Physics-based neural networks for simulation and synthesis of cyclic adsorption processes

Sai Gokul Subraveti, Zukui Li, Vinay Prasad,* and Arvind Rajendran*

Department of Chemical and Materials Engineering, University of Alberta, 12th floor, Donadeo Innovation Centre for Engineering (ICE), 9211-116 Street, Edmonton, Alberta T6G1H9, Canada

E-mail: vprasad@ualberta.ca; arvind.rajendran@ualberta.ca

Abstract

A computationally faster and reliable modelling approach called a physics-based artificial neural network framework for adsorption and chromatography emulation (PANACHE) is developed. PANACHE uses deep neural networks for cycle synthesis and simulation of cyclic adsorption processes. The proposed approach focuses on learning the underlying governing partial differential equations in the form of a physics-constrained loss function to simulate adsorption processes accurately. The methodology developed herein does not require any system-specific inputs such as isotherm parameters. Accordingly, unique neural network models were built to fully predict the column dynamics of different constituent steps based on unique boundary conditions that are typically encountered in adsorption processes. The trained neural network model for each constituent step aims to predict the entire spatiotemporal solutions of different state variables by obeying the underlying physical laws. The proposed approach is tested by constructing and simulating four different vacuum swing adsorption cycles for post-combustion CO_2 capture without retraining the neural network models. For each cycle, 50 simulations, each corresponding to a unique set of operating conditions, are carried out until the cyclic-steady state. The results demonstrated that the purity and recovery calculated from the neural network-based simulations are within 2.5% of the detailed model's predictions. PANACHE reduced computational times by 100 times while maintaining similar accuracy of the detailed model simulations.

1 Introduction

Cyclic adsorption processes are particularly attractive for their flexibility in process configurations and hence, are extensively used in industrial gas separations such as hydrogen purification, oxygen enrichment, methane purification, carbon dioxide removal, etc.[1,2] In the most commonly used processes such as pressure swing adsorption (PSA), vacuum swing adsorption (VSA), temperature swing adsorption (TSA), etc., one or more fixed bed adsorption columns (with or without interactions) packed with a suitable adsorbent undergo a sequence of steps through a cyclic variation of pressure or temperature in order to perform the separation. Several process configurations (or cycles) can be synthesized by altering the sequence of steps or interactions between the adsorption columns. Given their transient and modular nature, adsorption processes are complex and operate at cyclic steady state (CSS). Rigorous mathematical models based on the underlying physical laws are required to better understand, simulate, and design such complex processes.[3,4] Process simulations involve solving these rigorous models characterized by a system of coupled nonlinear partial differential equations (PDEs) repeatedly in time and space until CSS. Further, each process configuration needs to be thoroughly optimized where thousands of simulations are carried out in order to identify the optimal set of design variables that yield the best process performance. Inherently, this makes the design and optimization of cyclic adsorption processes computationally expensive, thereby limiting the design to few process configurations for practical applications.[5,6] Despite the computational challenges, efforts have also been made to develop superstructure-based optimal process design frameworks.[7–9] With the recent discovery of hundreds of thousands of adsorbents for gas separations, the current simulation and optimization tools based on rigorous mathematical models are computationally inadequate to handle such huge databases of adsorbents for process design and optimization.[10]

To address the computational challenges posed by adsorption process design and optimization, the use of machine learning techniques such as artificial neural networks (ANNs) have emerged as alternatives to rigorous mathematical models.[11,12] To this end, Sant Anna et

al.[13] developed three-layer feed-forward ANN (input layer, one hidden layer and one output layer) models for the separation of methane and nitrogen using PSA. Using these models in the optimization, the authors show that the computational times significantly reduced from 15.7 h to 50 s. Subraveti et al.[14] constructed three-layer feed-forward ANN models initially within an optimization framework and subsequently used them to determine the Pareto solutions of multi-objective maximization of CO_2 purity and recovery for a complex eight-step PSA cycle designed for pre-combustion CO_2 capture. As a result, the relative error of Pareto solutions in both objectives was less than 1% and accelerated the optimization routine by ten times. Xiao et al.[15] instead used a multi-output feed-forward ANN architecture to predict process performances in the PSA optimizations. Pai et al.[16] extended the use of feed-forward ANN models to predict the axial profiles of the intensive variables for a four-step VSA process at CSS, and the models were experimentally validated. Furthermore, Oliveira et al.[17] developed a real-time soft sensor for a PSA unit based on neural network models. Three types of ANN architectures, namely, feed-forward, recurrent, and long short-term memory (LSTM) based on multi-input and a single output, are used to predict the PSA process performance over the number of cycles. The LSTM-based deep neural networks were found to be reliable for optimization, control and online measurements of PSA units. However, in these studies, ANN models were trained to learn the mapping between the inputs and the outputs for a fixed process configuration. This means that the models need to be retrained if the process configuration is changed. Therefore, a more generalized framework is required in order to facilitate the adsorption cycle synthesis and allow for the evaluation of several process configurations. For instance, Leperi et al.[18] used ANN to model individual steps in typical PSA processes for post-combustion CO_2 capture. Each step was modelled using 12 four-layer ANN (input layer, two hidden layers and multi-output layer) models. The trained ANN models predicted the five state variables, i.e. absolute pressure, CO_2 gas phase mole fraction, CO_2 molar loading, N_2 molar loading, and column temperature, at ten different locations across the column. In the ANN model training, the normalized mean squared error

between the neural network predictions and the training data from rigorous simulations was minimized. While this approach allowed synthesis of different PSA cycles, such models can require a large amount of training data to obtain accurate predictions.

The latest groundbreaking advances in artificial intelligence, machine, and deep learning have allowed for the development of viable methodologies to model various physical systems comprising governing PDEs.[19] To this end, ANNs with several hidden layers, also known as deep neural networks (DNNs), have shown an enhanced ability to capture very complex non-linear dynamics that led to tremendous interest for modelling physical systems, particularly the idea of learning PDEs through constraint-based loss functions.[20–22] For instance, Raissi *et al.*[22] developed a physics-informed neural network framework in which fully connected DNNs are capable of learning the PDE solutions anywhere on the spatiotemporal domain with few training points by incorporating a physics-constrained loss function. The presence of PDE-based terms in the loss function inherently facilitates the physics-constrained regularization of the neural networks. Owing to the additional knowledge of physics, the learning process for neural networks require only small amounts of training data, compared to the complexity of the systems described. The philosophy of physics constraining the loss function was later adopted for several applications.[23–25] However, most of these studies focused on demonstrating the capabilities of physics-based neural networks to learn a single PDE solution, i.e. for given initial and boundary conditions. If initial or boundary conditions change, then the physics-based neural networks have to be retrained. In cyclic adsorption processes, the initial condition of each step depends on the previous step's final condition that changes every cycle. This means that the PDE solutions differ from step to step and also every cycle. Therefore, a more generalized framework is required to implement the physics-based neural networks for cyclic adsorption systems.

In the present study, a modelling framework called physics-based artificial neural network framework for adsorption and chromatography emulation (PANACHE) is developed to synthesize and simulate different adsorption processes. By choosing the appropriate training

philosophy, the framework developed herein does not require any system-specific inputs such as isotherm parameters. Accordingly, unique neural networks models are trained for different constituent steps typically encountered in cyclic adsorption processes. The trained neural network model for each constituent step aims to predict the entire spatiotemporal solutions of state variables for an arbitrary initial gas composition profile and step-parameters which can be step operating conditions (such as step times and pressures) and inlet feed compositions, inlet feed velocities, etc. obeying underlying physical laws. The proposed methodology is tested by constructing and simulating four different VSA cycles for post-combustion CO_2 capture. The cycle synthesis capabilities of this approach are demonstrated by comparing the neural network- and the detailed process model-based simulations based on the four VSA cycles considered for a variety of operating conditions.

2 VSA governing equations

The one-dimensional mathematical model describes the adsorption column dynamics in VSA cycles after incorporating the following assumptions:

1. Axially dispersed plug flow model to describe the gas phase.

2. No radial gradients exist for composition and pressure across the column.

3. The gas-phase behaves ideally.

4. Uniform bed properties along the column.

5. The linear driving force model accounts for the solid-phase mass transfer.

6. Pressure drop calculations are based on Darcy's law (valid for the column sizes and the operating conditions considered here).

7. System operated under isothermal conditions.

Based on the above assumptions, the model comprises a system of coupled nonlinear PDEs based on the conservation of mass and momentum and takes the form:

$$\frac{\partial c_i}{\partial t} = \frac{\partial}{\partial z}\left[cD_{\mathrm{L}}\frac{\partial y_i}{\partial z} - c_i v\right] - \frac{1-\varepsilon}{\varepsilon}\frac{\partial q_i}{\partial t} \tag{1}$$

$$\frac{1}{P}\frac{\partial P}{\partial t} = -\frac{1}{P}\frac{\partial(Pv)}{\partial z} - \frac{RT_{\mathrm{ref}}}{P}\frac{1-\varepsilon}{\varepsilon}\sum_{i=1}^{n_{\mathrm{comp}}}\frac{\partial q_i}{\partial t} \tag{2}$$

$$-\frac{\partial P}{\partial z} = \frac{150}{4}\frac{1}{r_{\mathrm{p}}^2}\left(\frac{1-\varepsilon}{\varepsilon}\right)^2 \mu v \tag{3}$$

Here Eqs. 1 and 2 are component and overall mass balances for the gas phase, respectively. In Eq. 1, c_i, y_i, q_i are the gas-phase concentration, the gas-phase molar composition, and the solid-phase loading of the component i, v is the interstitial velocity, ε is the bed void fraction, and D_{L} is the axial dispersion coefficient. The ideal gas law relates c_i and y_i as follows: $c_i = \frac{y_i P}{RT_{\mathrm{ref}}}$, where P is the total pressure, R is the universal gas constant, and T_{ref} is the reference temperature. Using Eqs. 1 and 2, the gas-phase molar composition of the first component, y_1, and the the total pressure P are calculated, respectively. From the solution of y_1, the gas-phase molar composition of the second component can be obtained simply by: $y_2 = 1 - y_1$. Equation 3 represents the Darcy's law for calculating the pressure drop throughout the column, where r_{p} and μ are the particle radius and the gas-phase viscosity.

In addition to above equations, the linear driving force model describes the mass transfer in the solid phase:

$$\frac{\partial q_i}{\partial t} = k_i(q_i^* - q_i) \tag{4}$$

where q^* is the equilibrium loading and k is the mass transfer coefficient expressed, based on the assumption that the molecular diffusion in the macropores controls the transport into

the solid phase, as follows:

$$k_{\mathrm{i}} = \frac{c_{\mathrm{i}}}{q_{\mathrm{i}}^*} \frac{15\varepsilon_{\mathrm{p}} D_{\mathrm{p}}}{r_{\mathrm{p}}^2} \tag{5}$$

The adsorption equilibria was quantified using the competitive dual-site Langmuir (DSL) isotherm model as shown below:

$$q_i^* = \frac{q_{\mathrm{sb},i} b_i c_i}{1 + \sum_i b_i c_i} + \frac{q_{\mathrm{sd},i} d_i c_i}{1 + \sum_i d_i c_i} \tag{6}$$

where q_i^* is the equilibrium solid-phase loading of the component i, $q_{\mathrm{sb},i}$ and $q_{\mathrm{sd},i}$ are saturation capacities for the two sites and, b_i and d_i are the adsorption equilibrium constants. It is worth noting that the heat effects that are prominent in gas adsorption systems are deliberately not considered in this work for simplicity. In the future, heat effects will be accounted for to represent gas adsorption processes accurately.

3 PANACHE model

Physics-based neural networks adopt modern deep learning techniques to infer the underlying physical laws that involve PDEs.[22] The framework used for this study is illustrated in Fig. 1. Here the neural networks are trained to learn the conservation laws of mass and momentum along with adsorption equilibria. To this end, the residuals of PDEs are incorporated into the loss function so that the neural networks are trained to match the labelled data while penalizing them for violation of physical laws. Such a formulation allows the neural networks to learn the spatiotemporal solutions with small amounts of labelled data. It is worth noting that the labelled data represents the training data at the initial and the boundaries of the spatiotemporal domain obtained from the high fidelity simulations. Typically, the initial and the boundary conditions are required as inputs to solve the PDEs. However, in the proposed approach, the initial and the boundary data (together as labelled data) and the PDEs are

incorporated as loss terms in the loss function, minimized in the training procedure. The neural network model ultimately learns the weights and the biases.
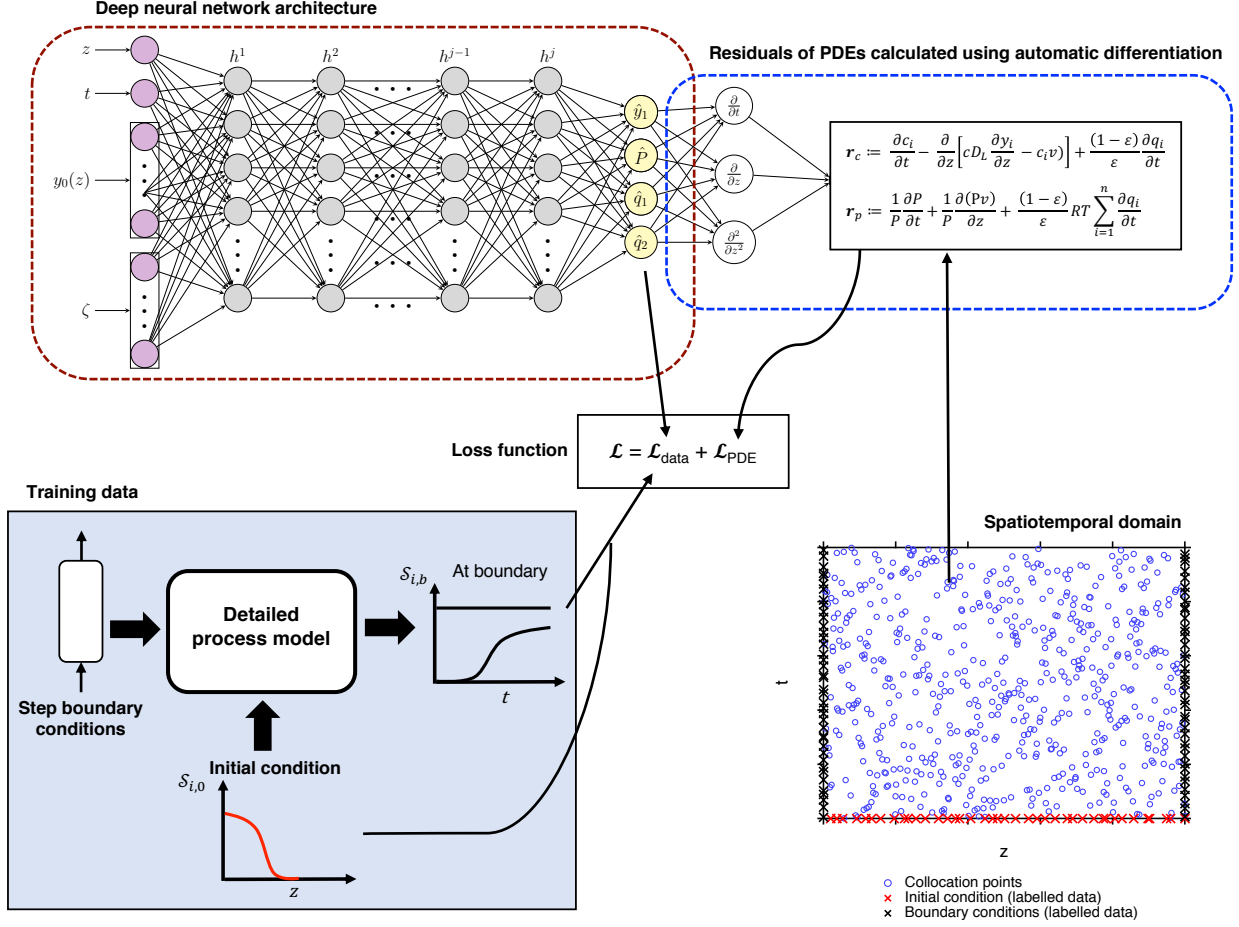

