Journal of Chromatography A 1708 (2023) 464346

Contents lists available at ScienceDirect

Journal of Chromatography A

journal homepage: www.elsevier.com/locate/chroma

Physics-informed neural networks to solve lumped kinetic model for chromatography process

Si-Yuan Tang [a,b], Yun-Hao Yuan [b], Yu-Cheng Chen [a], Shan-Jing Yao [a], Ying Wang [b], Dong-Qiang Lin [a,*]

[a] College of Chemical and Biological Engineering, Zhejiang University, Hangzhou 310058, China
[b] Manufacturing Science and Technology, Global Manufacturing, WuXi Biologics, Wuxi 214000, China

ARTICLE INFO

Keywords:
Chromatography
Lumped kinetic model
Physics-informed neural network
Process modeling
Artificial neural network
Digital twin

ABSTRACT

Numerical method is widely used for solving the mechanistic models of chromatography process, but it is time-consuming and hard to response in real-time. Physics-informed neural network (PINN) as an emerging technology combines the structure of neural network with physics laws, and is getting noticed for solving physics problems with a balanced accuracy and calculation speed. In this research, a proof-of-concept study was carried out to apply PINN to chromatography process simulation. The PINN model structure was designed for the lumped kinetic model (LKM) with all LKM parameters. The PINN structure, training data and model complexity were optimized, and an optimal mode was obtained by adopting an in-series structure with a nonuniform training data set focusing on the breakthrough transition region. A PINN for LKM (LKM-PINN) consisting of four neural networks, 12 layers and 606 neurons was then used for the simulation of breakthrough curves of chromatography processes. The LKM parameters were estimated with two breakthrough curves and used to infer the breakthrough curves at different residence times, loading concentrations and column sizes. The results were comparable to that obtained with numerical methods. With the same raw data and constraints, the average fitting error for LKM-PINN model was 0.075, which was 0.081 for numerical method. With the same initial guess, the LKM-PINN model took 160 s to complete the fitting, while the numerical method took 7 to 72 min, depending on the fitting settings. The fitting speed of LKM-PINN model was further improved to 30 s with random initial guess. Thus, the LKM-PINN model developed in this study is capable to be applied to real-time simulation for digital twin.

1. Introduction

As a foundation for achieving the smart control of modern biomanufacturing processes and realizing the digital twin, the process model plays a critical role for correlating the input variables with output performance indicators. Specifically, for a chromatography process, the lumped kinetic model (LKM) and general rate model (GRM) are widely used and proven effective in predicting the on-column adsorption (sample loading step) and desorption (elution step) behavior for both affinity chromatography and ion exchange chromatography [1–3]. Besides the applications in batch process, GRM was applied to state-of-the-art twin-column [4,5] or three-column [6] continuous processes for the optimization of process productivity and resin utilization [7]. Recently, its application was also extended to the area of integrated continuous process for smart control [8].

Since both the LKM and GRM consist of several partial differential equations (PDEs) and their equations are complicated, numerical methods are usually adopted in offline chromatography process simulation. Though these numeric methods can provide accurate and comprehensive results for process design, they usually take dozens of minutes to complete the calculation. Considering that the downstream chromatography process is usually done in a few hours and every step lasts for only several minutes, it is hard to deploy these numerical methods for real-time process simulation and control. Thus, for the application of mechanism models in downstream process digital twin, besides the model precision and accuracy, calculation speed is also a key performance indicator, and traditional numerical methods cannot provide a reasonable response to this requirement.

In contrary of obtaining numerical solutions of mechanism model, the data-driven model like artificial neural network usually provides fast

* Corresponding author.
E-mail address: lindq@zju.edu.cn (D.-Q. Lin).

https://doi.org/10.1016/j.chroma.2023.464346
Received 15 May 2023; Received in revised form 29 August 2023; Accepted 29 August 2023
Available online 9 September 2023

calculation speed. Since there is no physics constraint in the data-driving model, it's performance cannot be guaranteed, and using data-driving model alone is not considered a robust method for process control. To mitigate the gap between mechanism model and data-driven model, a hybrid approach is a feasible solution to leverage the benefits of data-driven model to speed up the calculation for mechanistic models [9], and some attempts have been made to achieve that.

Lin et al. [10,11] invented a hybrid method by generating thousands numerical solutions of GRM (breakthrough curves) using orthogonal collocation method, and then extracting the key features on these curves to train two artificial neural network (ANN) models. After that, the ANN models can be used to simulate the chromatography process as a surrogate model of GRM, and the time duration for solving neural network is much less than that for solving the PDEs. This invention makes it possible to achieve a faster process simulation in terminal, but it still has some limitations. Firstly, it is a kind of supervised learning. The two ANN models were generated based on the solutions of GRM through orthogonal collocation method, which means the error of numerical solutions will eventually be carried to the ANN and results in unexpected biases. Besides, the two ANN models are both `"black-box" model with relatively low degree of hybridization. Regardless of how the training data were generated, either from experiment data or numerical solution of GRM, the ANN model itself is still not explained by the constraints and PDEs of GRM, but by the `"shape" of breakthrough curve. From this perspective, this approximate solution is closer to a data-driven model rather than a mechanistic model, and its accuracy is not guaranteed physically.

Narayanan et al. [12] proposed a quantitative assessment for the degrees of hybrid model, and also adopted ANN to simplify the mechanistic model for protein A chromatography-based capture application. They designed three hybrid models with different degrees of hybridization, namely the Iso-Hybrid, the MTI-Hybrid and the Lumped-Hybrid model. All these three models adopted ANN to simplify the expressions of adsorption isotherm, mass transfer coefficient and solid phase concentration, but these ANN themselves are still data-driven model and not explained by any physics law. Though it doesn't address the problem that it is difficult to obtain the analytic solution and time-consuming to get the numerical solution for the PDEs, it demonstrated the possibility to use a more flexible data-driven function (ANN) instead of rigid mechanistic function to improve the prediction performance.

Lu et al. [13–15] presented a framework by introducing the physics-informed neural networks (PINN) for solving PDEs, which is considered applicable for multiple types of PDEs, integro-differential equations (IDEs), fractional differential equations (FDEs) and stochastic differential equations (SDEs). The concept of PINN is similar to the well-known generative adversarial network (GAN), which is also a kind of unsupervised learning model. The PINN is built on normal neural networks. The difference is that when the PINN is used for solving PDEs, the original PDEs and the constraints of PDEs are integrated into the loss function of the neural network. Since the differential terms of a neural network can easily be calculated through the automatic differentiation approach, by optimizing the neural network parameters to minimize the loss function to zero, those PDEs and constraints of PDEs can therefore be satisfied in the PINN, and the PINN can get trained. Theoretically, the native PINN can be trained with no requirement on numerical solutions. With these features, the PINN can solve mechanism models as fast as the data-driven model and as robust as numerical methods, which makes it an excellent tool for digital twin to achieve real-time process control and process optimization [16].

In recent years, the PINN approach has been used in different areas for the solution of different PDEs [17,18]. Santana et al. [19] adopted PINN to characterize fixed-bed column adsorption behavior, and demonstrate its great benefits, specifically in the solving speed for real-time optimization and process control. However, their model was built for a special case that didn't consider the impact of chromatography model parameters, and the trained model can't be transferred to other situations.

Despite its great advantage in applications, PINN is not easy to get well-trained as expected, and different mitigation actions have been tried to improve the spectral bias and convergence rate of the loss function [20]. One of the most straightforward method is the hybrid approach with the help of labeled data. Subraveti et al. [21] applied a physics-based neural network to simulate the dynamics of generic pulse injections in chromatography columns. Instead of a native PINN model that only using the PDEs, initial conditions and boundary conditions to form the loss function alone, they also combined some labeled data that the numerical solutions of PDEs solved with a finite volume method into the loss function to speed up the training procedure. However, they only considered limited model parameters (solute concentration and the time of injection) in their model, and the model can't be transferred to other situations when any other model parameter changes. Söderström [22] developed a PINN model for LKM with linear adsorption isotherm, and some observed data were also involved in the training of model. Four of eight LKM parameters, including the axial dispersion coefficient, external porosity, lumped mass transfer coefficient and Henry coefficient, were considered in the PINN model to deal with different situations. The positive results of this model demonstrated the effectiveness of using a PINN model to simulate the chromatography process. However, the parameters' range in this model is not wide enough to cover a typical chromatography process for bioproduct (e.g. monoclonal antibody) purification, and not all the parameters were considered in model training. Moreover, this model was evaluated based on the overall error between PINN and numerical solutions in its design space. The worst-case was not presented in the paper and the robustness of this model was not reported.

In this paper, a PINN model was developed for LKM with Langmuir adsorption as a proof-of-concept to demonstrate the feasibility of using PINN in real-time simulation. Figure 1 presents the overall workflow for this research. The structure of PINN was firstly designed and screened with two unfixed LKM parameters. Then all of the eight LKM parameters were introduced into selected PINN model structure, and the model performance was improved by adding extra monotonic constraints, optimizing data density, data distribution and model complexity. After optimization, the final LKM-PINN model was obtained. The model has been evaluated based on its overall error, error distribution and error in the worst condition compared to numerical solutions. Meanwhile, the error of every PDEs / constraints was also evaluated. The final LKM-PINN model was used for the fitting and predicting of experimental breakthrough curves, and the performance is better than numerical method. In summary, the LKM-PINN model developed in this study can be applied to real-time simulation and digital twin. The methodology provided in this paper can also be utilized to develop other PINN for different purposes.

