew, J.W. A fluidized-bed model for NiMgW-catalyzed CO₂ methanation. *Particuology* **2019**, *49*, 55–64. [[CrossRef](#)]
23. Lefebvre, J.; Bajohr, S.; Kolb, T. Modeling of the transient behavior of a slurry bubble column reactor for CO₂ methanation, and comparison with a tube bundle reactor. *Renew. Energy* **2020**, *151*, 118–136. [[CrossRef](#)]
24. Engelbrecht, N.; Chiuta, S.; Everson, R.C.; Neomagus, H.W.J.P.; Bessarabov, D.G. Experimentation and CFD modelling of a microchannel reactor for carbon dioxide methanation. *Chem. Eng. J.* **2017**, *313*, 847–857. [[CrossRef](#)]
25. Pérez, S.; Del Molino, E.; Barrio, V.L. Modeling and Testing of a Milli-Structured Reactor for Carbon Dioxide Methanation. *Int. J. Chem. React. Eng.* **2019**, *17*, 1–12. [[CrossRef](#)]
26. Güneş Baydin, A.; Pearlmutter, B.A.; Andreyevich Radul, A.; Mark Siskind, J. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2018**, *18*, 1–43.
27. Currie, R.; Fowler, M.W.; Simakov, D.S.A. Catalytic membrane reactor for CO₂ hydrogenation using renewable streams: Model-based feasibility analysis. *Chem. Eng. J.* **2019**, *372*, 1240–1252. [[CrossRef](#)]
28. Abiodun, O.I.; Jantan, A.; Omolara, A.E.; Dada, K.V.; Mohamed, N.A.; Arshad, H. State-of-the-art in artificial neural network applications: A survey. *Heliyon* **2018**, *4*, e00938. [[CrossRef](#)] [[PubMed](#)]
29. Gusmão, G.S.; Retnanto, A.P.; da Cunha, S.C.; Medford, A.J. Kinetics-Informed Neural Networks. *arXiv* **2020**, arXiv:2011.14473.
30. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*, 1st ed.; The MIT Press: Cambridge, MA, USA, 2012.
31. Ganti, H.; Kamin, M.; Khare, P. Design Space Exploration of Turbulent Multiphase Flows Using Machine Learning-Based Surrogate Model. *Energies* **2020**, *13*, 4565. [[CrossRef](#)]
32. Jin, X.; Cai, S.; Li, H.; Karniadakis, G.E. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **2021**, *426*, 109951. [[CrossRef](#)]
33. Neidinger, R.D. Introduction to Automatic Differentiation and MATLAB Object-Oriented Programming. *SIAM Rev.* **2010**, *52*, 545–563. [[CrossRef](#)]
34. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [[CrossRef](#)]
35. Jagtap, A.D.; Kharazmi, E.; Karniadakis, G.E. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* **2020**, *365*, 113028. [[CrossRef](#)]
36. Kim, J.; Lee, K.; Lee, D.; Jin, S.Y.; Park, N. DPM: A Novel Training Method for Physics-Informed Neural Networks in Extrapolation. *arXiv* **2020**, arXiv:2012.02681.
37. El Sibai, A.; Rihko Struckmann, L.K.; Sundmacher, K. Model-based Optimal Sabatier Reactor Design for Power-to-Gas Applications. *Energy Technol.* **2017**, *5*, 911–921. [[CrossRef](#)]
38. Schlereth, D.; Hinrichsen, O. A fixed-bed reactor modeling study on the methanation of CO₂. *Chem. Eng. Res. Des.* **2014**, *92*, 702–712. [[CrossRef](#)]
39. Koschany, F.; Schlereth, D.; Hinrichsen, O. On the kinetics of the methanation of carbon dioxide on coprecipitated NiAl(O)_x. *Appl. Catal. B* **2016**, *181*, 504–516. [[CrossRef](#)]
40. Kingma, D.P.; Lei Ba, J. Adam: A method for stochastic optimization. ICLR. *arXiv* **2015**, arXiv:1412.6980.
41. Rao, C.; Sun, H.; Liu, Y. Physics-informed deep learning for incompressible laminar flows. *Theor. Appl. Mech. Lett.* **2020**, *10*, 207–212. [[CrossRef](#)]
42. Ji, W.; Qiu, W.; Shi, Z.; Pan, S.; Deng, S. Stiff-PINN: Physics-informed neural network for stiff chemical kinetics. *arXiv* **2020**, arXiv:2011.04520.

43. Karimpouli, S.; Tahmasebi, P. Physics informed machine learning: Seismic wave equation. *Geosci. Front.* **2020**, *11*, 1993–2001. [[CrossRef](#)]
44. Xavier, G.; Yoshua, B. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 May 2010; pp. 249–256.
45. Salmon, J.K.; Moraes, M.A.; Dror, R.O.; Shaw, D.E. Parallel Random Numbers: As Easy as 1, 2, 3. In Proceedings of the SC'11: International Conference for High Performance Computing, Networking, Storage and Analysis, Seattle, WA, USA, 12–18 November 2011.
46. Bratley, P.; Fox, B.L. Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. *ACM Trans. Math. Softw. TOMS* **1988**, *14*, 88–100. [[CrossRef](#)]
47. Sun, L.; Gao, H.; Pan, S.; Wang, J.-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput. Methods Appl. Mech. Eng.* **2020**, *361*, 112732. [[CrossRef](#)]
48. Warey, A.; Han, T.; Kaushik, S. Investigation of Numerical Diffusion in Aerodynamic Flow Simulations with Physics Informed Neural Networks. *arXiv* **2021**, arXiv:2103.03115.