Figure 1: The physics-based neural network framework for adsorption and chromatography emulation (PANACHE) developed in the present study. *Top:* Physics-based deep neural network architecture. *Bottom left:* Training data generation using the detailed process model. *Bottom right:* Spatiotemporal domain with collocation points (blue), initial (red) and boundary (black) data.

The key idea of PANACHE is to recognize that P/V/T-SA processes comprise basic steps that can be identified by the position of the valves at the two ends of the column. The positions of valves at the two ends of adsorption columns in P/V/T-SA processes periodically change to implement different steps in the cycle. As a result, boundary conditions in each constituent step are different. Depending on the state (open or close) of the valves, con-

stituent steps can be categorized into generalized boundary conditions, such as open-open, open-closed, closed-open, etc. For instance, adsorption and purge steps are identified by open-open conditions; blowdown, evacuation, pressure equalization (donor) steps by closed (inlet) - open (outlet) conditions; and finally, pressurization and pressure equalization (receiver) steps by open (inlet) - closed (outlet) conditions. Hence, if each constituent step can be modelled by a suitable surrogate model, it should be, in principle, possible to synthesize cycles by combining them in a logical and physically feasible manner. However, in a cyclic process, the initial condition for each step depends on the final condition of the preceding step. This means that the initial conditions change based on the sequence and the duration of steps. To account for such variations in initial conditions, neural network models of each step type must be able to learn the solutions for an arbitrary initial condition. Another feature that affects the dynamics of adsorption columns is the step-parameters such as operating pressures, inlet gas conditions, etc. Here the neural network models are trained based on N_k different initial profiles and the step-parameters to have generalized capabilities. Although, in principle, different steps can be realized based on the same valve positions, the ultimate goal of the proposed modelling approach is to train the step neural network models that can take into account the dynamics of different possible steps for a given valve positions by providing appropriate boundary data in the model training.

One deep neural network is defined for each constituent step s in the VSA process, and the goal of the neural network is to learn the following mapping:

$$[z, t, y_0^s(z), \zeta^s] \xrightarrow{\boldsymbol{\theta^s}} [y_1^s(z, t), P^s(z, t), q_1^s(z, t), q_2^s(z, t)] \tag{7}$$

Here the neural network takes the following inputs: spatiotemporal coordinates (z, t), initial gas-phase molar composition profile, $y_0^s(z)$, of the first component, and step-parameters, ζ^s, for the step s. The neural network $f^s(z, t, y_0^s(z), \zeta^s, \boldsymbol{\theta^s})$ outputs the four state variables, namely, gas-phase composition of the first component, $y_1^s(z, t)$, the column total pressure,

$P^s(z, t)$, and solid loadings, $q_1^s(z, t)$ and $q_2^s(z, t)$. The symbol $\boldsymbol{\theta^s}$ represents the parameters of the neural network used for the individual step s. In other words, the neural network approximates the spatiotemporal solutions of four state variables of the step s for a given initial gas composition profile $y_0^s(z)$ and step-parameters ζ^s based on $\boldsymbol{\theta^s}$.

Based on the component and overall mass balances in Eqs. 1 and 2, the residuals can be defined as follows:

$$
\begin{aligned}
\boldsymbol{r_c}(z, t) &:= \frac{\partial c_1}{\partial t} + \frac{\partial}{\partial z}\left[c_1 v - cD_\mathrm{L}\frac{\partial y_1}{\partial z}\right] + \frac{1-\varepsilon}{\varepsilon}\frac{\partial q_1}{\partial t} \\
\boldsymbol{r_p}(z, t) &:= \frac{1}{P}\frac{\partial P}{\partial t} + \frac{1}{P}\frac{\partial (Pv)}{\partial z} + \frac{RT_0}{P}\frac{1-\varepsilon}{\varepsilon}\left[\frac{\partial q_1}{\partial t} + \frac{\partial q_2}{\partial t}\right]
\end{aligned}
\tag{8}
$$

and $v = \frac{4}{150\mu}\left(\frac{\varepsilon}{1-\varepsilon}\right)^2 r_\mathrm{p}^2\left(-\frac{\partial P}{\partial z}\right)$ is directly substituted in Eq. 8 to account for Darcy's pressure drop through the column. Equations 4-6 are not required to calculate $q_1(z, t)$ and $q_2(z, t)$. Instead, the neural networks are expected to learn solutions for $q_1(z, t)$ and $q_2(z, t)$ through constraints imposed on the right hand side of Eq. 8 while also matching the labelled data corresponding to q_1 and q_2. More details on the neural network architecture, the loss function, and the learning procedure are provided below.

3.1 Neural network architecture

Feed-forward deep neural networks are considered, comprising N_layers layers (one input layer, N_layers - 2 hidden layers and one output layer) with a predefined number of neurons. The neurons are interconnected to form a fully-connected complex network as shown in Fig. 1. The inputs to each neuron are combined with a set of coefficients called weights which can either dampen or amplify the input depending on its significance. In addition to weights, each neuron also has a bias. The inputs, weights and biases are combined in each neuron through a nonlinear activation function as shown below:

$$
X_l = \sigma_l(X_{l-1}W_l + b_l)
\tag{9}
$$

where X_{l-1} is the output of the $l-1$ layer; W_l and b_l represent the weight matrix and bias vector of the layer l, respectively and; X_l is the output of the l layer. The dimensions of weight matrix W_l and bias vector b_l are $N_{l-1} \times N_l$ and N_l, respectively, where N_{l-1} and N_l are the number of neurons in the $l-1$ and l layer, respectively. Collectively, the weight matrices ($\mathbf{W}$) and biases ($\mathbf{b}$) of the entire neural network are denoted using $\boldsymbol{\theta}$, i.e., ($\mathbf{W}$, $\mathbf{b}$) $\in \boldsymbol{\theta}$. In this way, each layer receives the outputs of a previous layer as inputs and feeds forward to the next layer. The number of hidden layers, N_{layers}, number of hidden neurons, N_l, and the activation functions (such as tanh, sine, sigmoid, etc.) are considered model hyperparameters and selected based on limited numerical experimentation.

3.2 Loss function

For enabling the mapping in Eq. 7, the hidden parameters, $\boldsymbol{\theta}$, need to determined by minimizing the loss function. The loss function constructed herein comprises two parts. Details of each part of the loss function are provided below:

3.2.1 Loss term: Labelled data

In this part of the loss function, the predictions of the neural network are constrained to match the labelled data. Here the labelled data refers to the training data obtained from the high fidelity simulations. The PDE solutions are unique to initial and boundary conditions of the each step s and proper enforcement of initial and boundary conditions is essential to have well-posed systems. Hence, the initial and the boundary data from the simulations are introduced as the labelled data. The loss term is expressed as the mean-squared error (MSE) between the neural network predictions and the labelled data. The loss term (for index $\#k$)

12

can be written as

$$\mathcal{L}_{data,0}^k = \frac{1}{N_{i,0}^k} \Bigg[\lambda_{0,y_1} \sum_{i=1}^{N_{i,0}^k} \Big[\hat{y}_1(z_i^k, 0, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - y_1(z_i^k, 0, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{0,P} \sum_{i=1}^{N_{i,0}^k} \Big[\hat{P}(z_i^k, 0, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - P(z_i^k, 0, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{0,q_1} \sum_{i=1}^{N_{i,0}^k} \Big[\hat{q}_1(z_i^k, 0, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - q_1(z_i^k, 0, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{0,q_2} \sum_{i=1}^{N_{i,0}^k} \Big[\hat{q}_2(z_i^k, 0, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - q_2(z_i^k, 0, y_0^k(z), \zeta^k) \Big]^2 \Bigg] \tag{10}$$

$$\mathcal{L}_{data,lb}^k = \frac{1}{N_{i,b}^k} \Bigg[\lambda_{lb,y_1} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{y}_1(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - y_1(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{lb,P} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{P}(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - P(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{lb,q_1} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{q}_1(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - q_1(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{lb,q_2} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{q}_2(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - q_2(z_{lb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 \Bigg] \tag{11}$$

$$\mathcal{L}_{data,rb}^k = \frac{1}{N_{i,b}^k} \Bigg[\lambda_{rb,y_1} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{y}_1(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - y_1(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{rb,P} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{P}(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - P(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{rb,q_1} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{q}_1(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - q_1(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 +$$

$$\lambda_{rb,q_2} \sum_{i=1}^{N_{i,b}^k} \Big[\hat{q}_2(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - q_2(z_{rb}^k, t_i^k, y_0^k(z), \zeta^k) \Big]^2 \Bigg] \tag{12}$$

Here $\mathcal{L}_{data,0}^k$, $\mathcal{L}_{data,lb}^k$, and $\mathcal{L}_{data,rb}^k$ represent the MSE on the initial, the left and the right boundary data, respectively. $\hat{y}_1$, $\hat{P}$, $\hat{q}_1$, and $\hat{q}_2$ are the neural network predictions whereas y_1, P, q_1, and q_2 denote the labelled data of four state variables. $N_{i,0}^k$ and $N_{i,b}^k$ represent the number of labelled initial and boundary data points. z_{lb}^k and z_{rb}^k are spatial coordinates on the left and the right boundary of the spatiotemporal domain. $k = 1, 2.., N_k$, represent different cases of initial profile and operating parameters. It is worth mentioning that all physical quantities in Eqs. 10-12 are in their non-dimensionalized form to scale to order $\sim \mathcal{O}(1)$ and the non-dimensionalization was carried out as outlined in Section 3.4. Finally, λ_0, λ_{lb}, and λ_{rb} are the weight terms ($[0, \infty]$). The choice of weights influence the constraints of matching the labelled data. Although there is no rule of thumb for choosing appropriate weights, limited numerical experimentation was carried out to estimate the weights that better fit both boundary and interior of spatiotemporal domain reasonably well. In addition to initial and boundary data, final column profiles of gas-phase composition are also provided in the training to improve prediction accuracies. The corresponding loss term, $\mathcal{L}_{data,f}^k$, for index #k can be written as

$$\mathcal{L}_{data,f}^k = \frac{\lambda_{f,y_1}}{N_{i,0}^k} \sum_{i=1}^{N_{i,0}^k} \left[\hat{y}_1(z_i^k, t_f^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) - y_1(z_i^k, t_f^k, y_0^k(z), \zeta^k) \right]^2 \tag{13}$$

Combining the four terms above leads to the first part of the loss function as shown below:

$$\mathcal{L}_{data}^k = \mathcal{L}_{data,0}^k + \mathcal{L}_{data,lb}^k + \mathcal{L}_{data,rb}^k + \mathcal{L}_{data,f}^k \tag{14}$$

3.2.2 Loss term: PDE residuals

The second part of the loss function introduces the physics regularization of the neural networks. The PDE residuals defined in Eq. 8 are incorporated here. For evaluating these residuals, N_r collocation points, i.e., (z_r, t_r), within the spatiotemporal domain are used as illustrated in Fig. 1. These collocation points are randomly chosen using Latin hypercube

sampling. Notably, this part of the loss function requires no additional labelled data from the simulations because the collocation points are just auxiliary points that help calculate the partial derivatives. The loss term for index $\#k$ expressed as the MSE is shown below:

$$\mathcal{L}_{residual}^k = \frac{1}{N_r^k} \left[\lambda_{r,c} \sum_{r=1}^{N_r^k} \boldsymbol{r}_c^k(z_r^k, t_r^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) + \lambda_{r,p} \sum_{r=1}^{N_r} \boldsymbol{r}_p^k(z_r^k, t_r^k, y_0^k(z), \zeta^k, \boldsymbol{\theta}^s) \right] \quad (15)$$

where N_r^k is the number of collocation points for case $\#k$. $\boldsymbol{r}_c^k$ and $\boldsymbol{r}_p^k$ correspond to the residuals of component and overall mass balances for case $\#k$, where $k = 1, 2.., N_k$. $\lambda_{r,c}$ and $\lambda_{r,p}$ are the weight terms.