2. Theory

2.1. Lumped kinetic model

The single component lumped kinetic model [22] was investigated in this study as,

$$\frac{\partial c}{\partial t}+\frac{(1-\varepsilon_t)}{\varepsilon_t}\frac{\partial q}{\partial t}+\frac{u}{\varepsilon_t}\frac{\partial c}{\partial x}=D_{ax}\frac{\partial^2 c}{\partial x^2} \tag{1}$$

where c is the concentration in mobile phase, q is the concentration in stationary phase, ε_t is the total porosity, u is the superficial velocity, D_{ax} is the axial dispersion coefficient, t is the processing time, and x is the spatial coordinate of bed.

In the case of linear binding model, we have

$$\frac{\partial q}{\partial t}=k_a c-k_d q \tag{2a}$$

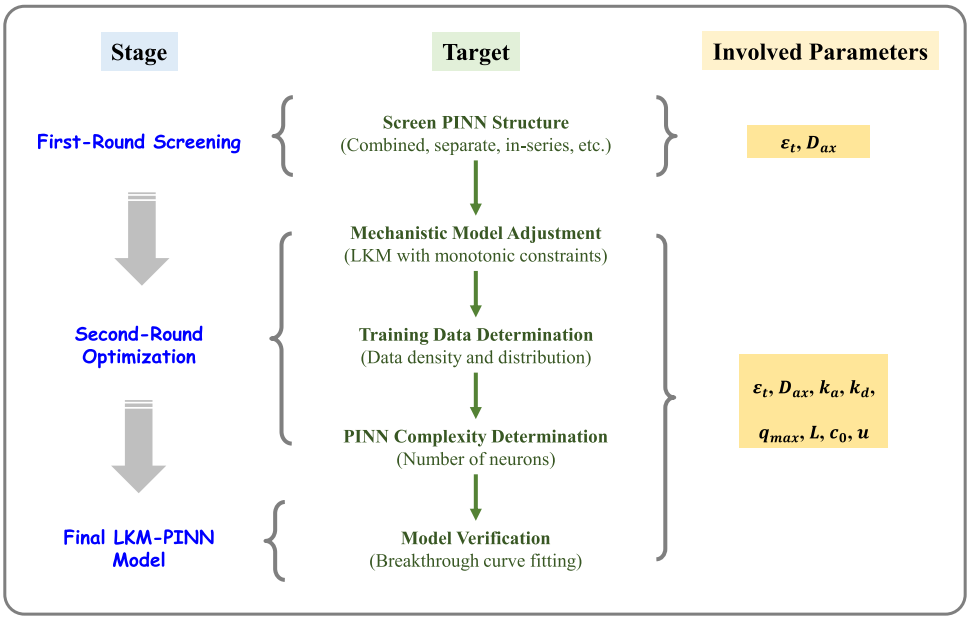


Fig. 1. Workflow for PINN development.

Similarly, in the case of Langmuir binding model, we have

$$\frac{\partial q}{\partial t} = k_a c q_{\max}\left(1 - \frac{q}{q_{\max}}\right) - k_d q \tag{2b}$$

where k_a is the rate constant for adsorption, k_d is the rate constant for desorption and q_{max} is the maximum adsorption capacity.

For a binding (adsorption) process with constant loading concentration c_0, Eqs. (1) and (2a) (or Eq. (2b)) are subjected to the initial and boundary conditions as

$$\text{(I.C. 1)}\ c(x, t=0) = 0 \tag{3a}$$

$$\text{(I.C. 2)}\ q(x, t=0) = 0 \tag{3b}$$

$$\text{(B.C. 1)}\ \frac{\varepsilon_t D_{ax}}{u}\frac{\partial c}{\partial x}\bigg|_{x=0} = c(x=0, t) - c_{inj}(t) \tag{4a}$$

$$\text{(B.C. 2)}\frac{\partial c}{\partial x}\bigg|_{x=L} = 0 \tag{4b}$$

and

$$c_{inj}(t) = \begin{cases} 0 & t = 0 \\ c_0 & t > 0 \end{cases} \tag{5}$$

where L is the column bed height and c_{inj} is the injection concentration.

Furthermore, for a binding process on fixed-bed, we have

$$\frac{\partial c}{\partial t} \geq 0,\ \forall x \geq 0 \tag{6a}$$

$$\frac{\partial q}{\partial t} \geq 0,\ \forall x \geq 0 \tag{6b}$$

$$\frac{\partial c}{\partial x} \leq 0,\ \forall t \geq 0 \tag{6c}$$

$$\frac{\partial q}{\partial x} \leq 0,\ \forall t \geq 0 \tag{6d}$$

Equations (6a) to (6d) can be reorganized as below to assist model training as

$$\frac{\partial c}{\partial t} - \left|\frac{\partial c}{\partial t}\right| = 0 \tag{7a}$$

$$\frac{\partial q}{\partial t} - \left|\frac{\partial q}{\partial t}\right| = 0 \tag{7b}$$

$$\frac{\partial c}{\partial x} + \left|\frac{\partial c}{\partial x}\right| = 0 \tag{7c}$$

$$\frac{\partial q}{\partial x} + \left|\frac{\partial q}{\partial x}\right| = 0 \tag{7d}$$

Equations (7a) to (7d) were introduced into PINN model training since the second-round optimization, which is expected to improve the model performance by minimizing unexpected fluctuations and keeping the pattern of overall trend consistent with that of a regular adsorption process.

2.2. Design space of LKM parameters

The parameters of LKM have been classified into three categories. The first one (Type I) is for the process parameter or material property, including ε_t, L, u, D_{ax}, k_a, k_d, q_{max} and c_0. The second one (Type II) is for the coordinate variable, including x and t. The third one (Type III) is for the output variable, including c and q. In this study, it was assumed that the Type II parameters are independent of Type I parameters.

Table 1 presents the parameters involved in LKM with their units and design space in this study. The design space was defined based on the knowledge for a typically protein chromatography process, and the range of u and t have been expanded in final model to match the situation of real industrial purification process.

2.3. Physics-informed neural network

Compared to native physics-informed neural network model, the PINN model employed in this study is a hybrid model with both physics law and numerical solutions in its loss function to assist model training. Denoting the trainable parameters (weights and biases) of neural networks in PINN model as $\boldsymbol{\theta}$, the PINN model is a function of LKM parameters (as input variables of neural networks) with neural network model parameters $\boldsymbol{\theta}$ (the weights and bias of every neuron), which is

Table 1
List of LKM parameters and their design space for PINN model.

Type	Parameter	Unit	Range (for optimization)	Range (for final model)
Type I	ε_t	-	[0.5, 0.9]	[0.5, 0.9]
	L	M	[0.1, 0.3]	[0.1, 0.3]
	u	m/h	[0.5, 4]	[0.5, 8]
	q_{max}	g/L resin	[0.01, 200]	[0.01, 200]
	k_a (Linear)	L·(L resin)$^{-1}$·s^{-1}	[10^{-5}, 1]	NA
	k_a (Langmuir)	L·g^{-1}·s^{-1}	[10^{-5}, 1]	[10^{-5}, 1]
	k_d	s^{-1}	[10^{-5}, 1]	[10^{-5}, 1]
	D_{ax}	m^2/s	[10^{-9}, 10^{-2}]	[10^{-9}, 10^{-2}]
	c_0	g/L	[0.5, 10]	[0.5, 10]
Type II	t	S	[0, 3600]	[0, 10800]
	x	M	[0.1, 0.3]	[0.1, 0.3]
Type III	c	g/L	[0, c_0]	[0, c_0]
	q (Linear)	g/L resin	$[0, \frac{k_a c_0}{k_d}]$	NA
	q (Langmuir)	g/L resin	$[0, \frac{k_a c_0}{(k_a c_0 + k_d) * q_{max}}]$	$[0, \frac{k_a c_0}{(k_a c_0 + k_d) * q_{max}}]$

denoted as $\mathcal{N}(\varepsilon_t; L; u; D_{ax}; k_a; k_d; q_{\max}; c_0; t; x; \boldsymbol{\theta})$.

The loss function of PINN model $\mathscr{L}(\boldsymbol{\theta}; \mathscr{T}_{PDE}; \mathscr{T}_{Data})$ consists of two parts: The PDE residue loss $\mathscr{L}_{PDE}(\boldsymbol{\theta}; \mathscr{T}_{PDE})$ for physics laws and the pure data loss $\mathscr{L}_{Data}(\boldsymbol{\theta}; \mathscr{T}_{Data})$ for numerical solutions, which can be written as

$$\mathscr{L}(\boldsymbol{\theta}; \mathscr{T}_{PDE}; \mathscr{T}_{Data}) = \mathscr{L}_{PDE}(\boldsymbol{\theta}; \mathscr{T}_{PDE}) + \mathscr{L}_{Data}(\boldsymbol{\theta}; \mathscr{T}_{Data}) \tag{10}$$

The initial conditions and boundary conditions of LKM were combined into the PDE residue loss function $\mathscr{L}_{PDE}$ during the training of PINN model as

$$\mathscr{L}_{PDE}(\boldsymbol{\theta}; \mathscr{T}_{PDE}) := \omega_f \mathscr{L}_f(\boldsymbol{\theta}; \mathscr{T}_f) + \omega_b \mathscr{L}_b(\boldsymbol{\theta}; \mathscr{T}_b) + \omega_i \mathscr{L}_i(\boldsymbol{\theta}; \mathscr{T}_i) \tag{11}$$

and

$$\mathscr{L}_f(\boldsymbol{\theta}; \mathscr{T}_f) := \frac{1}{|\mathscr{T}_f|} \sum_{x \in \mathscr{T}_f} x^2 \tag{12a}$$

$$\mathscr{L}_b(\boldsymbol{\theta}; \mathscr{T}_b) := \frac{1}{|\mathscr{T}_b|} \sum_{x \in \mathscr{T}_b} x^2 \tag{12b}$$

$$\mathscr{L}_i(\boldsymbol{\theta}; \mathscr{T}_i) := \frac{1}{|\mathscr{T}_i|} \sum_{x \in \mathscr{T}_i} x^2 \tag{12c}$$

where $\mathscr{T}_f$, $\mathscr{T}_b$ and $\mathscr{T}_i$ are residuals for PDE equations, B.C. equations and I.C. equations respectively; and ω_f, ω_b and ω_i are the weights for each term.