3.2.3 Overall loss function

The overall MSE $\mathcal{L}$ defined by combining the loss terms from Eqs. 14 and 15 takes the following form:

$$\mathcal{L} = \sum_{k=1}^{N_k} \left[\mathcal{L}_{data}^k + \mathcal{L}_{residual}^k \right] \quad (16)$$

Here, individual terms $\mathcal{L}_{data}^k$ and $\mathcal{L}_{residual}^k$ obtained in each case k are summed over N_k cases of different initial profiles $y_0^s(z)$ and operating parameters, ζ^s, for step s. The idea here is to minimize $\mathcal{L}$ such that the neural network aims to learn the unique spatiotemporal solutions corresponding to different initial profiles and operating parameters. Hence, the constraints of matching the labelled data and reducing the PDE residuals close to zero are imposed for each case k and together minimize the MSE values obtained from all cases considered.

3.3 Training

The objective of training the deep neural networks is to determine the optimized weights and biases associated with each neuron in such a way that minimizes the loss function described in Eq. 16. Initially, the weights are specified using Xavier initialization[26] and the biases

are initialized based on random normal distribution. The training follows a backpropagation approach where the gradients of loss function with respect to each of the training weight computed based on the chain rule along with the learning rate are used to update the weights and the biases. This procedures continues to iterate until convergence is achieved. The loss function was minimized using Limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm, a quasi-Newton full-batch gradient-based optimization method. It is worth noting that the convergence to the global minimum with hundreds of thousands of adjustable parameters and the complex loss function may not be possible; however, studies show that this approach is capable of determining parameters for weights and biases that provide good prediction accuracies provided the hyperparameters are appropriately selected.[22]

Since the loss function requires computing the residuals of the PDEs, the spatial and temporal derivatives of outputs of the neural network are calculated using automatic differentiation.[27] Automatic differentiation uses chain rule to compute the partial derivatives of loss function through backpropagation of errors from the output layer to the input layer. The analytically defined connections between the layers of the deep neural network enable the implementation of this technique. In terms of accuracy, automatic differentiation offers higher accuracy compared to numerical differentiation as errors arising from truncation and rounding-off are avoided.[27] Automatic differentiation has been implemented in the deep learning frameworks such as Tensorflow[28] and PyTorch.[29] We implemented the training procedure using the deep learning library Tensorflow and the partial differential operators were computed using "tf.gradients()" in Tensorflow. The use of automatic differentiation on the neural network removes the need to have labelled data at the collocation points. This is an important feature of this approach. It is important to reiterate that the neural network model is trained for each step defined by a unique set of boundary conditions and the cycles are not used for the training. Given that AD has been implemented using in-built "tf.gradients" function in Tensorflow, quantifying the accuracy of AD implementation is not straightforward. Therefore, overall prediction accuracies of neural network models are considered for

16

ensuring appropriate implementation of differentiation routines.

3.4 Non-dimensionalization and normalization

The state variables, $y_1^s(z,t)$, $P^s(z,t)$, $q_1^s(z,t)$, and $q_2^s(z,t)$ have different orders of magnitude that can lead to difficulties while calculating backpropagated gradients during neural network training.[23,30] Hence, the physical quantities are non-dimensionalized in Eqs. 1 - 3 to have an appropriate scaling, i.e.$\sim \mathcal{O}(1)$. The non-dimensionalized quantities are defined as follows:

$$\bar{P} = \frac{P}{P_{\text{ref}}}, \quad \bar{c}_i = \frac{y_i \bar{P}}{P_{\text{ref}}} \quad \bar{q}_1 = \frac{q_1}{q_{\text{ref},1}}, \quad \bar{q}_2 = \frac{q_2}{q_{\text{ref},2}}, \quad \bar{v} = \frac{v}{v_{\text{feed}}} \tag{17}$$

Here P_{ref} is the reference pressure ($P_{\text{ref}} = 1$ atm), $q_{\text{ref},1}$ and $q_{\text{ref},2}$ are the equilibrium loadings of the first and the second component at feed conditions, and v_{feed} is the feed interstitial velocity (m s^{-1}). Since the gas-phase composition y is a molar fraction that lies in [0,1], this state variable is not non-dimensionalized.

The inputs are normalized to scale between [-1,1] to enhance the robustness of the neural network training[22,23,30] as follows:

$$\bar{z} = 2\frac{z}{L} - 1, \quad \bar{t} = 2\frac{t}{t_{\text{ref}}} - 1, \quad \bar{y}_0(z) = 2\frac{y_0(z) - min(y_0(z))}{max(y_0(z)) - min(y_0(z))} - 1 \tag{18}$$

where $\bar{z}$, $\bar{t}$, and $\bar{y}_0(z)$ are the normalized inputs to the neural network. L is the column length and t_{ref} is the maximum step duration. The operating parameters, ζ, are scaled such that the values lie in the range of [-1,1].

Based on the non-dimensionalized variables, Eqs. 1-3 can be rewritten as

$$\frac{\partial \bar{c}_i}{\partial \bar{t}} = \frac{\partial}{\partial \bar{z}}\left[-\psi_1 \bar{c}_i \bar{v} + \psi_2 \frac{\partial y_i}{\partial \bar{z}} \right] - \omega_i \frac{\partial \bar{q}_i}{\partial \bar{t}} \tag{19}$$

$$\frac{1}{\bar{P}}\frac{\partial \bar{P}}{\partial \bar{t}} = -\psi_1 \frac{1}{\bar{P}}\frac{\partial (\bar{P}\bar{v})}{\partial \bar{z}} - \sum_{i=1}^{n_{comp}} \omega_i \frac{\partial \bar{q}_i}{\partial \bar{t}} \tag{20}$$

$$-\frac{\partial \bar{P}}{\partial \bar{z}} = \psi_3 \bar{v} \tag{21}$$

The dimensionless groups in the above equations are given by

$$\psi_1 = \frac{v_{\text{feed}} t_{\text{ref}}}{L}, \quad \psi_2 = \frac{D_L t_{\text{ref}}}{L^2}, \quad \psi_3 = \frac{150}{4} \frac{1}{r_{\text{p}}^2} \left(\frac{1-\varepsilon}{\varepsilon}\right)^2 \frac{\mu v_{\text{feed}} L}{P_{\text{ref}}}, \quad \omega_i = \frac{RT_{\text{ref}} q_{\text{ref,i}}}{P_{\text{ref}}} \frac{1-\varepsilon}{\varepsilon}$$

The non-dimensionalization and the normalization ensures that all the variables and inputs are scaled to order $\mathcal{O}(1)$. Prior to training, the labelled data provided to the neural network are scaled based on Eqs. 17-18 and the residuals are defined based on Eqs. 19-21. The predicted quantities are finally reverted to their original form at the end of the simulation.

3.5 Detailed model simulations

The VSA cycles are simulated using our detailed one-dimensional mathematical model.[3] The system of coupled nonlinear PDEs are numerically solved by discretizing the spatial terms into 50 finite volumes using the total variation diminishing (TVD) scheme with van-Leer flux limiter. The resulting ordinary differential equations (ODEs) are integrated in time based on *ode23s* solver in MATLAB. Individual steps are simulated by imposing appropriate boundary conditions.[3] All cycle simulations are carried out using standard unibed approach, i.e., a single column undergoes all steps in the cycle sequentially, until it reaches CSS. The CSS was considered to be achieved when the mass balance error for the entire cycle equals to 0.5% or less in the five consecutive cycles. The simulations provide detailed composition, pressure and temperature spatiotemporal profiles from the initial cycle to CSS which are then used to calculate process performance indicators. It is worth noting that the detailed simulations are validated elsewhere against both lab-scale[31] and pilot-scale experiments.[32]
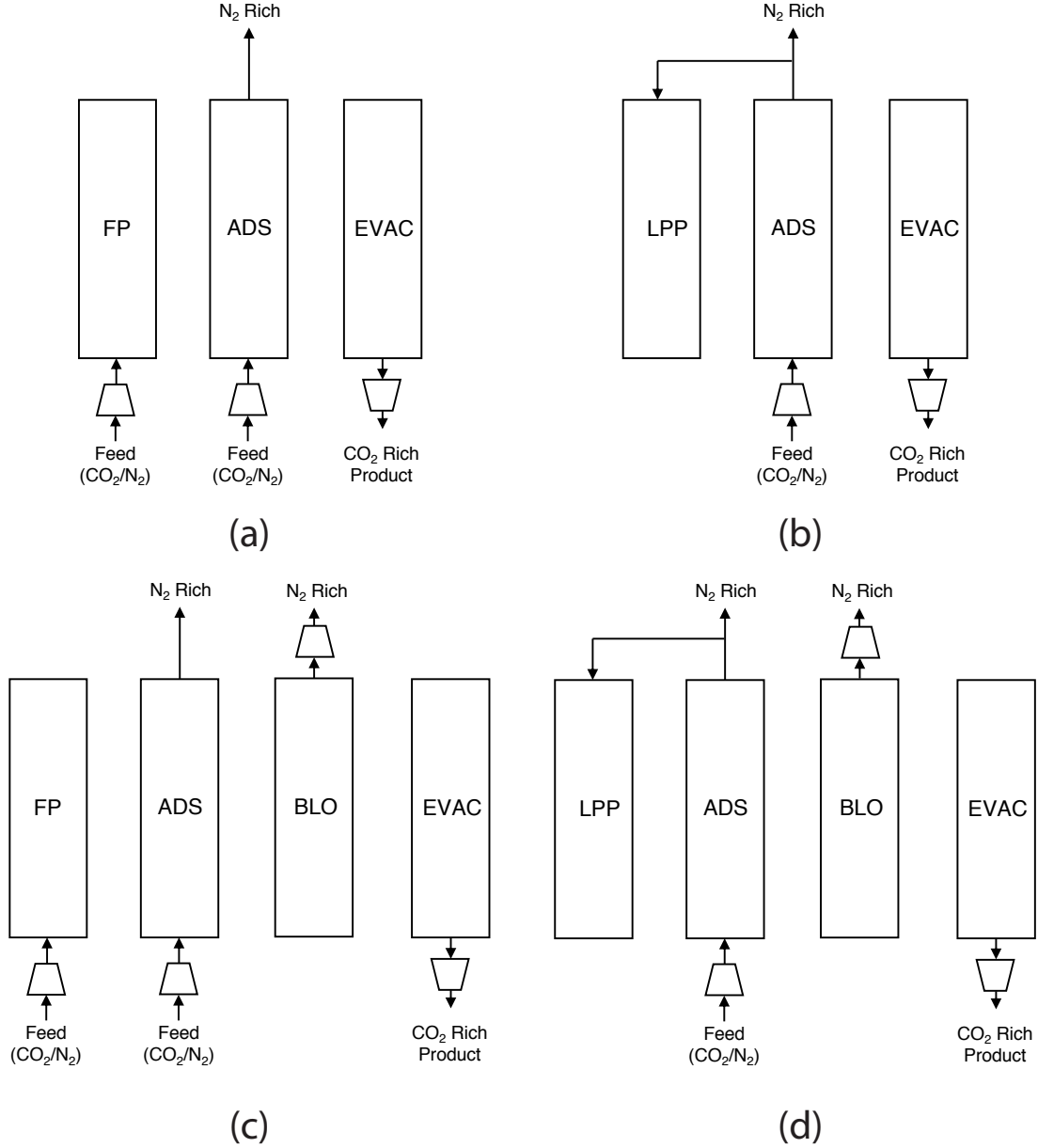
Figure 2: Four different VSA cycles considered for demonstrating the cycle synthesis capabilities of physics-based neural networks.

4 Results and discussion

Four simple VSA cycles, illustrated in Fig. 2, are considered to demonstrate the ability of the proposed methodology to synthesize and simulate different cyclic adsorption processes. The feed consists of a binary mixture of 20 mol% CO_2 and 80 mol% N_2 at 1 bar and

25 °C. The constituent steps used to construct these cycles can be categorized into four different step types: the adsorption step, the blowdown step, the evacuation step and the pressurization step. First, a simple three-step VSA cycle, shown in Fig. 2(a), consists of the following steps is considered: 1) In the adsorption step (ADS), the feed mixture introduced in the column with constant interstitial velocity (v_{feed}) at $P_{\text{H}} = 1$ bar undergoes separation through preferential adsorption of the heavy component CO_2. On the other hand, the light component N_2 leaves the column. 2) In the evacuation step (EVAC), the column pressure is reduced to a low pressure (P_{L}) in the counter-current direction using a vacuum pump with constant interstitial velocity (v_{EVAC}) at the boundary, similar to realistic conditions,[33,34] to collect the CO_2 rich product at the feed end of the column. 3) The feed pressurization step (FP) pressurizes the column to 1 bar using the feed mixture through a blower with constant interstitial velocity at the boundary, v_{FP}. The second cycle in Fig. 2(b) is a variant of the first cycle. Instead of using the fresh feed for pressurizing the column in the pressurization step, the light product from the adsorption step is used to pressurize the column from the light product end. In the third and the fourth cycles, the blowdown step is included after the adsorption step in the first and the second cycles to depressurize the column to an intermediate pressure P_{I} in the co-current direction through the light product end to remove N_2 from the column. The vacuum pump is used to remove the gas by implementing a constant interstitial velocity (v_{BLO}) boundary condition at the light product end. Although the underlying constituent steps are the same in these cycles, in principle, these cycles can yield different performances and are considered unique from a process design perspective. The VSA cycle performs the separation with IISERP MOF2, a novel metal-organic framework, as an adsorbent. Previous screening studies have shown superior performance of IISERP MOF2 for post-combustion CO_2 capture.[34,35] The CO_2 and N_2 isotherms on IISERP MOF2 at 25 °C are shown in Fig. 3 and the DSL isotherm parameters are reported in Table S1 in the Supporting Information. Each step in the VSA cycle can be distinguished based on boundary conditions provided in Table 1. Hence, separate

neural network models are developed for each step to predict its spatiotemporal dynamics. As can be seen from Table 1, the durations of blowdown, evacuation, and pressurization step can be calculated through implementation of a constant velocity boundary condition based on the pressures P_H, P_I, and P_L.
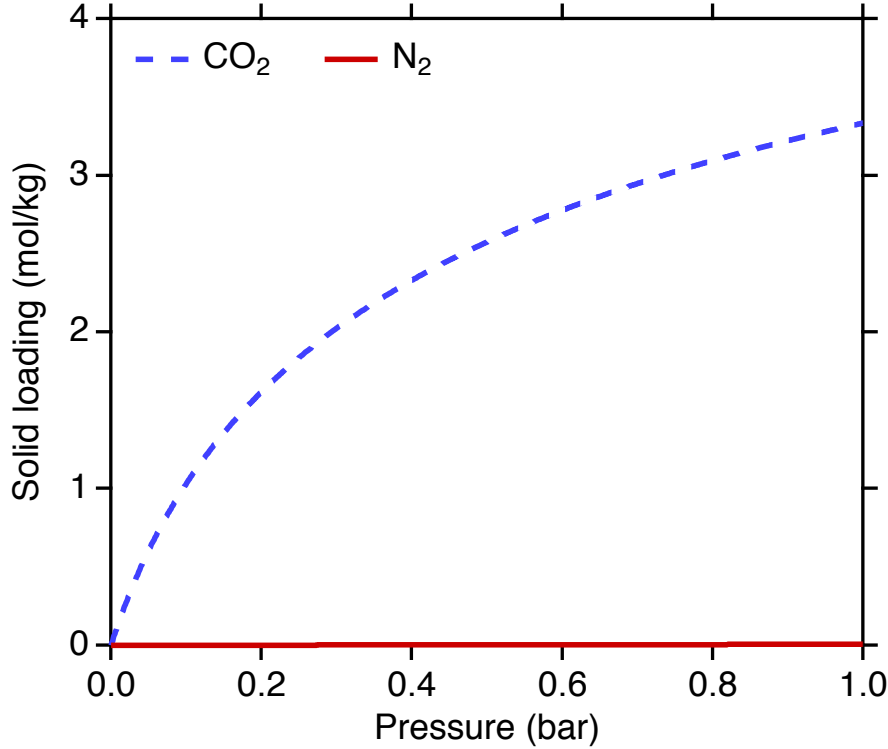