When the monotonicity constraints Eq. (7a)–(7d) were introduced, one additional loss term $\mathscr{L}_m$ was added to Eq. (11) to represent this loss, which is given by

$$\mathscr{L}_m(\boldsymbol{\theta}; \mathscr{T}_m) := \frac{1}{|\mathscr{T}_m|} \sum_{x \in \mathscr{T}_m} x^2 \tag{12d}$$

where $\mathscr{T}_m$ are residuals for Eqs. (7a)–(7d).

The error between numerical solutions and PINN model predictions were combined into the pure data loss function $\mathscr{L}_{Data}$ during the training of PINN model:

$$\mathscr{L}(\boldsymbol{\theta}; \mathscr{T}_{Data}) := \omega_c \mathscr{L}_c(\boldsymbol{\theta}; c_z) + \omega_q \mathscr{L}_q(\boldsymbol{\theta}; q_z) \tag{13}$$

and

$$\mathscr{L}_c(\boldsymbol{\theta}; c_z) := \frac{1}{n} \sum_{z=1}^{n} (c_z - \widehat{c}_z)^2 \tag{14a}$$

$$\mathscr{L}_q(\boldsymbol{\theta}; q_z) := \frac{1}{n} \sum_{z=1}^{n} (q_z - \widehat{q}_z)^2 \tag{14b}$$

where $\mathscr{L}_c$ and $\mathscr{L}_q$ are the pure data loss for c and q, respectively; ω_c and ω_q are the weights for each term.

Figure 2 presents the scheme of this PINN model. As mentioned above, there are two kinds of data points for the calculation of PDE loss and pure data loss. One is the training point that have numerical solutions (namely "pure data point"), and the other one is the training point for model terms calculation without numerical solution (namely "grid point"). The function of pure data points for PINN is the same as that for a normal neural network, which used for the model training by minimizing the error between prediction values ($\widehat{c}$ and $\widehat{q}$) and observed values (c and q). The grid points without numerical solution were used to minimize the PDE loss by calculating the differentiation terms ($\frac{\partial c}{\partial t}$, $\frac{\partial c}{\partial x}$, $\frac{\partial q}{\partial t}$ and $\frac{\partial q}{\partial x}$) at these given points. The model parameters $\boldsymbol{\theta}$) are determined by minimizing the overall loss combining PDE loss and pure data loss.

The root mean squared error (RMSE) is obtained by taking the squared root of the average squared difference between the predicted value and the actual value. It was used for model training and testing on various targets, including the loss functions (actual values equal 0) and pure data points or test data points obtained from numerical method. In this study, all the RMSEs were calculated based on the normalized dimensionless data.

2.4. Activation functions of neural networks

The sigmoid function was adopted as the output layer activation to enhance the model performance by restricting the predicted values in the range of 0 to 1 [23]. For the hidden layers, as suggested by Mishra et al. [24], PINNs with ReLu and other non-smooth activation functions fails to converge to the right exact solution in the limit of an infinite training dataset. Therefore, the tanh function was adopted as the activation function for all the hidden layers.

2.5. Input variables normalization

During the training of neural network with gradient descent, it is recommended to normalize the input variables to avoid gradient vanishing. In this study, the input variables ε_t, L, u, c_0, t and x were normalized to a range [−1,1] through a linear map $f: \mathbb{R} \to [-1, 1]$. Specifically, for given variable z with upper limit $z_{\max}$ and lower limit $z_{\min}$, it was normalized to z' with the map

$$f(z; z_{\max}; z_{\min}) := 1 + 2\frac{z - z_{\max}}{z_{\max} - z_{\min}} \tag{8a}$$

For input variables D_{ax}, k_a, k_d and $q_{\max}$ that have wide range and high

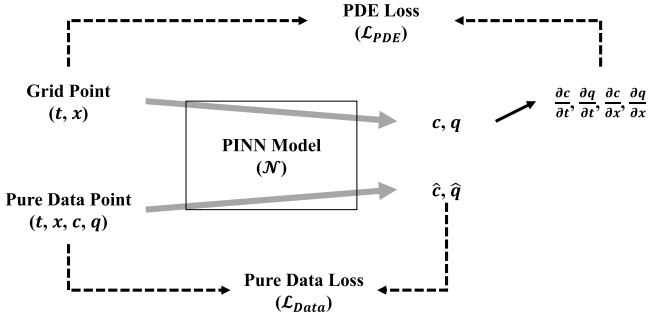


Fig. 2. Scheme of PINN.

variance, a log transformation map $f_{log}: \mathbb{R}\rightarrow[-1,1]$ was applied to normalize the input variables to a range [−1,1]. Specifically, for given variable z with upper limit z_{max} and lower limit z_{min}, it was normalized to $z^{'}$ with the map

$$f_{\log}(z; z_{\max}; z_{\min}) := 1 + 2\frac{\log_{10} z - \log_{10} z_{\max}}{\log_{10} z_{\max} - \log_{10} z_{\min}} \tag{8b}$$

2.6. Output/target variable normalization

The target variables are also recommended to be normalized to the same scale to avoid the effects of variability of each target variable on the scale-sensitive cost function. A linear map $f: \mathbb{R}\rightarrow[0,1]$ is applied on both target variables for lumped kinetic model to a range [0, 1].

For mobile phase concentration c, it was normalized to $c^{'}$ with the map:

$$f(c; c_0) := \frac{c}{c_0} \tag{9a}$$

and for stationary phase concentration q, it was normalized to $q^{'}$ with the map:

$$f(q; a; c_0) := \frac{q}{ac_0} \tag{9b}$$

where $a = \frac{k_a}{k_d}$ for a linear binding model and $a = \frac{k_a q_{max}}{k_a c_0 + k_d}$ for a Langmuir binding model.

The outputs of a sigmoid function ranges from 0 to 1, which naturally clips the predicted values to normalization range for target variables.

3. Methods

3.1. PINN model structure

Based on the classification of LKM parameters, six different PINN model structures were investigated in the first-round screening as presented in Fig. 3.

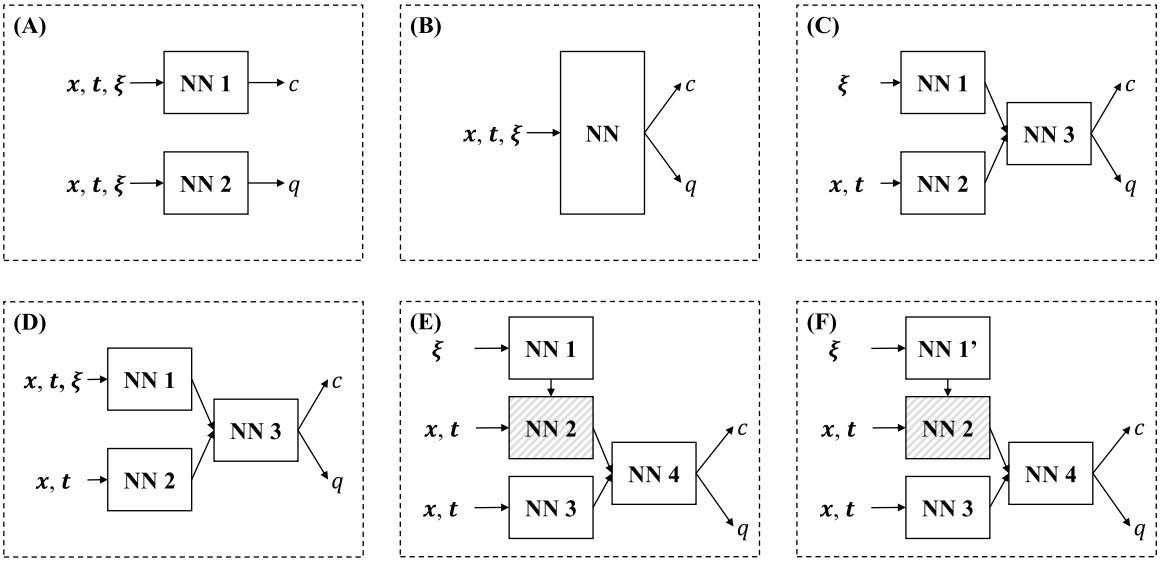
Design (A) consists of two different neural networks (NN 1 and NN 2) to approximate the function of Type III variables c and q with Type I and II variables, respectively. Design (B) consists of one combined neural network to approximate the functions, and c and q are strongly connected in this approach. Design (C) consists of three neural networks (NN 1, NN 2 and NN 3). The first two neural networks (NN 1 and NN 2) receive the Type I and Type II variables, respectively, and the last neural network (NN 3) combined the output of NN 1 and NN 2 to generate output variables c and q. These three PINN model structures were also discussed by Söderström [22].

Besides, three new PINN model structures were introduced in this paper. Design (D) was optimized based on Design (C), which added the Type II input variables x and t to NN 1 in addition to NN 2 and made NN 1 also a function of x and t. Thus, the correlation between Type I and Type II variables can be captured in this design. Design (E) consists of four neural networks (NN 1, NN 2, NN 3 and NN 4). The NN 1 and NN 3 are designed to receive the Type I and Type II variables, respectively. In addition to NN 3, another neural network NN 2 is also used to receive the Type II variables, but the weights and bias of NN 2 are all determined by NN 1 (as an output of NN 1). NN 4 is used to combined the output of NN 2 and NN 3 to generate output variables c and q. Compared to Design (D), the NN 1 in Design (D) was replaced by an in-series structure with NN 1 and NN 2 in Design (E), which decoupled the Type I and Type II variables. Design (F) has a similar structure to Design (E) and the only difference is that the deep neural network NN 1 in design (E) was replaced by a flat neural network NN 1' in Design (F). The philosophy of Design (E) and Design (F) is that physically the Type I parameters determine the "shape" of the curve: $c(x,t)$ and $q(x,t)$ (relationship between NN 1 and NN 2), and NN 3 was added to slightly increase the flexibility of the entire PINN model.

Different configurations of neural networks were investigated with the same quantity of total trainable neurons at different stage. For the same design, the number of neurons in each neural network were adjusted to set different weights on each neural network, with the same total neurons unchanged. Table 2 summarizes the configurations of PINN model investigated in this work for first-round model screening with two volatile Type I variables. Note that the neurons in NN 2 of Design (E) and Design (F) are not trainable and not counted in the total trainable neurons.

The second-round optimization was carried out on Design (E), and to characterize all eight unfixed Type I variables. The neurons of each layer were doubled compared to that of the first-round screening. The trainable neurons were further doubled again to investigate the effects of model complexity, and the configurations are summarized in Table 3.

The final model presented in this paper incorporates the findings and interpretations of the results obtained from both the first-round



ξ: Vector of type I parameters; NN: Neural network.

Fig. 3. PINN structures screened for LKM.

Table 2
Configurations of PINN model for first-round screening study.

Neural network		Design (A)	Design (B)	Design (C)			Design (D)			Design (E)					Design (F)				
Model ID		A1	B1	C1	C2	C3	D1	D2	D3	E1	E2	E3	E4	E5	F1	F2	F3	F4	F5
NN 1	No. of layers	6	6	3	3	3	3	3	3	3	3	3	2	4	1	1	1	1	1
	Neurons per layer	3	6	5	3	1	5	3	1	5	3	1	5	5	15	9	3	10	20
NN 2	No. of layers	6	–	3	3	3	3	3	3	3	3	3	2	4	3	3	3	2	4
	Neurons per layer	3	–	1	3	5	1	3	5	5	3	1	5	5	5	3	1	5	5
NN 3	No. of layers	–	–	3	3	3	3	3	3	3	3	3	2	4	3	3	3	2	4
	Neurons per layer	–	–	6	6	6	6	6	6	1	3	5	1	1	1	3	5	1	1
NN 4	No. of layers	–	–	–	–	–	–	–	–	3	3	3	4	2	3	3	3	4	2
	Neurons per layer	–	–	–	–	–	–	–	–	6	6	6	6	6	6	6	6	6	6

Table 3
Configurations of PINN model for second-round optimization.

Model ID		Model (E1-1)	Model (E1-2)	Model (E1-3)	Model (E1-4)	Model (E1-5)	Model (E1-6)
NN 1	No. of layers	3	3	3	3	3	3
	Neurons per layer	10	10	10	10	10	20
NN 2	No. of layers	3	3	3	3	3	3
	Neurons per layer	10	10	10	10	10	10
NN 3	No. of layers	3	3	3	3	3	3
	Neurons per layer	2	2	2	2	2	4
NN 4	No. of layers	3	3	3	3	3	3
	Neurons per layer	12	12	12	12	12	24

screening and second-round optimization. The structure of final model was optimized based on model (E1-6), which the neural networks (NN 1, NN 2, NN 3, and NN 4) have three layers each. To address the challenges posed by the wider time horizon and flow rate range, the neurons at each layer of NN 1, NN 2, NN 3 and NN 4 were increased to 80, 10, 16, and 96, respectively. This increase in the number of neurons enhances the model's capacity to learn complex relationships between the input variables and the output response.

3.2. Selection of data points for PINN model training and verification

The selection of training data points for Type I variables was based on the Design of Experiment (DoE) strategy. For the first-round screening, two Type I variables, ε_t and D_{ax}, are volatile and the response surface method (RSM) with nine points (four point at each corner, four points at each edge and one center point) was applied. The selection of training data points for Type II variables is different from that of Type I variables. There are 40 pure data points for Type II variables in the domain: 10 points distributed evenly at $x = 0$ and $x = L$, respectively (excluding $t = 0$, for B.C. 1 and B.C. 2); 8 points at $t = 0$ (two of them located at 33 % and 66 % of the x range, three of them distributed evenly at [95 %, 100 %] of the x range and three of them distributed evenly at [0 %, 5 %] of the x range, for I.C.); 12 points distributed randomly in domain (for PDEs). Similarly, there are 2000 grid points in the domain: 450 points distributed evenly at $x = 0$ and $x = L$, respectively; 100 points at $t = 0$ (50 of them distributed evenly at (5 %, 95 %) of the x range, 25 of them distributed evenly at [95 %, 100 %] of the x range and 25 of them distributed evenly at [0 %, 5 %] of the x range); 1000 points distributed randomly in domain. An example of these distributions is presented in Fig. S1 (supplementary material).

As to the testing data points for the first-round screening, based on the number of Type I variables involved, 5^2 data points were selected. For the Type II variables, 500 × 500 points were picked evenly in the design space of x and t for each condition of Type I variables.

For the second-round optimization, all eight Type I variables were involved in the DoE design, and a resolution-V fractional factorial design (with design generators $I = 12346$, $I = 12357$ and $I = 12458$) was applied with 2^5 corner points plus one center points (In total 33). In addition, due to limited computational capacity, an optimized LHS (Latin hypercube sampling) algorithm with maximin design was applied to fill the design space of the high dimensionality problem. This hybrid algorithm ensures the minimum distance between points while preserve a Latin square structure, which improve the possibility of adequately filling the design space than random sampling with the same given number of sampling points [25]. 167 LHS points were sampled along with the original 33 DoE points (in total 200) for the second-round optimization, and it was further increased to 367 LHS points to improve the performance.

The selection training data points of Type II variables for the second-round optimization has the same distribution as that of the first round, namely "Group A" data points. However, higher sampling density was applied to the group A data points used in second-round optimization to avoid the possibility of failing to capture the breakthrough process of the conditions of Type I variables that result in fast breakthrough. For pure data points, 6000, 250, 250 and 250 points were sampled for the domain, B.C.1, B.C.2 and I.C., respectively. For grid points, 114000, 4750, 4750 and 4750 points in addition of pure data points were sampled for the domain, B.C.1, B.C.2 and I.C., respectively.

An alternative approach was also adopted in second-round optimization for the selection of training data points of Type II variables, namely "Group B" data points. To better capture the transition region (where the response c or q varies from 0 to 1 in the space of t and x), this approach increases the sampling density in the transition region only instead of evenly or randomly in the domain, with a similar philosophy as Shi et al. have ever adopted [26]. The transition region is determined by the time (t) reaching different ratios of max mobile phase concentration (c) at each of n segments of the column. The right boundary of transition region corresponds the last segment that shows breakthrough (where $c > 0$) at t_{max}. For the pure data points, three points corresponding 5 %, 50 %, and 95 % of max breakthrough at nine segment position (In total 27) along with 256 uniformly distributed data points were sampled in this region, and 3 × 25 points were sampled for B.C.1, B.C.2 and I.C. As to the grid points, in addition to the 283 pure data points in domain, 3600 points distributed evenly in the transition region, 625 points distributed evenly and 1192 points distributed following LHS algorithm (In total 5700 points) were sampled in the domain of t and x, and 3 × 500 points were sampled for the B.C.1, B.C.2 and I.C. For Type I variables of this approach, 7467 LHS points were sampled along with the original 33 DoE points (in total 7500) given the same total number of grid points. For the final model training, it was further increased to 29967 for better performance. An example of the group B points distribution in the space of t and x is presented in Fig. S2

(supplementary material).

As to the testing data points for the second-round optimization and final model, 500 data points were sampled randomly with LHS for Type I variables with different categories to avoid duplications of training data points, and 50 × 5000 points were picked evenly in the design space of x and t for each combination of Type I variables. Table 4 gives a summary of the study design of second-round PINN model optimization.

3.3. PINN model training schemes

Different approaches have been introduced for the chromatography modeling to enhance the training of the PINNs. Santana et al. [19] proposed an adaptive schedule for the weights of column inlet residue loss term (namely B.C. 1) while Söderström [22] used completely pure simulated data for both column inlet and outlet. In this study, a hybrid of theirs was used which consists of weights modifications and provision of pure data points. For weights modifications, a weight of 0.1 for the B.C.1 was set and 5 % of pure data was provided for all the collocation points on the domain, boundary conditions and initial conditions for model training of the first-round screening and second-round optimization. For the final model training, the quadruple of training data and increase in model complexity causes extremely long training time. To accelerate the training process, the model was trained on the pure data points only for first few epochs as stage 1 to obtain ideal initial weights and bias for stage 2 training. Training on pure data does not involve the calculation for derivatives through auto-differentiation, which saves computation time. The stage 2 training still follows the approach used for the first-round screening and second-round optimization, which involves all loss functions presented above and follows the same weights modifications.

The training of PINNs involves the determination of multiple hyperparameters. For the optimizers, the Adam optimizer, a first-order stochastic gradient descent method, over the second-order L-BFGS method were chosen for better scalability and computational efficiency for neural networks with larger number of parameters [27]. For the learning rate of the Adam optimizer, the starting learning rate was set at 10^{-3} and decreased to 10^{-5} as the final learning rate with an exponential decay. An L2 regularization function was imposed to prevent overfitting [28] as well. For the first round and second round optimization, the training epochs are 5000 and the batch size used are 2^8 and 2^{19} respectively. For the final model, the training epochs for stage 1 and stage 2 are 4000 and 1000 respectively, and the batch size used is 2^{19}.