Figure 3: CO_2 and N_2 isotherms on IISERP MOF2 at 298.15K.[35]

The VSA cycle performance can be tuned based on the following operating parameters. For the adsorption step, feed velocity (v_{feed}) and duration of the adsorption step (t_{ADS}) can be varied. In blowdown and evacuation steps, pressures P_I, P_L, and vacuum pump velocities, v_{BLO}, v_{EVAC} are the variables. The pressurization inlet velocity, v_{FP}, is an operating parameter in the pressurization step. In the present study, v_{feed}, v_{BLO}, v_{EVAC}, and v_{FP} are held constant and are not considered operating parameters in neural network training. In other words, the VSA cycle simulations are restricted to fixed vacuum pump and blower sizes.

Table 1: Boundary conditions and the related parameters (ζ) for different steps considered in this study.

Step	z=0	z=L	ζ
Adsorption	$v\big|_{z=0} = v_{\text{feed}}$	$P\big|_{z=L} = P_{\text{ADS}}$	-
	$D_{\text{L}} \frac{\partial y_i}{\partial z}\big|_{z=0} = -v\big|_{z=0}\left(y_{i,\text{feed}} - y_i\big|_{z=0}\right)$	$\frac{\partial y_i}{\partial z}\big|_{z=L} = 0$	
Blowdown	$\frac{\partial P}{\partial z}\big|_{z=0} = 0$	$v\big|_{z=L} = v_{\text{BLO}}$	-
	$\frac{\partial y_i}{\partial z}\big|_{z=0} = 0$	$\frac{\partial y_i}{\partial z}\big|_{z=L} = 0$	
Evacuation	$v\big|_{z=0} = v_{\text{EVAC}}$	$\frac{\partial P}{\partial z}\big|_{z=L} = 0$	P_0
	$\frac{\partial y_i}{\partial z}\big|_{z=0} = 0$	$\frac{\partial y_i}{\partial z}\big|_{z=L} = 0$	
Pressurisation	$v\big|_{z=0} = v_{\text{FP}}$	$\frac{\partial P}{\partial z}\big|_{z=L} = 0$	P_0
	$D_{\text{L}} \frac{\partial y_i}{\partial z}\big|_{z=0} = -v\big|_{z=L}\left(y_{i,\text{feed}} - y_i\big|_{z=L}\right)$	$\frac{\partial y_i}{\partial z}\big|_{z=L} = 0$	y_{in}

4.1 Neural network training

4.1.1 Neural network architecture

Unique neural networks are trained to learn the spatiotemporal dynamics of four different constituent steps: the pressurization step, the adsorption step, the co-current blowdown step, and the counter-current evacuation step. The neural network architecture employed for all steps consists of an input layer, ten hidden layers with 100 neurons each, and one output layer with four neurons (four state variables). The choice of specified number of hidden layers and neurons was based on limited numerical experimentation that was carried out where no significant influence on the neural network training performance was observed when the

number of hidden layers were varied between 8 to 12 layers and the number of neurons were varied from 80 to 120. The number of neurons in the input layer varies from step to step. The common inputs for all steps are the spatiotemporal coordinates (z, t), and the initial gas composition at 50 spatial locations (based on the finite volume discretization of the detailed model). In addition to these 52 inputs, other inputs related to the step-parameters are included depending on the step. For this study, the only variable in the adsorption step is the step duration t_{ADS}. Since time t is already one of the inputs to the neural networks, the variation of t_{ADS} can be achieved by extracting different slices of spatiotemporal solutions of the four state variables. Therefore, t_{ADS} is explicitly not considered as a step-parameter input. As the blowdown step occurs after the adsorption step, the column pressure at the $z = 1$ boundary is always at 1 bar initially. Subsequently, it reduces to the final column pressure, P_{I}, when subjected to a constant vacuum pump flow. Similar to the adsorption step, slices of spatiotemporal solutions can be extracted for any P_{I}. Again, there is no need for an additional step-parameter. In the evacuation step, the initial column pressure, P_0, will be either P_{H} or P_{I} and then depressurizes to P_{L}. Here, initial column pressure, P_0, depends on the preceding step and the spatiotemporal solutions of evacuation step are dependent on P_0. Hence, P_0 is considered as an input to the evacuation step model. Similarly, the initial column pressure, P_0, in the pressurization step is also an input for the pressurization step model. Moreover, the inlet gas-phase composition $(y_{\text{in,P}})$ in the pressurization step can be a variable depending on the stream (i.e. feed or light product) used for pressurizing the column. This makes $y_{\text{in,P}}$ a parameter for the pressurization step. Hence, P_0 and $y_{\text{in,P}}$ are the additional inputs for the pressurization step. To summarize, the adsorption and the blowdown step neural networks have 52 inputs, the evacuation step neural neural network has 53 inputs, and the pressurization step input has 54 inputs. Since the proposed approach requires the calculation of first and second order derivatives using automatic differentiation, the nonlinear activation in the neural networks must be differentiable. Hence, a hyperbolic tangent function is used for the nonlinear activation in the adsorption and sinusoidal function

for other steps. It is worth noting that the neural network hyperparameters such as the number of hidden layers, the number of hidden neurons, activation function, and weight terms ($\lambda_{data,0}$, $\lambda_{data,lb}$, $\lambda_{data,rb}$, $\lambda_{data,f}$, $\lambda_{r,c}$, and $\lambda_{r,p}$) are chosen based on limited numerical experimentation.

4.1.2 Generating initial profiles for training

To gather training data for each model, individual steps must be simulated at various initial column profiles and step-parameters to enable the mapping in Eq. 7. As previously mentioned, the column profiles change drastically depending on the sequence of steps implemented, step-parameters, and the number of times the cycle is simulated until CSS. To ensure appropriate sampling of different initial column profiles for training, knowledge of various column profiles encountered in cyclic adsorption processes is essential. One way to learn about the types of column profiles typically encountered is to simulate various adsorption processes using the detailed model and gather different column profiles of individual steps for various operating conditions. The other approach involves synthetically generating different types of column profiles based on mathematical functions, splines, etc. The drawback of the latter approach is the loss of column profile characteristics specific to adsorption processes. Here, the former approach is used to gather different initial profiles by first simulating different VSA cycles using the detailed model based on the simulation parameters provided in Table S2 in the Supporting Information. The detailed model-based cycle simulations are carried out at different operating conditions generated randomly using Latin hypercube sampling. For each set of operating conditions, VSA cycle simulations are carried out until the CSS. At the same time, the gas-phase composition column profiles after every step are gathered from the initial cycle to the CSS. From these, N_k=60 different initial column profiles and the corresponding step-parameters of each step are randomly chosen and used for training the model. Note that N_k=60 was determined as an appropriate number of initial column profiles based on a numerical experimentation reported in the supporting information. Moreover,

while selecting the initial gas-phase composition profiles for training, it was ensured that initial column profiles are unique and diverse by gathering them from a well-sampled space that corresponds to different cycles, operating conditions, cycle number before the system reaches CSS, and also the extent to which the fronts propagated into the column. Figure S2 in the supporting information illustrates different initial profiles used in the neural network training for each step.

4.1.3 Data acquisition and learning procedure

Next, individual steps are separately simulated based on these initial profiles and step-parameters to obtain the labelled data for model training. The temporal domains defined for the individual steps in the step simulations are larger than the typical operating ranges used in the cycles. For instance, the blowdown and the evacuation steps are simulated for durations where the column pressures reach lower than the desired pressures in the cycles. The rationale behind defining such large temporal domains in the individual step simulations is the flexibility to extract different slices of temporal solutions within the larger domain, when using a trained neural network. In essence, this approach allows us to use the neural network model for predictions with all meaningful step times. It is worth noting that a unique simulation is carried out for each set of initial profiles and step-parameters. In the data acquisition procedure, the simulation data from each case is first gathered, and then the inputs and the physical quantities are subjected to normalization and non-dimensionalization, respectively. Following this, the initial and the boundary data of four state variables based on N_k cases are gathered for each step to form the labelled data. Although the data generated in simulations is massive, only a tiny portion is extracted as the labelled data. For inferring the PDE solution within the spatiotemporal domain, $N_r = 250$ collocation points are randomly generated using Latin hypercube sampling for each initial profile and step-parameter. It is worth reiterating that no additional simulation data is required for the collocation points.

The learning procedure is implemented in Tensorflow v1.15[28] and the computations are performed based on a single Quadro P5000 GPU card where it took approximately 4-8 hours training each model. It is worth reiterating that no system-specific inputs such as mass transfer coefficients, CO_2 and N_2 isotherm parameters are fed to the neural networks in the learning process.

4.1.4 Comparison of predicted and detailed model solutions

The effectiveness of the trained neural network models in learning the spatiotemporal solutions of VSA governing equations can be visualized in Fig. 4, where the spatiotemporal solutions based on both neural network predictions and detailed model simulations are compared for one of the test cases in the previous section. For the discussion, the blowdown step is shown as an example. Here, the emphasis remains on whether the neural network model has accurately learnt the interior of the spatiotemporal solutions of the four state variables. As can be seen from the figure, the results produced by the physics-based neural networks are in very good agreement with the detailed model solutions for all four state variables and the relative $\mathcal{L}_2$ error calculated for the four state variables are as follows: $y_1(z,t)$: 4.3e-02, $P(z,t)$: 2.9e-03, $q_1(z,t)$: 2.9e-02, and $q_2(z,t)$: 1.7e-02. Remarkably, this also means that the neural networks accurately learnt the underlying interdependencies of each state variable in the VSA process. The prediction accuracies indicate that the methodology employed herein can successfully enable the desired coupling of the state variables by simultaneously minimizing the residuals of component and overall mass balances along with labelled data. It is worth reiterating that the use of physics-based residuals in the loss function has allowed the neural networks to learn the adsorption column dynamics with one fully connected multi-output architecture instead of conventional approaches of having multiple surrogate models for each state variable. Another interesting feature is the ability to predict q_1 and q_2 solutions. The detailed model calculations of q_1 and q_2 require the adsorption isotherm and the linear driving force model. In the present methodology, instead of explicitly providing

26

these equations, the initial, final and boundary data of q_1 and q_2 are provided. By simultaneously allowing the neural networks to match this data along with minimizing the overall PDE residuals, the q_1 and q_2 terms in the PDEs are forced to obey the imposed constraints, thereby predicting the interior solutions of q_1 and q_2 reasonably well.

4.2 Cycle synthesis and simulations

Here, the effectiveness of the proposed approach is tested for cycle synthesis by constructing and simulating the four VSA cycles illustrated in Fig. 2. For this, the cycle simulations are carried out by varying t_{ADS}, P_{I}, and P_{L} between 5-50 seconds, 0.7-0.99 bar and 0.18-0.23 bar, respectively. Both neural networks and the detailed process model are initialized with an arbitrary initial condition and each of the constituent steps are simulated sequentially. The final state of the column in a step is set as the initial condition for the subsequent one. The simulations are carried out until CSS and the results produced by the two approaches are compared at CSS. In detailed process simulations, the CSS condition is reached when the overall mass balance error equals 0.5% or less for five consecutive cycles. The CSS was assumed to be achieved in neural network-based simulations when there is no variation in the column dynamics of each step in the cycle during repeated cycling. It is worth noting that the overall mass balance error in neural network simulations may not necessarily converge to less than 0.5% due to slight discrepancies in the neural network predictions compared to the original solutions (see Section 4.3 for the detailed discussion).