3.4. Parameter estimation for breakthrough curves

The parameter estimations are performed through both of CADET platform [29–32] and final LKM-PINN model on the measured breakthrough curves (where $x = L$). The parameter estimation for LKM involves determining the value of ε_t, D_{ax}, q_{max}, $k_a$and k_d as the rest of inputs are known.

For CADET, a parameter estimation and error modelling module called CADET-Match was used following the instructions. For LKM-PINN, the method of inverse mapping with traditional stochastic gradient descent (SGD) algorithm was adopted using finite difference to calculate the gradients. The loss function J is designed as the RMSE between the LKM-PINN outputs of given inputs X:

$$J = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\widehat{c}(t_i;X) - c_i\right)^2} \tag{16}$$

where $\widehat{c}(t_i;X)$ is the function between c and t when $x = L$ of LKM-PINN model with parameters X (X is $\{\varepsilon_t, L, u, D_{ax}, k_a, k_d, q_{\max}, c_0\}$ or its subset), c_i and t_i are the ith point ($i \in 1,\ 2,\ \ldots,\ N$) of measured data, and N is the total data points number involved in fitting.

The SGD algorithm updates the inputs value in each epoch through a learning rate of η and gradient between loss function J and inputs X:

$$X \leftarrow X - \eta\frac{\partial J}{\partial X} \tag{17}$$

where the gradient is calculated using the finite difference method with small perturbation value h:

$$J^{+} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\widehat{c}(t_i;X+h) - c_i\right)^2} \tag{18a}$$

$$J^{-} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\widehat{c}(t_i;X-h) - c_i\right)^2} \tag{18b}$$

$$\frac{\partial J}{\partial X} = \frac{J^{+} - J^{-}}{2h} \tag{19}$$

3.5. Measurement of breakthrough curves

Three breakthrough curves were measured to verify the performance of PINN model. The experiments were carried out with AKTA Pure M 150 device (Cytiva). The chromatography columns, Omnifit EZ with 0.66 cm inner diameter (Cat. # 006EZ-06-25-AA, Diba Industries) and HiScale 16/40 with 1.6 cm inner diameter (i.d.) (Cat. # 28964424, Cytiva), were packed with MabSelect PrismA resin (Cat. # 17-5498, Cytiva) at 18.0 cm and 12.8 cm bed height (h), respectively. The columns were equilibrated with 50 mM Tris-acetate and 150 mM NaCl

Table 4
Study design of the second-round PINN model optimization.

Item	Model (E1-1)	Model (E1-2)	Model (E1-3)	Model (E1-4)	Model (E1-5)	Model (E1-6)
Monotonic constraint	No	Yes	Yes	Yes	Yes	Yes
Total trainable neurons	72	72	72	72	72	144
Total training data points	27 million	27 million	30 million	54 million	54 million	54 million
Data points for Type I parameters	33 (DoE) + 167 (LHS)	33 (DoE) + 167 (LHS)	33 (DoE) + 367 (LHS)	33 (DoE) + 367 (LHS)	33 (DoE) + 7467 (LHS)	33 (DoE) + 7467 (LHS)
Pure data points in domain of Type II parameters	6000	6000	6000	6000	256	256
Pure data points on B.C. 1	250	250	250	250	25	25
Pure data points on B.C. 2	250	250	250	250	25	25
Pure data points on I.C.	250	250	250	250	25	25
Pure data points in transition region	NA	NA	NA	NA	27	27
Grid points in domain of Type II parameters	120000	120000	60000	120000	881 (Evenly) + 1192 (LHS)	881 (Evenly) + 1192 (LHS)
Grid points on B.C. 1	5000	5000	5000	5000	500	500
Grid points on B.C. 2	5000	5000	5000	5000	500	500
Grid points on I.C.	5000	5000	5000	5000	500	500
Grid points in transition region	NA	NA	NA	NA	3627	3627

buffer at pH 7.5 before loading the model protein, and regenerated with 1 M acetic acid followed by 0.1 M NaOH before the next cycle. The model protein used is a monoclonal antibody expressed by a CHO cell line, and titrated to pH 7.4 with a conductivity of 13.7 mS/cm before loading. Five breakthrough curves were measured and the loading flow rate were 180, 270, 540, 192 and 192 cm/h respectively. The first three breakthrough curves were measured with a loading concentration of 4.762 mg/mL, the fourth breakthrough curve with a loading concentration of 5.194 mg/mL, and the fifth breakthrough curve with a loading concentration of 2.541 mg/mL. The initial concentration and breakthrough concentration of model protein were measured based on the UV adsorption value at 280 nm wavelength with UV sensor of AKTA Pure M 150.

All solutions were filtrated with Millipore Express PLUS 0.22 μm PES membrane (Cat. # GPWP04700, Merck), and the loading sample was filtrated with Corning 1 L Bottle Top Filter 0.22 μm PES membrane (Cat. # 431174, Corning). The Tris(hydroxymethyl)aminomethane (Cat. # 4102), acetic acid (Cat. # 9526), sodium chloride (Cat. # 3627) were purchased from Avantor, and the sodium hydroxide was customized and purchased from Spectrum.

3.6. Implementation

All the numerical solutions were obtained via CADET platform [29–32] deployed in Python. All PINN models investigated are implemented using SciANN [33], a Keras/Tensorflow API for physics-informed machine learning. A server equipped with two Intel® Xeon® Gold 6342 CPUs and four NVIDIA A30 GPUs was used for model training and simulation.

4. Results

4.1. Effects of PINN structure

In the first-round screening, all six model PINN model structures have been compared. The structure (A) and (B) has only one form for each, and the model (A1) and (B1) were selected as a benchmark for structures (A) and (B), respectively. For structures (C) to (F), the best design was selected based on the loss (RMSE) of c, q and PDE equations (including I.C. and B.C.) in both training set and testing set, that is (C2), (D1), (E1) and (F1), respectively. The results of those models are summarized as shown in Fig. 4. The detailed results of all models, including

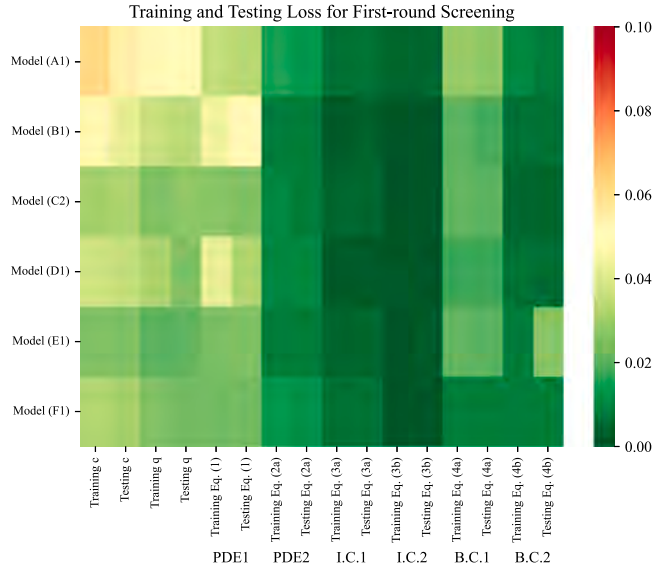


Fig. 4. Contour plot for the training and testing loss (RMSE) of the first-round PINN model structure screening.

(C1) – (C3), (D1) – (D3), (E1) – (E5), (F1) – (F5), are attached in Table S1 – S5 (supplementary material).

Though the effort of this research is to enhance the physical properties of the neural network, the neural network itself is still a "black-box". Introducing the PINN improved the learning target to reduce the biases of pure data model, and organizing the model structure would help to reduce the unnecessary confounding among different model parameters. Based on the results of loss terms on PDEs, all models are capable to learn the physics laws. As the benchmark, for the simple models (A1) and (B1), which make no differentiation for input variables, (B1) presents better performance than (A1) with lower loss in both training dataset and testing dataset. Specifically, the loss in Eq. (2a) of (B1) is much lower than that of (A1). The Eq. (2a) demonstrates the relationship between c and q. Compared to (B1), (A1) separates the output variables c and q into two different neural networks, and the connection between c and q is weaker. As a result, it presents worse performance in Eq. (2a), and it indicates that the model structure does have significant impact on the model performance as expected. For the application of PINN in LKM, which the output variables c and q are correlated, it is recommended to express them in one neural network.

As to the complex models (C2), (D1), (E1) and (F1), they were divided into at least two layers with one neural network for output variables and some neural networks for input variables. The relationship between equation loss and model structure for complex models is not as straightforward as the simple models (A1) and (B1). Overall, for the complex models, the output losses for c and q are all better than the simple models, and the structures E and F proposed in this paper have better performance than C and D. The inherent drawback of structures B, C and D is that the effects of Type I variables are actually fully confounded with that of Type II variables in NN 2. This confounding results that the model is not capable to distinguish their effects when different combination of Type I and II variables yield the same output. Though NN 3 was added to structure E and F to increase the model flexibility, which also increased the confounding between Type I and II variables, it is only a minor part compared to NN 2. Furthermore, based on the results that the optimal models for structure E and F are model E1 and F1 respectively, it can be concluded that better performance can be obtained with less neurons in NN 3. Also, it is not surprising that E1 has better performance than F1 with a depth structure in NN 1 instead of a flat structure, which is quite common in deep learning.

From the first-round screening data, model (E1) presents the best performance among all models. The loss of c and q are both lowest among all the model structures in training set as well as testing set. Meanwhile, the losses of physics equations are comparable to that of the other model structures. Thus, this model structure was selected as the optimal one for the second-round model optimization with all eight LKM model parameters.