4.2.1 Simulation of three-step FP cycle

First, a simple three-step VSA cycle with feed pressurization (FP) is constructed based on the individual feed pressurization, adsorption, and evacuation step models. A specific process operating condition chosen randomly: $t_{\text{ADS}} = 25$ s, P_{L}=0.2 bar is considered to test the ability of neural network-based simulations to predict CSS profiles. Figures 5 and 6 illustrate the column profiles of four state variables after cycle #1 and at CSS from neural
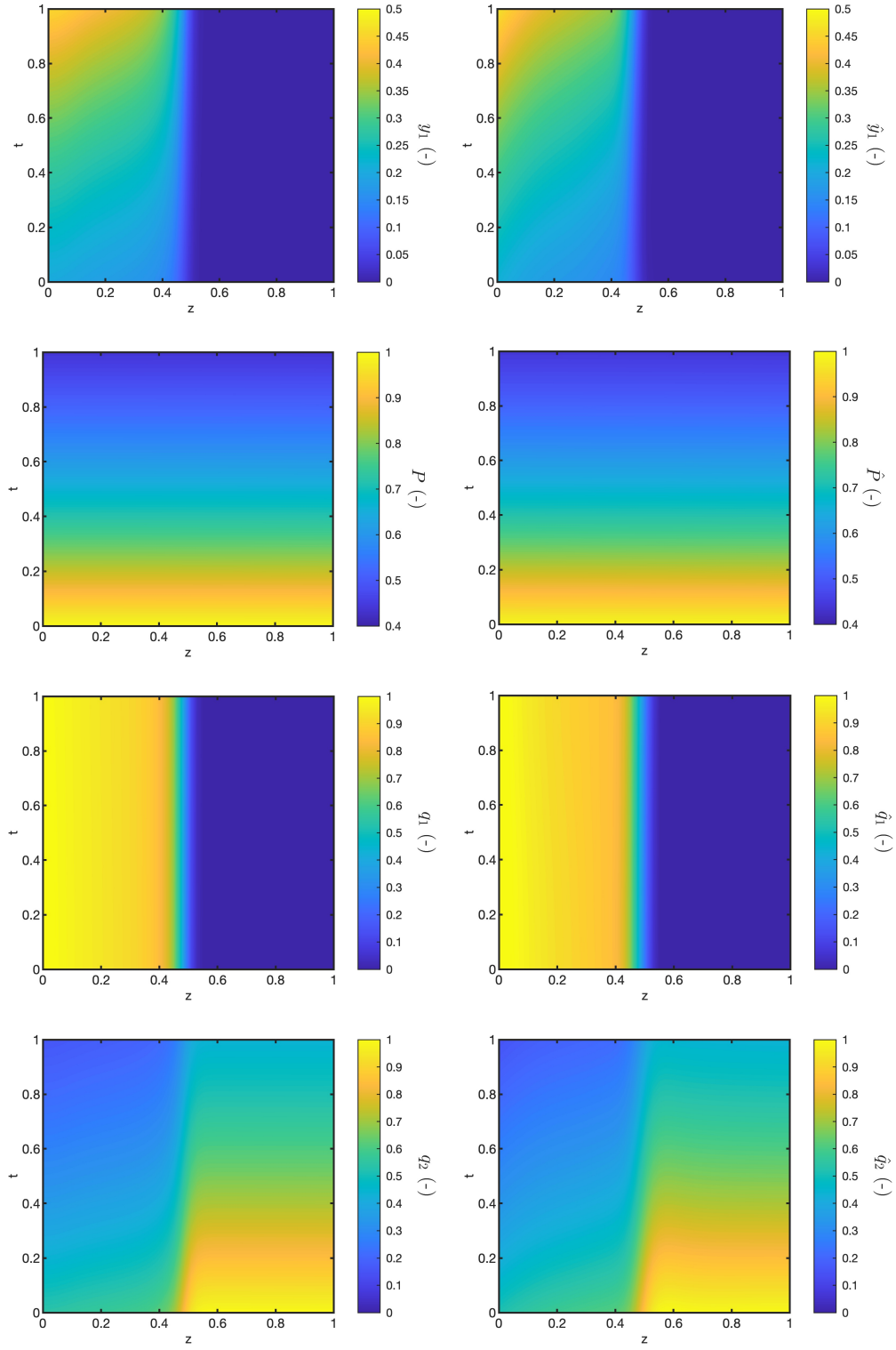
Figure 4: Comparison of spatiotemporal solutions of the four state variables based on the detailed process model (left hand panel) and the blowdown step neural network predictions (right hand panel) for one of the test initial column profile. In both the simulations, the blowdown step is initialized with a column pressure of 1 bar.

network- and detailed process model-based simulations, respectively. Overall, the neural network predictions are accurate. However, the neural network model predicted CO_2 gas-phase compositions and CO_2 solid loadings at the end of the cycle #1 marginally advanced further through the column. Nevertheless, this discrepancy vanished at CSS where the profiles calculated from both the simulations are in good agreement. In Table 2, the CO_2 purity and CO_2 recovery calculated based on both neural network and detailed process simulations are reported. Moreover, the computational times are also provided. The neural network-based calculations of CO_2 purity and CO_2 recovery have an error deviation of 3% and the neural networks simulations are almost 31 times faster than the detailed model simulations.
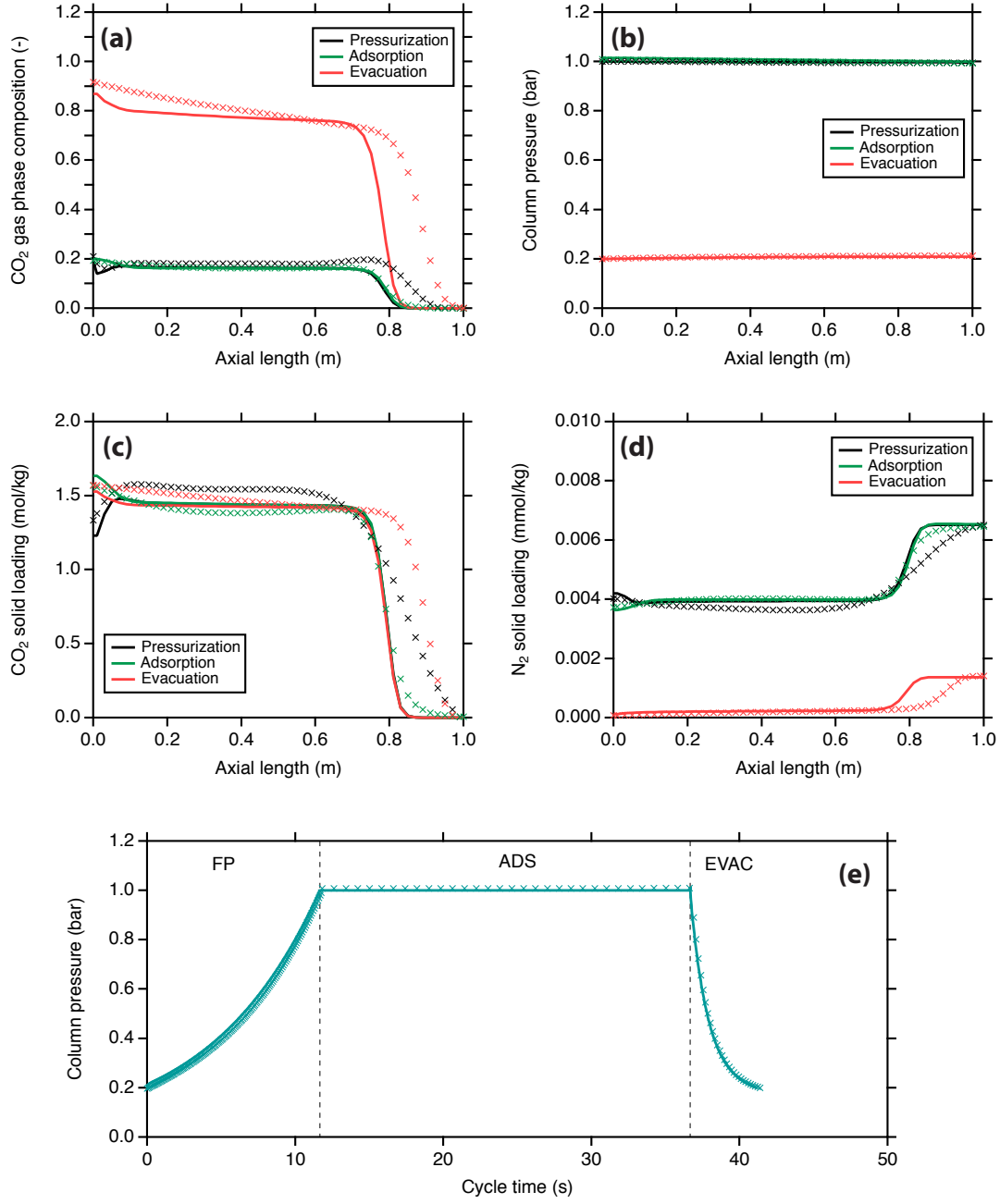
Figure 5: Comparison of cycle #1 profiles of four state variables obtained from the neural network (markers) and the detailed model (lines) simulations of three-step FP cycle for the operating condition: t_{ADS}=25 s and $P_L = 0.2$ bar.
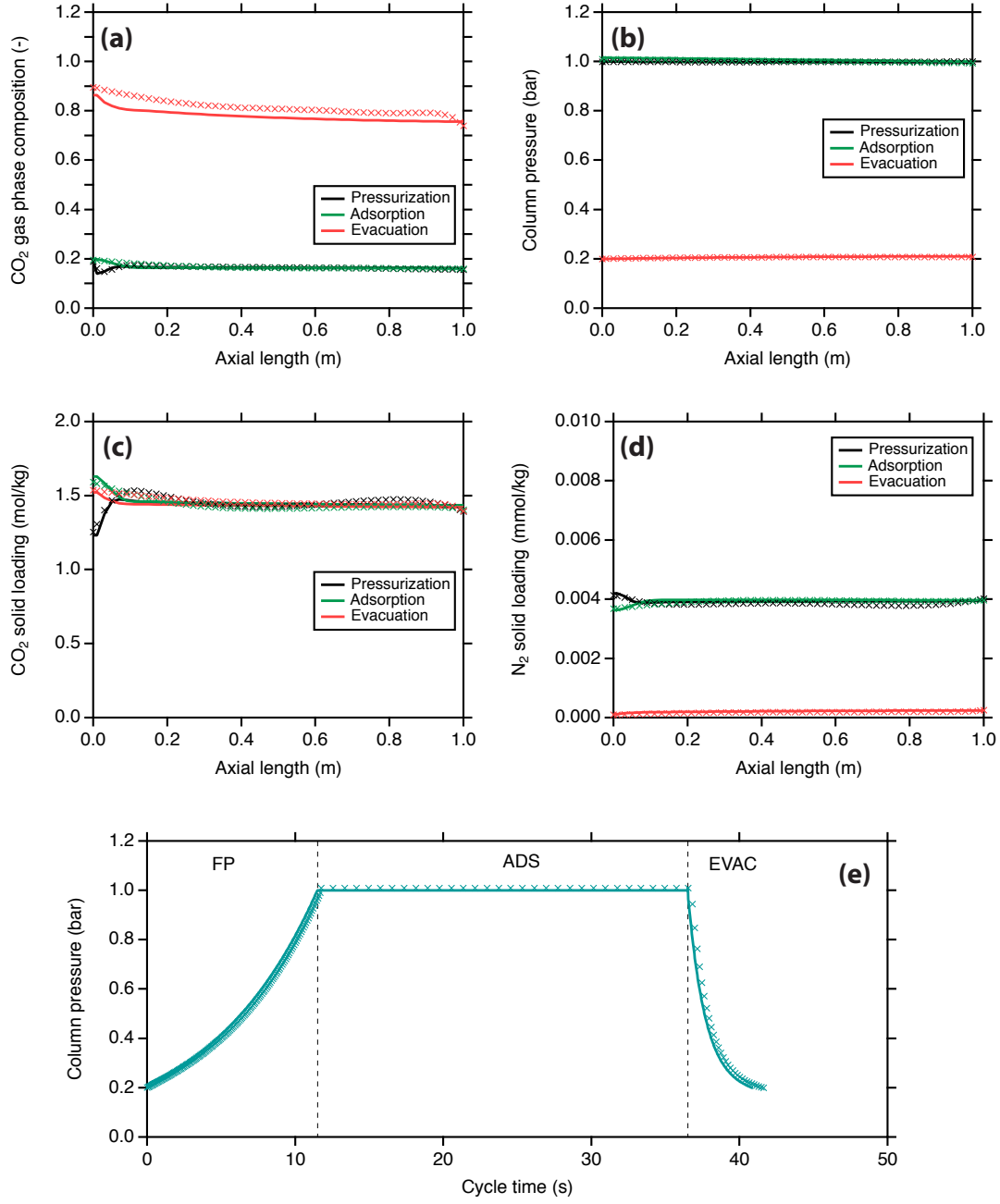
Figure 6: Comparison of cyclic steady state (CSS) profiles of four state variables obtained from the neural network (markers) and the detailed model (lines) simulations of three-step FP cycle for the operating condition: t_{ADS}=25 s and $P_{\text{L}} = 0.2$ bar.

Table 2: Summary of CO_2 purity and CO_2 recovery calculated based on neural networks (NN) and detailed process model (DM) simulations for the operating condition: $t_{ADS}=25$ s and $P_L = 0.2$ bar. For four-step cycles, $P_I = 0.75$ bar was used. The number of cycles to reach CSS and the computational time are also reported. Note that both the neural network and the detailed process model simulations were carried out on a CPU and the CPU seconds reported were on a 128 GB and 3.10 GHz workstation.

VSA cycles	CO_2 purity (%)		CO_2 recovery (%)		Computational time (s)	
	DM	NN	DM	NN	DM	NN
Three-step FP cycle	46.1	48.9	29.0	32.5	231.5	7.5
Three-step LPP cycle	52.1	50.6	40.2	37.9	330.2	17.9
Four-step FP cycle	53.2	55.5	26.2	29.3	247.8	7.9
Four-step LPP cycle	59.9	57.4	37.0	34.3	351.3	41.3

4.2.2 Simulation of three-step LPP cycle

Another three-step cycle can be constructed by pressurizing the column using light product from the adsorption step instead of feed pressurization, as illustrated in Fig. 2(b). In the light product pressurization (LPP) step, the inlet CO_2 gas-phase composition depends on the outlet stream of the adsorption step. For simplicity, it is assumed that the outlet stream of the adsorption step is first directed into a well-mixed tank before feeding to pressurize the column in the LPP step. Both neural network and detailed process model simulations are carried out based on this additional assumption. The same operating condition as earlier is used to compare the CSS profiles from both models. Figure S3 in the supporting information shows the column profiles of four state variables after cycle #1 and Fig. 7 illustrates the column profiles of four state variables at CSS from both simulations, respectively. The neural network profile predictions are remarkable. The impact of pressurizing the column with outlet stream of the adsorption step can be visualized from the CO_2 gas-phase compositions/solid loadings at the light product end. The final CO_2 solid loading at $z = 1$ at the end of the LPP step at CSS is ≈ 1 mol kg^{-1}, whereas in the previous case with FP, the final CO_2 solid loading at $z = 1$ at CSS is ≈ 1.4 mol kg^{-1}. It is worth mentioning that the pressurization step neural network model learnt the impact of inlet CO_2 gas-phase composition on the

spatiotemporal dynamics very well. As can be seen from Table 2, the calculated CO_2 purity and CO_2 recovery from the neural network simulations deviate by 2% from the detailed model simulations. The computational speeds for neural networks simulations were up to 18 times faster than the detailed model simulations.
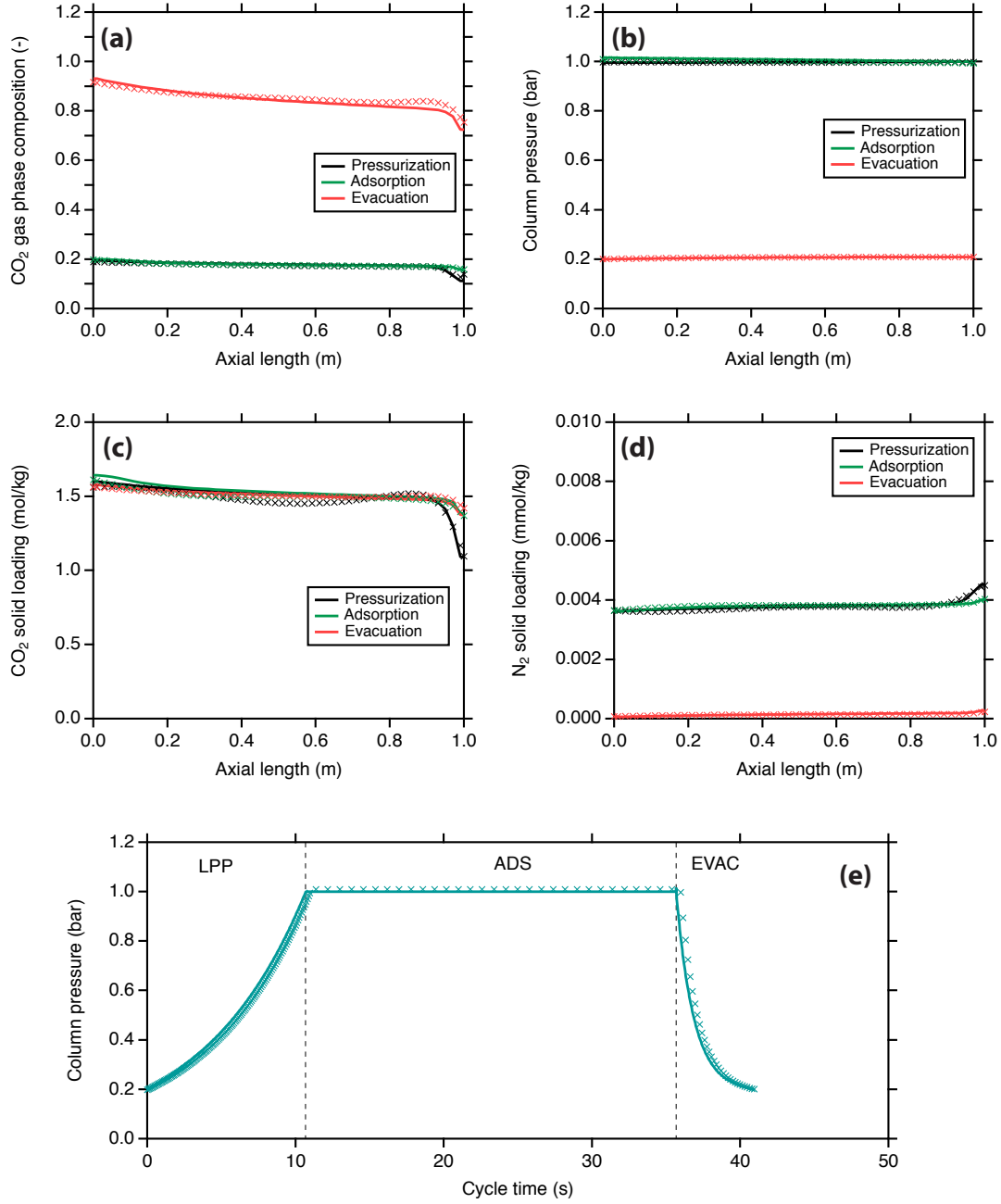
Figure 7: Comparison of cyclic steady state (CSS) profiles of four state variables obtained from the neural network (markers) and the detailed model (lines) simulations of three-step LPP cycle for the operating condition: t_{ADS}=25 s and $P_L = 0.2$ bar.

4.2.3 Simulation of four-step FP cycle

Here the four-step FP cycle, shown in Fig. 2(c), is constructed using the pressurization, the adsorption, the blowdown, the evacuation step models. For the operating condition of t_{ADS}

= 25 s, P_I=0.75 bar, and P_L=0.2 bar, the column profiles after cycle #1 and at CSS from both neural network- and detailed model-based simulations can be visualized in Fig. S4 (in the supporting information) and Fig. 8, respectively. The profiles calculated from both the simulations are in good agreement, although the neural network predicted CO_2 gas-phase compositions and CO_2 solid loadings at the end of the cycle #1 marginally advanced further through the column. Nevertheless, the overall effectiveness of the neural network models in predicting the CSS profiles are excellent. As can be seen from Table 2, the inclusion of the blowdown step between the adsorption and the evacuation step improved the CO_2 purity while the CO_2 recovery remained almost the same compared to the first case. This is because the blowdown step removed some residual N_2 from the column, as can be seen from the increase in the CO_2 gas-phase compositions across the column before the evacuation step. Finally, the CO_2 purity and the CO_2 recovery from the neural network simulations within 3% deviation compared to the detailed model simulations. In terms of computational efficiency, neural networks simulations saved up to 31 times the computational run-time.
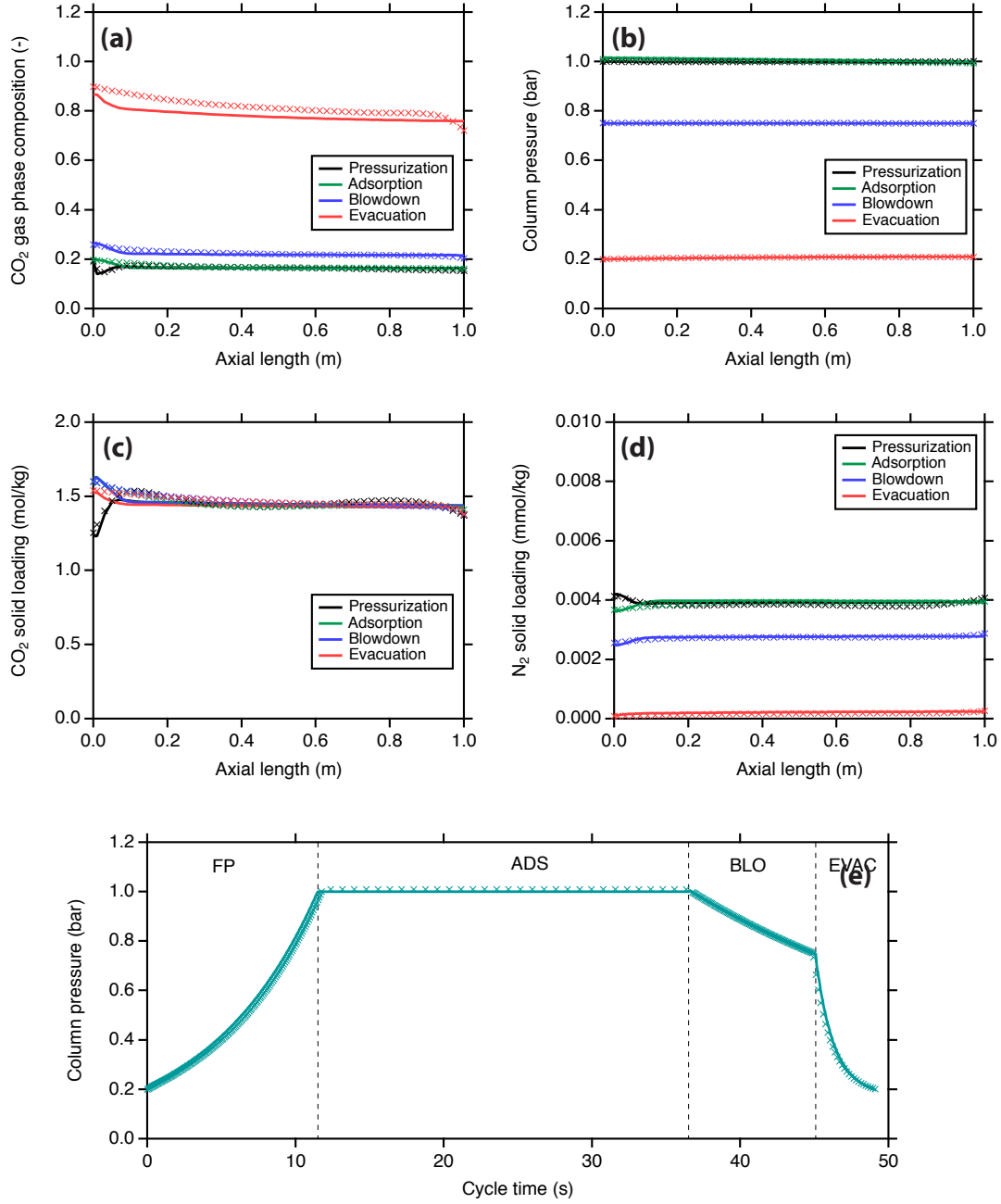
Figure 8: Comparison of cyclic steady state (CSS) profiles of four state variables obtained from the neural network (markers) and the detailed model (lines) simulations of four-step FP cycle for the operating condition: t_{ADS}=25 s, $P_I = 0.75$ bar, and $P_L = 0.2$ bar.

4.2.4 Simulation of four-step LPP cycle

Finally, the four-step LPP cycle is constructed and simulated for the operating condition: $t_{ADS} = 25$ s, P_I=0.75 bar, and P_L=0.2 bar. The profiles of four step variables after cycle #1

and at CSS are shown in Fig. S5 (in the supporting information) and Fig. 9, respectively. Again, the predictions from the neural network simulations are in very good agreement with the detailed model simulations. Based on Table 2, the CO_2 purity and the CO_2 recovery are within 3% deviation. The computational speeds achieved were 8.5 times higher than the detailed model.
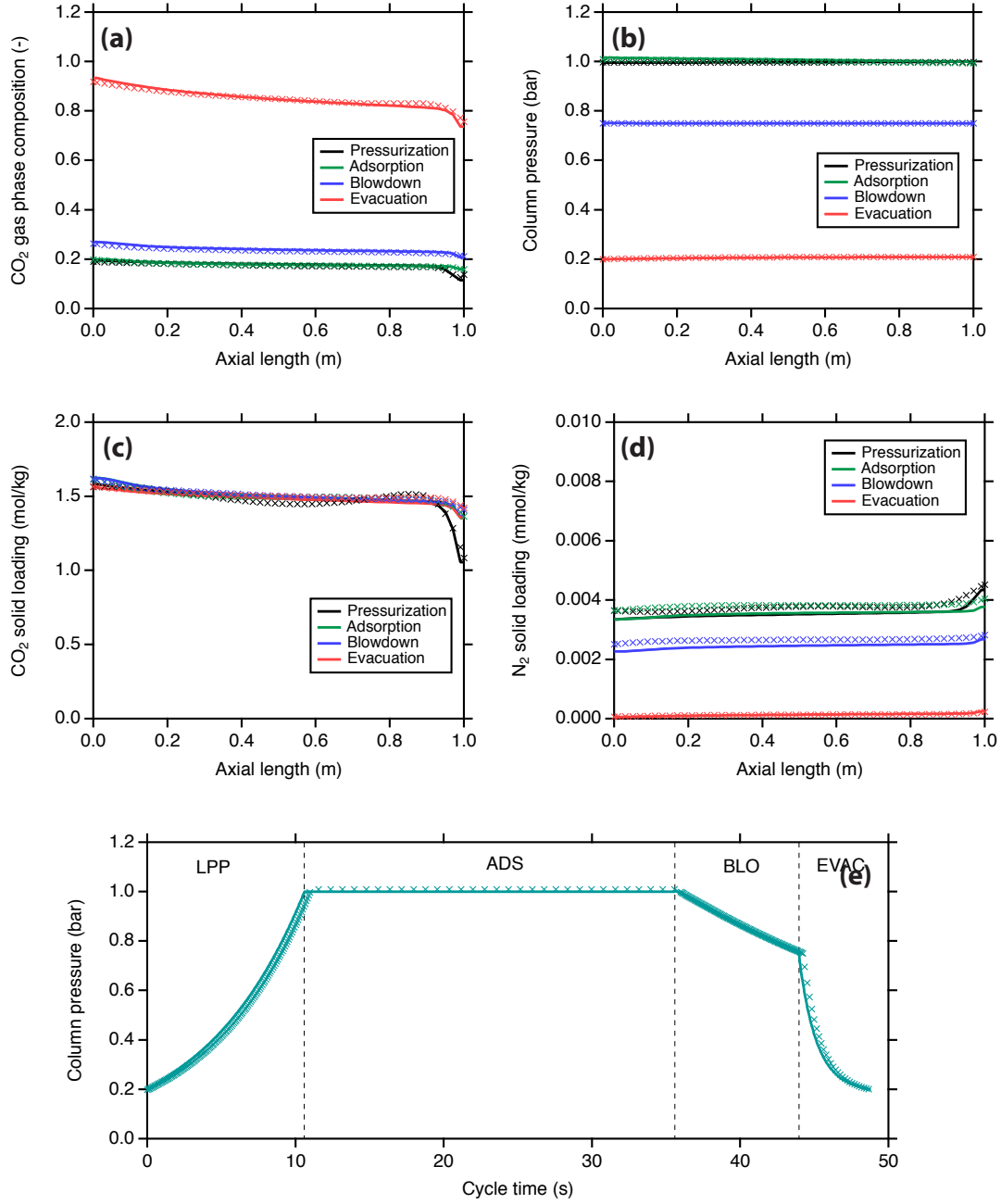
Figure 9: Comparison of cyclic steady state (CSS) profiles of four state variables obtained from the neural network (markers) and the detailed model (lines) simulations of four-step LPP cycle for the operating condition: t_{ADS}=25 s, $P_I = 0.75$ bar, and $P_L = 0.2$ bar.