4.2. Improvements with monotonic constraint

With regard to an on-column adsorption process, Eqs. (7a) to (7d) can be added to the PDE loss terms to improve the training performance. The results of the second-round model optimization studies with and without monotonic constraint are summarized in Fig. 5 (the raw data have been summarized in Table S6 (supplementary material)), where the model (E1-1) and model (E1-2) represent the design without and with the monotonic constraint, respectively. The model (E1-1) is a benchmark model trained based on the conclusions drawn from the first-round screening, and the detailed design was introduced in Section 3. Compared to the model (E1), similar training performance was obtained from model (E1-1). However, the performance on testing set is significantly worse than that of model (E1). The increase of dimension (unfixed Type I variables) is considered the most contributor to this result and consequently the main purpose of the second-round optimization is to improve the prediction performance of PINN on test set.

From the perspective of RMSE, adding monotonic constraint didn't

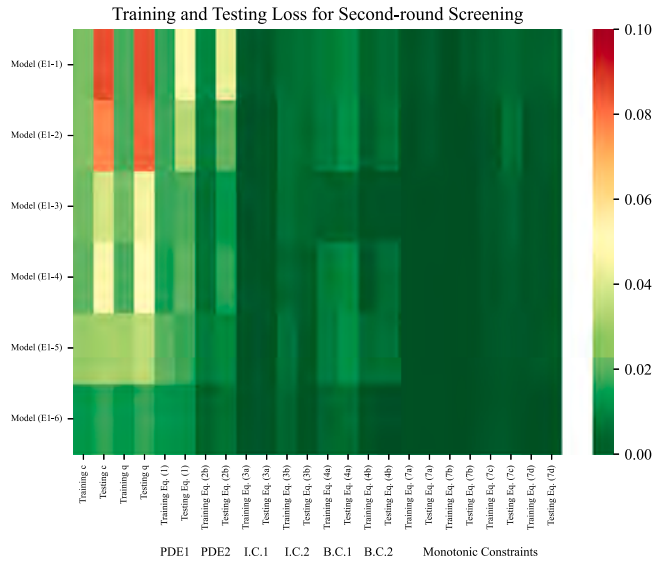


Fig. 5. Contour plot for the training and testing loss (RMSE) of the second-round PINN optimization.

improve the testing performance quite well, with minor decreasing in the loss of c and q. But adding monotonic constraint did help to control the overall trend of predicted values and overcome the problem that the training process would fall into some local optimal solutions during gradient descent. An example of the breakthrough curve ($x = L$) has been given in Fig. S3 (supplementary material) to demonstrate the effects of monotonic constraint on model performance. Thus, in following training procedure of PINN models, the monotonic constraint was added by default to avoid such kind of unexpected fluctuating.

4.3. Improvements with data density and distribution

The data set used for PINN model training was initially designed based on the knowledge obtained from the first-round screening, with 200 grid points for Type I variables and 135k grid points for Type II variables (27 million grid points in total, also defined as Group A data set in section 3). This kind of data set has a high density in Type II variables and low density in Type I variables, which works well in the first-round screening with only two Type I variables. However, with more unfixed Type I variables introduced in the second-round optimization, though the DoE strategy was applied to improve the data distribution, the results of model (E1-1) and model (E1-2) still didn't meet the expectations. A straightforward conclusion is that the limited data points for Type I variables is not sufficient to support the high dimension model, and then three approaches were verified to solve this problem and improve the model performance, including re-arranging the data points between Type I and Type II variables, increasing the density of data points for Type I variables and changing the overall sampling strategy. The results after these improvements are summarized in Fig. 5, which were denoted as model (E1-3), model (E1-4) and model (E1-5). The model (E1-3) simply reduced half of the grid data points for Type II variables, and increased 200 extra LHS points for Type I variables (Though there is an error that the grid points on B.C. 1, B.C. 2 and I.C. didn't reduce half of that in model (E1-2) accordingly, it doesn't impact the overall trend). Compared to the model (E1-2), with a consistent number of total grid points, model (E1-3) has significant improvement in the loss of c and q in testing set, which reduced about 50 %. Meanwhile, it was also tried in model (E1-4) to keep the data point density of Type II variables consistent to that of model (E1-1) and (E1-2), which actually added 200 LHS points for Type I variables and doubled the total grid points for training. But from the loss on training data set and testing data set, it can be found that the improvement is quite limited compared to model (E1-3). Based the results of model (E1-3) and (E1-4), it can be concluded that the bottleneck for PINN model at high dimension is the grid points for Type I variables, and further increase the data points for Type II variables didn't contribute to a better model performance.

Diving into detailed matrix of c and q in the space of t and x, it can be found that the matrix is quite "spare", and most of their values are 1 or 0 (red and green). An example of the responses c and q as a function of Type II variables x and t was plotted as shown in Fig. S4 (supplementary material). The critical part for the simulation of adsorption behavior, the transition range between 1 and 0, is typically very narrow. In this case, when a model was trained with grid points evenly or randomly selected from the whole space of t and x, the overall RMSE would significantly be "diluted" by the less critical area with solutions 1 and 0, and further increasing grid points would not increase the overall training performance as expected. To address this problem, an optimized model (E1-5) was trained using the "Group B" data that focused on the transition region as introduced in Section 3, and the results are shown in Fig. 5.

Trained with group B data, the model (E1-5) presents slightly worse performance on training set, but its performance between training set and testing set is quite consistent. With almost 20-fold less grid points in Type II variables, a scientific selection of points in the transition region gave a better performance on testing set than model (E1-3) and (E1-4) that trained with group A data. Also, though the transition region was defined based on the value of c, it helped the training of q at the same time. Thus, this approach is considered effective to further improve the model performance.

In further, the worst-case of model (E1-5) testing set that has the highest RMSE has been examined in detail to understand how far this model would be biased away from traditional numerical solutions. Figure 6 presents the contour plots of the mobile phase concentration obtained via CADET, (E1-5) and their differences under this condition. The best combination and worst combination for (E1-5), including both c and q, are summarized in Fig. S5 and Fig. S6 (supplementary material), respectively. From these results, it can be found that though the overall testing loss of model (E1-5) is considered low, the biases are still quite significant between the PINN solutions and CADET solutions. The notable red area in Fig. 6 (C) indicates that current PINN model (E1-5) will generate unexpected results in some conditions and not considered robust enough. Thus, the PINN model needs to be further optimized for applications.

4.4. Improvements with model complexity

For the neural network, model complexity is generally considered a major factor to the performance, but it is usually limited by the data source, computer hardware and training time. Unlike traditional pure data neural network model, which has great demands on training data set when model neurons increase, since training of PINN model doesn't have a strict requirement on the pure data, PINN has almost no limitation to increase the neurons from the perspective of data source. Thus, the model complexity was also investigated in the second-round model optimization study to figure out the potential of PINN model for LKM approximation.

Compared to the model (E1-5), a new model (E1-6) with doubled trainable neurons (20 neurons on each layer of NN 1, 4 neurons on each layer of NN 3 and 24 neurons on each layer of NN 4) has been trained, and the results are summarized in Fig. 5. From the RMSE alone, the extra neurons in model (E1-6) boosted the model performance to a new level, and the loss of c and q on testing set reached 0.016. With doubled model neurons and no additional training data introduced, the testing loss reduced to 47 %–53 % compared to model (E1-5). This almost linear performance improvement demonstrates the feasibility to develop a high precision LKM-PINN model for industrial applications.

To examine if the PINN model is practically applicable, the distribution of testing loss on every individual point in testing set (500 LHS points in total) was calculated for model (E1-5) and (E1-6), and

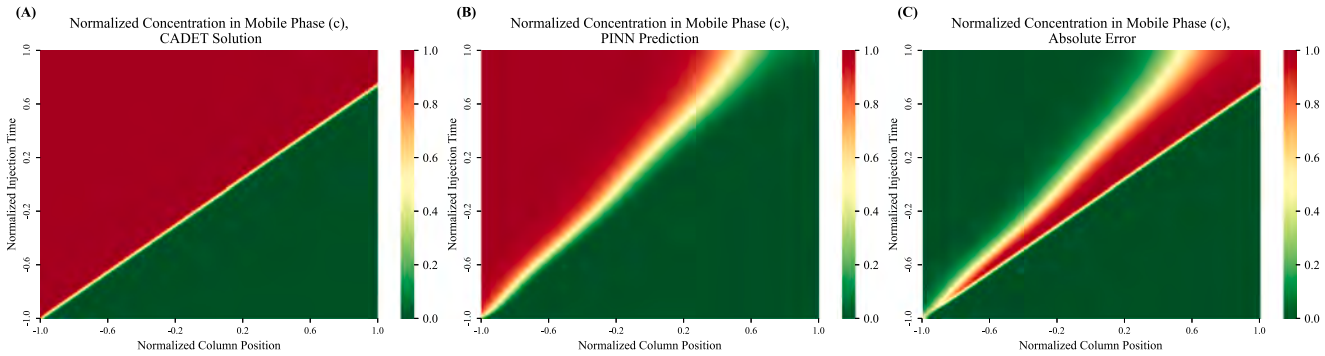


Fig. 6. Contour plot for mobile phase concentration obtained via (A) CADET, (B) model (E1-5) and (C) their absolute differences with the worst combination of Type I variables for model (E1-5) ($\varepsilon_t = 0.530$, $L = 0.141$ m, $u = 1.245$ m/h, $q_{max} = 11.317$ g/L resin, $k_a = 4.616 \times 10^{-2}$ L·g^{-1}·s^{-1}, $k_d = 7.814 \times 10^{-3}$ s^{-1}, $D_{ax} = 1.549 \times 10^{-9}$ m^2/s and $c_0 = 1.328$ g/L). The color scale represents the normalized values.

presented in Fig. 7. Besides the overall RMSE, the distribution of individual RMSE of (E1-6) is also closer to 0 than that of (E1-5). Meanwhile, the distribution for (E1-6) is also much narrow, and the highest RMSE for (E1-6) is only about 0.2 compared to about 0.7 for (E1-5). This result indicate that the model (E1-6) is more robust than model (E1-5).