4.2.5 Comparative performance for various operating conditions

The performance of the neural network models are tested based on 50 different randomly chosen operating conditions spanning the entire range considered for cycle synthesis study.

Here, CO_2 purity and CO_2 recovery calculated at CSS from neural network and detailed process model simulations are compared. Figure 10 compares the CO_2 purities obtained from both the models and the shaded region represents $\pm$ 2.5% deviation. For all the cycles, the CO_2 purity calculated based on neural network simulations lie within the 2.5% difference from the original values. On the other hand, the parity plot for CO_2 recovery illustrated in Fig. 11 show that the most of the predictions based on neural network simulations are within the 2.5% margin. Finally, the computational times incurred for 50 simulations of all four cycles are reported in Table 3. The differences in computational speed-ups for different cycles can be attributed to the fact that each cycle configuration takes different the number of cycles to achieve CSS condition depending on the operating conditions.
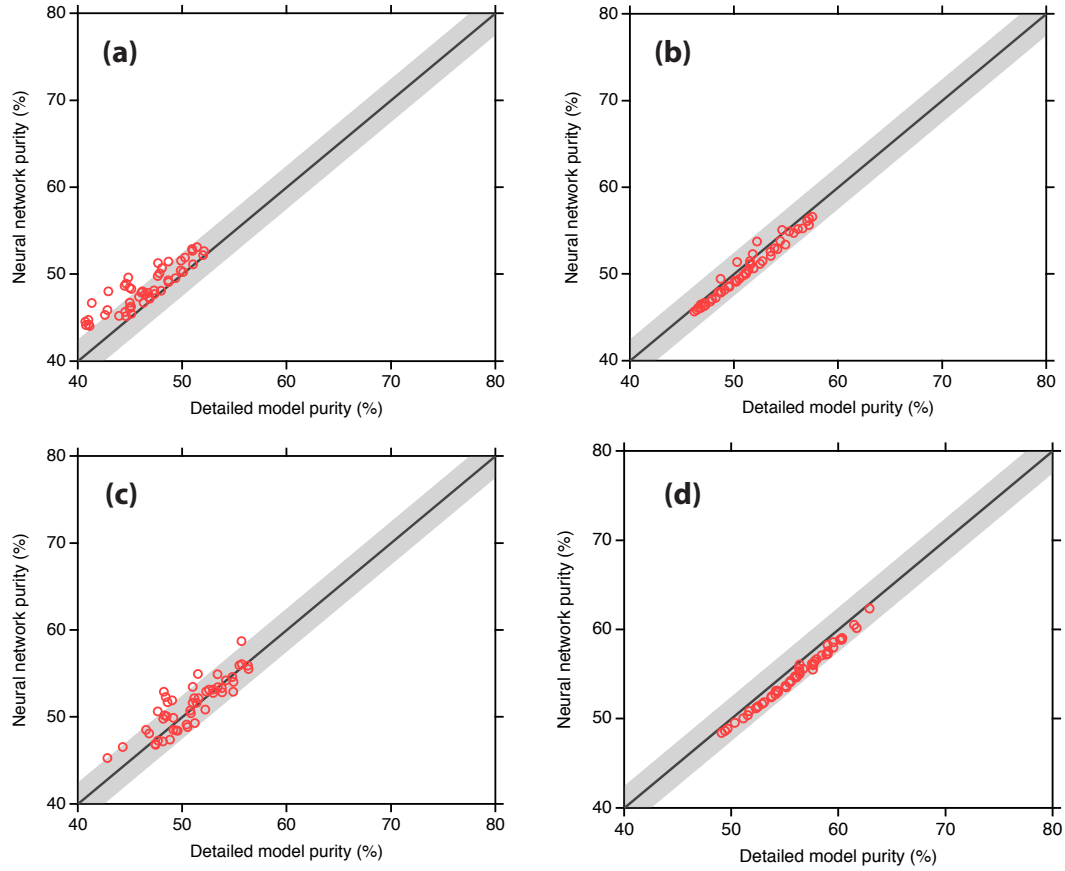
Figure 10: Parity plots of CO_2 purity obtained from the detailed process model- and the neural network-based simulations of (a) three-step FP cycle (b) three-step LPP cycle (c) four-step FP cycle (d) four-step LPP cycle at 50 different operating conditions. The shaded region represents $\pm 2.5\%$ deviation.

Figure 11: Parity plots of CO_2 recovery obtained from the detailed process model- and the neural network-based simulations of (a) three-step FP cycle (b) three-step LPP cycle (c) four-step FP cycle (d) four-step LPP cycle at 50 different operating conditions. The shaded region represents $\pm 2.5\%$ deviation.
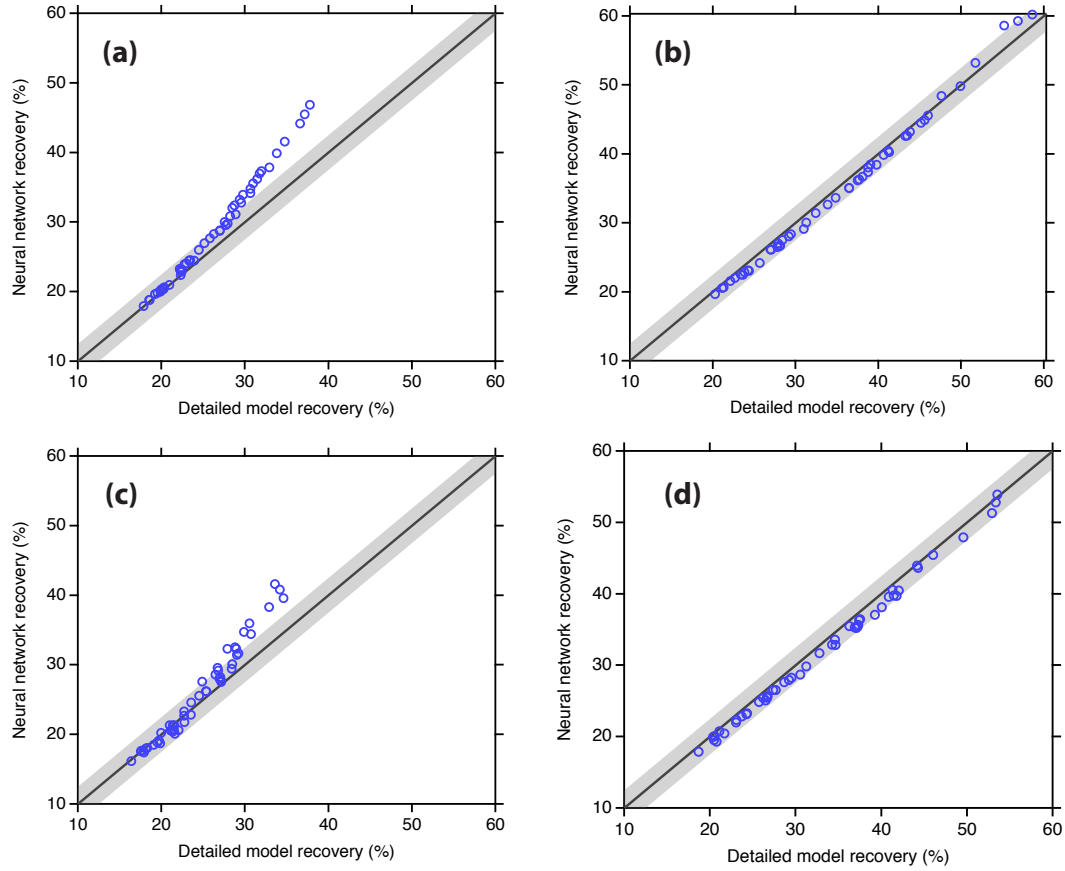
Table 3: Summary of computational times for carrying out 50 simulations using PANACHE, the detailed process model, and also PANACHE enabled detailed process model. Note that both the neural network and the detailed process model simulations were carried out on a CPU and the CPU seconds reported were on a 128 GB and 3.10 GHz workstation.

VSA cycles	Computational time (s)		
	Detailed model	PANACHE	PANACHE enabled detailed model
Three-step FP cycle	8452	409	4626
Three-step LPP cycle	11637	117	4371
Four-step FP cycle	9041	450	3764
Four-step LPP cycle	12422	129	4273

4.3 PANACHE enabled detailed process model

As mentioned previously, one of the challenges with the neural network-based simulations is achieving an overall mass balance convergence of less than 0.5% at CSS. While some of the simulations in Section 4.2.5 resulted in the overall mass balance error less than 0.5%, the overall mass balance error in other simulations was typically between 0.5-19%, although the column profiles remained nearly invariant after repeated cycling. This was primarily due to the small discrepancies while predicting the evacuation step duration in neural network simulations. For instance, see Fig. 6(e) where the neural networks slightly over-predicts the evacuation step duration compared to the detailed process model. Because of this, the moles out of the evacuation step calculated by the neural network simulations are marginally overestimated, resulting in increased overall mass balance error; however this effect remains insignificant on purity and recovery calculations (i.e. within 2.5% error). While the effectiveness of this approach relies on the flexibility to synthesize different cycles through faster computations, especially for screening several thousands of adsorbents, achieving the overall mass balance convergence can be a potential challenge. To overcome this, the CSS profiles from the neural network simulations can be fed into the detailed process model until the mass balance converges to less than 0.5% to increase the accuracy. This section explores the

possibility of using PANACHE to accelerate the convergence of the detailed process model. Here, the CSS profiles from PANACHE in Section 4.2.5 are fed as initial conditions to the detailed process model. Subsequently, the detailed process model is simulated until the overall mass balance is less than 0.5%. Figures S6 and S7 illustrate the parity plots of CO_2 purity and recovery for the four different VSA cycles. The computational times recorded for this hybrid approach can be seen in Table 2. As can be seen, the hybrid approach resulted in almost 100% accuracy in predicting CO_2 purity and recovery, however with only ≈ 3 times the computational speeds.

5   Conclusions

PANACHE, a physics-based neural network framework based on modern deep learning techniques was developed to rapidly synthesize and simulate cyclic adsorption processes. The deep neural networks employed herein are trained to learn the full spatiotemporal solutions of different state variables in cyclic adsorption processes by obeying the underlying conservation laws of mass and momentum along with adsorption equilibria. Unique neural networks are trained for different constituent steps of cyclic adsorption processes. The generalization capabilities are enabled by training the neural network models based on different initial conditions and operating parameters. In the present study, the proposed methodology was tested on the case of post-combustion CO_2 capture by synthesizing four simple VSA cycles using individual neural network models.

The results demonstrated that the trained neural network accurately predicts the spatiotemporal solutions of four state variables: CO_2 gas phase composition, column pressure, CO_2 solid loading, and N_2 solid loading. Particularly, the dynamics of state variables interior of the spatiotemporal domain were well-captured, even though no labelled data was provided. Using the trained neural network models, a three-step VSA cycle with feed pressurization, adsorption, and evacuation steps are constructed and simulated. The results

showed that the neural network-based simulations accurately predicted the CSS profiles of each step in the cycle. Later on, the a three-step VSA cycle with light product pressurization is simulated and the neural network predictions were found to be in excellent agreement with the detailed process simulations. Finally, the complexity was increased by synthesizing two four-step VSA cycles with feed and light product pressurization steps. The comparative performances between the neural network and the detailed model simulations at various operating conditions demonstrated the effectiveness of this methodology. In terms of computational speeds, the neural network simulations were up to 100 times faster than the detailed model simulations. Moreover, a hybrid approach was explored where the CSS profiles from the neural networks simulations were later fed into the detailed process model to improve the overall mass balance convergence while achieving $3\times$ computational speeds. In future, this methodology will be extended to incorporate more steps in order to synthesize and optimize complex adsorption cycles, a step towards superstructure-based optimal cycle synthesis.

While the capabilities of the proposed approach was successfully demonstrated for one material and different process configurations, future research should also focus on extending the methodology that can incorporate different isotherms as inputs, thereby allowing the framework to simultaneously evaluate different materials and processes. In addition, efforts have to be made in the subsequent studies to further improve the computational speeds of this methodology. Different sampling strategies to generate collocation points must also be explored in future to improve the model training in regions where solutions are stiff.

Nomenclature

Roman symbols

b adsorption equilibrium constant for site 1 (m^3 mol^{-1})

b_l bias of layer l of the neural network

$\boldsymbol{b}$ biases of the neural network

c	gas-phase concentration (mol m^{-3})
$\bar{c}$	non-dimensionalized gas-phase concentration (-)
d	adsorption equilibrium constant for site 2 (m^3 mol^{-1})
D_{L}	axial dispersion (m^2 s^{-1})
D_{m}	molecular diffusivity (m^2 s^{-1})
D_{P}	macropore diffusivity (m^2 s^{-1})
h	hidden layer
k	mass transfer coefficient (s^{-1})
L	column length (m)
n_{comp}	number of components
N_{i}	number of labelled data points
N_{k}	number of initial profiles
N_{l}	number of neurons in layer l of the neural network
N_{layers}	total number of layers in neural network architecture
N_{r}	number of collocation points
P	pressure (Pa)
$\bar{P}$	non-dimensionalized pressure (-)
q	concentration in the solid phase (mol kg^{-1})
$\bar{q}$	non-dimensionalized solid-phase concentration (-)
q^*	equilibrium concentration in the solid phase (mol kg^{-1})
q_{sb}	saturation concentration in the solid phase for site 1 (mol kg^{-1})
q_{sd}	saturation concentration in the solid phase for site 2 (mol kg^{-1})
q_{ref}	equilibrium concentration in the solid phase at feed condition (mol kg^{-1})
r_{i}	column inner radius (m)
r_{o}	column outer radius (m)
r_{p}	particle radius (m)
$\boldsymbol{r}$	residual of the partial differential equation

R	universal gas constant (Pa m^3 mol^{-1} K^{-1})
s	constituent step
t	time (s)
$\bar{t}$	normalized time (-)
T	temperature (K)
v	interstitial velocity (m s^{-1})
$\bar{v}$	non-dimensionalized interstitial velocity (-)
W_l	weights of layer l of the neural network
$\boldsymbol{W}$	weight matrices of the neural network
y	gas-phase composition (-)
z	bed coordinate (m)
$\bar{z}$	normalized axial coordinate (-)

Greek and math symbols

ε	column void fraction (-)
ε_p	particle void fraction (-)
ζ	step-related parameters
$\boldsymbol{\theta}$	parameters of the neural network
λ	weight term in the loss function
μ	gas viscosity (kg m^{-1} s^{-1})
ρ_s	adsorbent particle density (kg m^{-3})
σ	nonlinear activation function
τ	tortuosity (-)
ψ	dimensionless quantity in component and overall mass balance equations
ω	dimensionless quantity in component and overall mass balance equations
$\mathcal{L}$	mean-squared error
$\mathcal{L}_2$	averaged relative L2-norm

$\mathcal{S}$ state variable

Acronyms

ADS	adsorption step
ANN	artificial neural network
BLO	blowdown step
CSS	cyclic steady state
DM	detailed process model
DNN	deep neural network
DSL	dual-site Langmuir isotherm model
EVAC	evacuation step
FP	feed pressurization
L-BFGS	limited-memory Broyden-Fletcher-Goldfarb-Shanno
LDF	linear driving force
LPP	light product pressurization
LSTM	long short-term memory
MSE	mean-squared error
NN	neural network
ODE	ordinary differential equation
PANACHE	Physics-based artificial neural network framework for adsorption and chromatography emulation
PDE	partial differential equation
PSA	pressure swing adsorption
TSA	temperature swing adsorption
TVD	total variation diminishing
VSA	vacuum swing adsorption

Subscripts

0	initial
ADS	adsorption
BLO	blowdown
c	component mass balance
EVAC	evacuation
f	final
FP	feed pressurization
H	high
I	intermediate
in	inlet
L	low
lb	left boundary $(z = 0)$
p	overall mass balance
rb	right boundary $(z = L)$
ref	reference value for normalization

Acknowledgements

Supporting Information

Supporting Information containing dual-site Langmuir adsorption isotherm parameters, VSA simulation parameters, section on selecting appropriate N_k, figures illustrating the influence of averaged $\mathcal{L}_2$ norm on the number of initial profiles, different initial profiles used in neural network training, and the comparison of profiles for three-step LPP cycle, four-step FP cycle, and four-step LPP cycle is available free of cost through the ACS Website. The codes used for implementing the PANACHE model can be downloaded free of cost from https://github.com/ArvindRajendran

References

1. Sircar, S. Pressure swing adsorption. *Ind. Eng. Chem. Res.* **2002**, *41*, 1389–1392.

2. Ruthven, D. M.; Farooq, S.; Knaebel, K. S. *Pressure swing adsorption*; VCH Publishers, New York, 1994.

3. Haghpanah, R.; Majumder, A.; Nilam, R.; Rajendran, A.; Farooq, S.; Karimi, I. A.; Amanullah, M. Multiobjective optimization of a four-step adsorption process for post-combustion CO_2 capture via finite volume simulation. *Ind. Eng. Chem. Res.* **2013**, *52*, 4249–4265.

4. Biegler, L. T.; Jiang, L.; Fox, V. G. Recent advances in simulation and optimal design of pressure swing adsorption systems. *Sep. Purif. Rev.* **2005**, *33*, 1–39.

5. Haghpanah, R.; Nilam, R.; Rajendran, A.; Farooq, S.; Karimi, I. A. Cycle synthesis and optimization of a VSA process for postcombustion CO_2 capture. *AIChE J.* **2013**, *59*, 4735–4748.

6. Subraveti, S. G.; Pai, K. N.; Rajagopalan, A. K.; Wilkins, N. S.; Rajendran, A.; Jayaraman, A.; Alptekin, G. Cycle design and optimization of pressure swing adsorption cycles for pre-combustion CO_2 capture. *Appl. Energy* **2019**, *254*, 113624.

7. Agarwal, A.; Biegler, L. T.; Zitney, S. E. A superstructure-based optimal synthesis of PSA cycles for post-combustion CO_2 capture. *AIChE J.* **2010**, *56*, 1813–1828.

8. Agarwal, A.; Biegler, L. T.; Zitney, S. E. Superstructure-based optimal synthesis of pressure swing adsorption cycles for precombustion CO_2 capture. *Ind. Eng. Chem. Res.* **2010**, *49*, 5066–5079.

9. Dowling, A. W.; Vetukuri, S. R. R.; Biegler, L. T. Large-scale optimization strategies for pressure swing adsorption cycle synthesis. *AIChE J.* **2012**, *58*, 3777–3791.
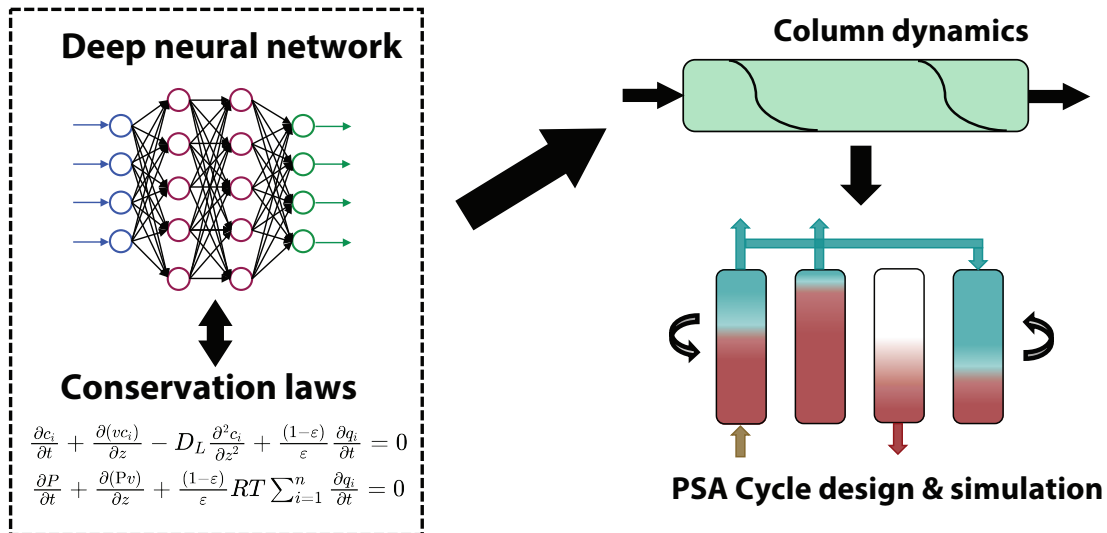
10. Farmahini, A. H.; Krishnamurthy, S.; Friedrich, D.; Brandani, S.; Sarkisov, L. Performance-based screening of porous materials for carbon capture. *Chem. Rev.* **2021**, *121*, 10666–10741.

11. Yan, Y.; Borhani, T. N.; Subraveti, S. G.; Pai, K. N.; Prasad, V.; Rajendran, A.; Nkulikiyinka, P.; Asibor, J. O.; Zhang, Z.; Shao, D.; Wang, L.; Zhang, W.; Yan, Y.; Ampomah, W.; You, J.; Wang, M.; Anthony, E. J.; Manovic, V.; Clough, P. T. Harnessing the power of machine learning for carbon capture, utilisation, and storage (CCUS)–A state-of-the-art review. *Ener. Env. Sci.* **2021**, *14*, 6122 – 6157.

12. Subraveti, S. G. Machine learning-based design and techno-economic assessments of adsorption processes for CO_2 capture.Ph.D. dissertation, University of Alberta, Edmonton, 2021.

13. Anna, H. R. S.; Barreto, A. G.; Tavares, F. W.; de Souza, M. B. Machine learning model and optimization of a PSA unit for methane-nitrogen separation. *Comput. Chem. Eng.* **2017**, *104*, 377 – 391.

14. Subraveti, S. G.; Li, Z.; Prasad, V.; Rajendran, A. Machine learning-based multiobjective optimization of pressure swing adsorption. *Ind. Eng. Chem. Res.* **2019**, *58*, 20412–20422.

15. Xiao, J.; Li, C.; Fang, L.; Böwer, P.; Wark, M.; Bénard, P.; Chahine, R. Machine learning–based optimization for hydrogen purification performance of layered bed pressure swing adsorption. *Int. J. Energy Res.* **2020**, *44*, 4475–4492.

16. Pai, K. N.; Prasad, V.; Rajendran, A. Experimentally validated machine learning frameworks for accelerated prediction of cyclic steady state and optimization of pressure swing adsorption processes. *Sep. Purif. Technol.* **2020**, *241*, 116651.

17. Oliveira, L. M. C.; Koivisto, H.; Iwakiri, I. G.; Loureiro, J. M.; Ribeiro, A. M.; Nogueira, I. B. Modelling of a pressure swing adsorption unit by deep learning and artificial Intelligence tools. *Chem. Eng. Sci.* **2020**, *224*, 115801.

18. Leperi, K. T.; Yancy-Caballero, D.; Snurr, R. Q.; You, F. 110th anniversary: Surrogate models based on artificial neural networks to simulate and optimize pressure swing adsorption cycles for CO_2 capture. *Ind. Eng. Chem. Res.* **2019**, *58*, 18241–18252.

19. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.

20. Lagaris, I.; Likas, A.; Fotiadis, D. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans Neural Netw.* **1998**, *9*, 987–1000.

21. Lagaris, I.; Likas, A.; Papageorgiou, D. Neural-network methods for boundary value problems with irregular boundaries. *IEEE Trans Neural Netw.* **2000**, *11*, 1041–1049.

22. Raissi, M.; Perdikaris, P.; Karniadakis, G. 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.

23. Kissas, G.; Yang, Y.; Hwuang, E.; Witschey, W. R.; Detre, J. A.; Perdikaris, P. Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. *Comput. Methods Appl. Mech. Eng.* **2020**, *358*, 112623.

24. Zhu, Q.; Liu, Z.; Yan, J. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Comput Mech* **2021**, *67*, 619–635.

25. 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.

26. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.* **2010**, *9*, 249–256.

27. Baydin, A. G.; Pearlmutter, B. A.; Radul, A. A.; Siskind, J. M. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.* **2018**, *18*, 1–43. .

28. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; Kudlur, M.; Levenberg, J.; Monga, R.; Moore, S.; Murray, D. G.; Steiner, B.; Tucker, P.; Vasudevan, V.; Warden, P.; Wicke, M.; Yu, Y.; Zheng, X. TensorFlow: A system for large-scale machine learning. arXiv. 2016 `https://arxiv.org/abs/1605.08695` (accessed 2022-02-02).

29. Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in pytorch. In *Proc. Adv. Neural Inf. Process. Syst. (NIPS) Workshop*, Long Beach, CA, USA, 2017.

30. LeCun, Y. A.; Bottou, L.; Orr, G. B.; Müller, K.-R. In *Neural Networks: Tricks of the Trade: Second Edition*; Montavon, G., Orr, G. B., Müller, K.-R., Eds.; Springer Berlin Heidelberg: Berlin, Heidelberg, 2012; pp 9–48.

31. Estupiñan Perez, L.; Sarkar, P.; Rajendran, A. Experimental validation of multi-objective optimization techniques for design of vacuum swing adsorption processes. *Sep. Purif. Technol.* **2019**, *224*, 553 – 563.

32. Krishnamurthy, S.; Rao, V. R.; Guntuka, S.; Sharratt, P.; Haghpanah, R.; Rajendran, A.; Amanullah, M.; Karimi, I. A.; Farooq, S. CO_2 capture from dry flue gas by vacuum swing adsorption: A pilot plant study. *AIChE J.* **2014**, *60*, 1830–1842.

33. Khurana, M.; Farooq, S. Integrated Adsorbent Process Optimization for Minimum Cost of Electricity Including Carbon Capture by a VSA Process. *AIChE J.* **2019**, *65*, 184–195.

34. Subraveti, S. G.; Roussanaly, S.; Anantharaman, R.; Riboldi, L.; Rajendran, A. Techno-economic assessment of optimised vacuum swing adsorption for post-combustion CO_2 capture from steam-methane reformer flue gas. *Sep. Purif. Technol.* **2021**, *256*, 117832.

35. Burns, T. D.; Pai, K. N.; Subraveti, S. G.; Collins, S. P.; Krykunov, M.; Rajendran, A.; Woo, T. K. Prediction of MOF performance in vacuum swing adsorption systems for postcombustion CO_2 capture based on integrated molecular simulations, process optimizations, and machine learning models. *Environ. Sci. Technol.* **2020**, *54*, 4536–4544.

Physics-based ANN framework for Adsorption & CHromatography Emulation (PANACHE)

Deep neural network

Column dynamics

Conservation laws

$$\frac{\partial c_i}{\partial t} + \frac{\partial (v c_i)}{\partial z} - D_L \frac{\partial^2 c_i}{\partial z^2} + \frac{(1-\varepsilon)}{\varepsilon} \frac{\partial q_i}{\partial t} = 0$$

$$\frac{\partial P}{\partial t} + \frac{\partial (Pv)}{\partial z} + \frac{(1-\varepsilon)}{\varepsilon} RT \sum_{i=1}^{n} \frac{\partial q_i}{\partial t} = 0$$

PSA Cycle design & simulation

For Table of Contents Only