The worst combination of Type I variables in testing set of model (E1-6), which yielded the highest RMSE, was evaluated in detail, and its prediction performance on mobile phase concentration c is summarized in Fig. 8 (The best combination and worst cobination for (E1-6), including both c and q, are summarized in Fig. S7 and Fig. S8 (supplementary material), respectively). The contour plots in Fig. 8 present the prediction results of CADET (A), model (E1-6) (B) and their absolute differences (C). It can be found that the solutions of (E1-6) are quite close to CADET solution with negligible shifts, compared to the notable gap of (E1-5) presented in Fig. 6. This result is consistent to the value of RMSE, and indicates that the model (E1-6) is capable to deliver a robust result as well as traditional numerical method like CADET.

Figure 5 presents all the loss terms to present how above strategies contributed to the model performance, including introducing monotonic constraint, increasing data density, adjusting data distribution and increasing model complexity. In summary, the monotonic constraint gives limited improvement on the model performance, but since it has no harm to the model training for adsorption behavior, it was still adopted in this research. Data density can improve the model performance in some extent, but the optimization on data distribution is more critical to the adsorption model which the distribution of effective data points mainly focus on the transition area of breakthrough. Also, a more complex model can significantly improve the model performance with compromising the calculating speed. Based on these results, the model (E1-6) was then used as the prototype to develop the final LKM-PINN model.

4.5. Final LKM-PINN model and applications in breakthrough curve prediction

Based on the design of prototype model (E1-6) introduced in Section 4.4, the final LKM-PINN model has been designed and trained on a wider parameters' range (described in Section 2.2) to meet the requirements of real industry cases, and the loss terms after training are summarized in Table 5. As expected, the final model has a rather low RMSE for both c and q. The RMSE in best parameters combination and worst parameters combination for the final model, including both c and q, are summarized in Fig. S9 and Fig. S10 (supplementary material), respectively. From the distribution data presented in Fig. 9, it can be found that all the RMSEs are below 0.1. Meanwhile, the RMSEs of every PDE equations and I.C. / B.C. are all considered low, which indicates that the final LKM-PINN model well satisfies the physics laws of LKM model.

To investigate the worst-case performance of the final model, the relationship between mobile phase concentration (c) and Type II variables under combination of Type I variables yielded the highest RMSE (0.092) was calculated and compared with numerical solutions obtained via CADET. Figure 10 presents the results of this condition (the performance of the best case and the impact on q have been summarized in the supplementary information). Though there are still some minor gaps,

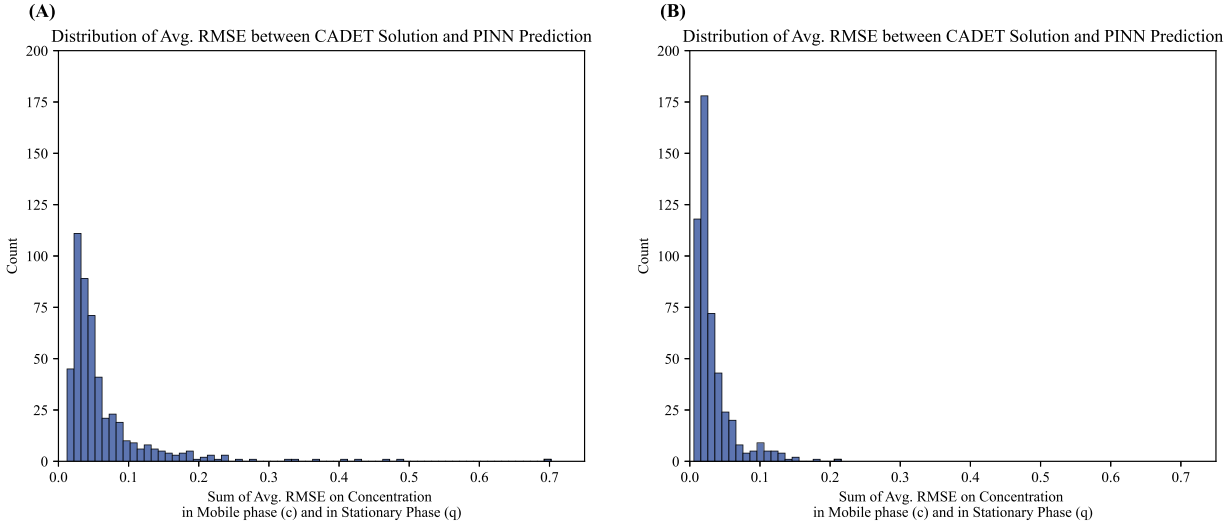


Fig. 7. Distribution of average RMSE on testing set of (A) model (E1-5) and (B) model (E1-6).

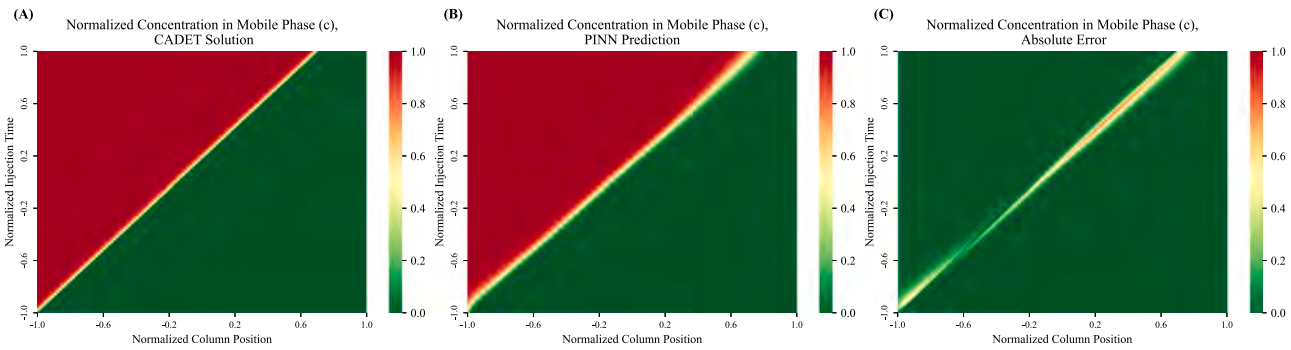


Fig. 8. Contour plot for mobile phase concentration obtained via (A) CADET, (B) model (E1-6) and (C) their absolute differences with the worst combination of Type I variables for model (E1-6) ($\varepsilon_t = 0.857$, $L = 0.239$ m, $u = 2.579$ m/h, $q_{max} = 173.326$ g/L resin, $k_a = 6.325 \times 10^{-1}$ L·g^{-1}·s^{-1}, $k_d = 8.065 \times 10^{-5}$ s^{-1}, $D_{ax} = 1.543 \times 10^{-6}$ m^2/s and $c_0 = 2.476$ g/L). The color scale represents the normalized values.

Table 5
RMSE of loss terms of final LKM-PINN model.

Dataset	c	q	Eq. (1)	Eq. (2b)	Eq. (3a)	Eq. (3b)	Eq. (4a)	Eq. (4b)
Training set	0.007	0.007	0.002	0.001	0.003	0.001	0.008	0.003
Testing set	0.006	0.007	0.002	0.001	0.003	0.001	0.011	0.003

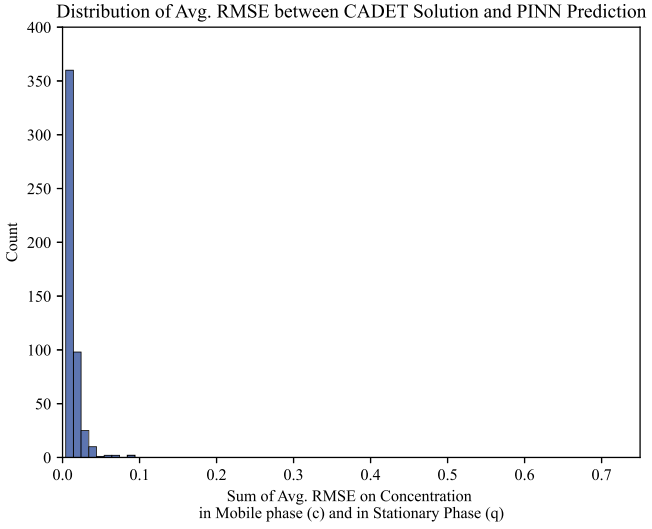


Fig. 9. Distribution of average RMSE on testing set of final LKM-PINN model.

the overall prediction between CADET and the final LKM-PINN model are quite consistent, and it demonstrates that the final LKM-PINN model is capable to give a similar prediction performance as numerical method.

The simulation performance of the final LKM-PINN model was investigated by fitting real experimental data measured at different flow rate (residence time), column size and loading concentration. The simulation results were further compared with the numerical solutions obtained through CADET. Figure 11 presents their fitting performance on the "known" data (The "known" data were used for LKM parameters estimation), and their predicting performance on the "unknown" data (The "unknown" data were not used for LKM parameters estimation). The upper limit of q_{max} was 200 g/L resin for LKM-PINN model, but it was found that the q_{max} fitted by CADET exceeded 200 g/L resin. Thus, another CADET fitting was carried out by capping the q_{max} within 200 g/L resin for better comparison with LKM-PINN. In both cases, the final LKM-PINN model present excellent performance for real experimental data fitting and predicting. From the perspective of RMSE, both the fitting and predicting performance of LKM-PINN model is comparable to the results obtained via CADET. The fitted model parameters summarized in Table 6 also indicate that the fitted parameters of two different methods are comparable.

As to the fitting time, when using the same initial guess (all parameters started from their low limits), the PINN model took 160 s to complete the data fitting, while CADET took 7 min to 72 min, depending

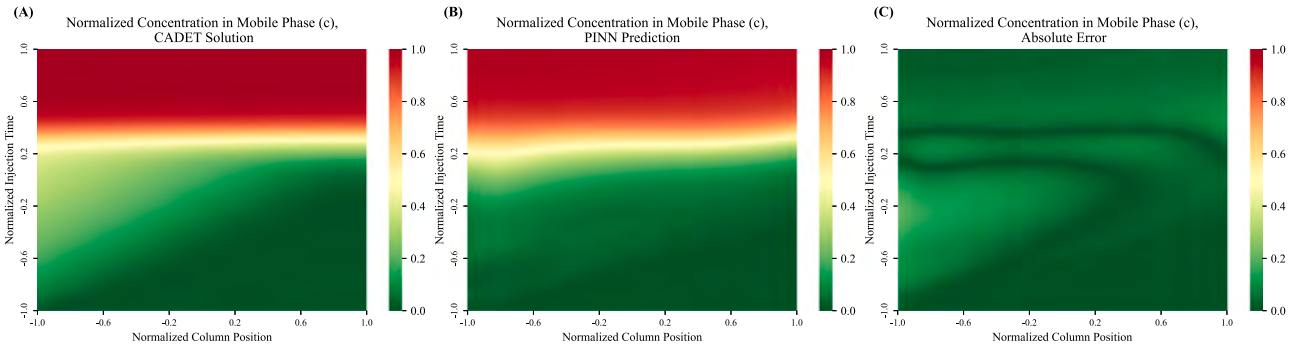


Fig. 10. Contour plot for mobile phase concentration obtained via (A) CADET, (B) the final PINN model and (C) their absolute differences with the worst combination of Type I variables for the final PINN model ($\varepsilon_t = 0.542$, $L = 0.250$ m, $u = 2.948$ m/h, $q_{max} = 149.023$ g/L resin, $k_a = 1.174 \times 10^{-1}$ L·g^{-1}·s^{-1}, $k_d = 1.283 \times 10^{-2}$ s^{-1}, $D_{ax} = 7.22 \times 10^{-4}$ m^2/s and $c_0 = 2.937$ g/L). The color scale represents the normalized values.

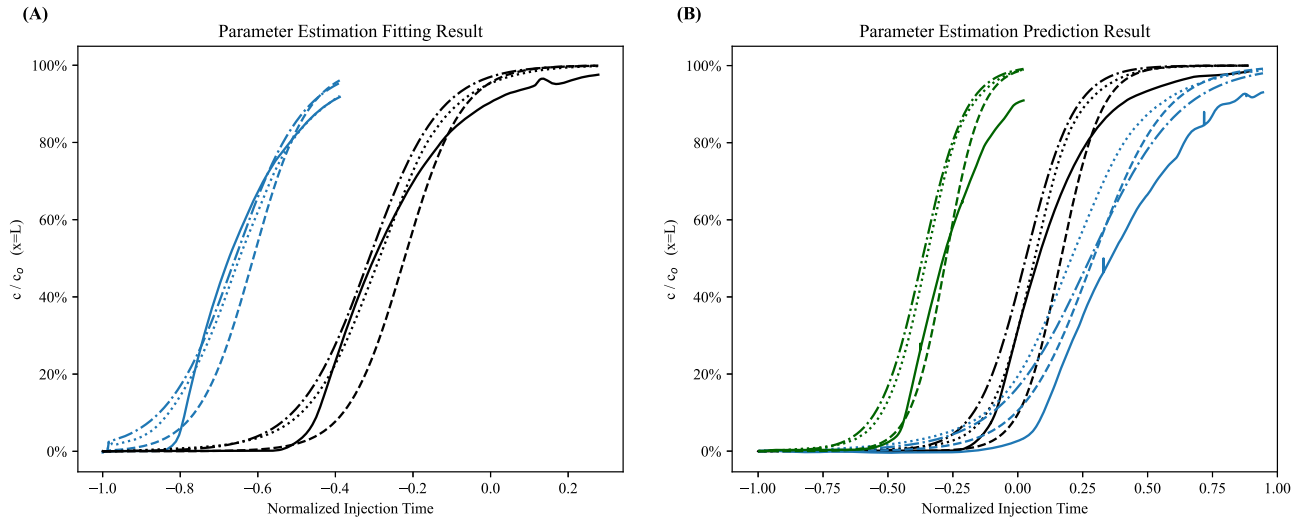


Fig. 11. Performance of LKM-PINN method and CADET method for (A) experimental data fitting and (B) prediction. Solid line: Experimental data; Dash line: Simulation data with CADET method; Dash dotted line: Simulation data with CADET method (q_{max} limited within 200 g/L resin); Dotted line: Predicted data with LKM-PINN method. Blue line: (A) $u = 5.4$ m/h, $c_0 = 4.762$ mg/mL, i.d. = 0.66 cm, $h = 18.0$ cm, (B) $u = 1.92$ m/h, $c_0 = 2.541$ mg/mL, i.d. = 1.6 cm, $h = 12.8$ cm; Black line: (A) $u = 2.7$ m/h, $c_0 = 4.762$ mg/mL, i.d. = 0.66 cm, $h = 18.0$ cm, (B) $u = 1.8$ m/h, $c_0 = 4.762$ mg/mL, i.d. = 0.66 cm, $h = 18.0$ cm; Green line: (B) $u = 1.92$ m/h, $c_0 = 5.194$ mg/mL, i.d. = 1.6 cm, $h = 12.8$ cm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Fitting results with CADET method and LKM-PINN method.

Fitted method	ε_t	D_{ax} [m^2/s]	k_a [$L \cdot g^{-1} \cdot s^{-1}$]	k_d [s^{-1}]	q_{max} [g/L resin]	RMSE (Run 1*)	RMSE (Run 2*)	RMSE (Run 3*)	RMSE (Run 4*)	RMSE (Run 5*)
CADET (CADET-Match)	0.643	2.79×10^{-8}	5.59×10^{-4}	4.51×10^{-4}	263	0.100	0.091	0.084	0.076	0.056
CADET (Limit q_{max}, CADET-Match)	0.645	1.00×10^{-9}	4.40×10^{-4}	1.00×10^{-5}	200	0.054	0.052	0.071	0.082	0.135
LKM-PINN (SGD)	0.599	2.48×10^{-8}	4.59×10^{-4}	2.18×10^{-4}	200	0.048	0.036	0.045	0.132	0.115

* Run 1: u = 5.4 m/s, $c_0 = 4.762$ mg/mL, i.d. = 0.66 cm, h = 18.0 cm. Run 2: u = 2.7 m/s, $c_0 = 4.762$ mg/mL, i.d. = 0.66 cm, h = 18.0 cm. Run 3: u = 1.8 m/s, $c_0 = 4.762$ mg/mL, i.d. = 0.66 cm, h = 18.0 cm. Run 4: = 1.92 m/s, $c_0 = 2.541$ mg/mL, i.d. = 1.6 cm, h = 12.8 cm. Run 5: u = 1.92 m/s, $c_0 = 5.194$ mg/mL, i.d. = 1.6 cm, h = 12.8 cm.

on the fitting settings. The fitting speed of PINN model was further improved to 30 s by using random values for all parameters as the initial guess. Based on the data above, the final LKM-PINN model can simulate the chromatography adsorption behavior with better prediction performance than traditional numerical method and complete the fitting within a reasonable duration. This feature makes it a useful tool for chromatography process real-time simulation.

5. Discussion

PINN as an emerging technology presents a potential to provide a high efficiency solution for the PDE problems. As there are quite limited publications reported nowadays for using PINN for chromatography problems, it is still quite challenging to construct an applicable PINN in this area. In this study, the application of a PINN for LKM has been investigated and the major factors impacting the development of PINN for LKM have been studied. Based on the experiences obtained in the two rounds of optimization and final model constructions, it was found that the application of PINN for chromatography problems is not as straightforward as its concept, and a systematical optimization is mandatory to get a high performance PINN.

Specifically, for the application in adsorption behavior discussed in this research, it was found that the input variables, which were classified into two categories based on their functions, are suggested to be decoupled to reduce the confounding between them. Thus, an in-series structure is recommended to handle the Type I variables and Type II variables separately. However, a rigid decoupling would also increase the difficulty for model training on the other hand, and it is recommended to add Type II variables to the secondary layer of the neural network of Type I variables to improve the flexibility, which is proven to benefit the model performance. Unlike the input variables, the output variables of LKM do have inherent connection, and using one neural network with strong connection could meet the requirement of physics laws of LKM. Consequently, it is strongly recommended that the structure of PINN should be carefully investigated and the underlying physics laws should be considered case-by-case.

As the PINN can use the "grid points" to train the neural network, the requirement in pure data is extremely low compared to regular artificial neural network trained with experimental data. However, the introducing of certain pure data points (e.g. 5 %) can still improve the convergence rate for PINN instead of using grid points alone. In real cases, the number of total data points introduced for model training is typically limited by the computer hardware (GPU memory). Also, to handle more data for model training, longer time is required to get convergence, which is also determined by the hardware (GPU frequency). Thus, it is suggested to optimize the distribution of grid points to fully utilize the resources. In this research, to simulate the adsorption process, the grid points were mainly distributed into the transition region, which was related to the breakthrough of protein concentration. Definitely, for different applications, the distribution of data points

should be considered in different ways.

The low requirement in experimental data makes it possible to increasing the model complexity (number of neurons) and/or using large scale AI model for PINN applications, and it does help to improve the model performance significantly. However, it was found that the increasing of model complexity not only increased the time required for model training, but also increased the time required for inference (data not shown). As inference speed is the main advantage of PINN compared to numerical methods, and also quite important for real time simulation, though the model complexity was not systematically studied in this research, it is a key factor and suggested to be optimized to balance the inference speed and model performance.

When the optimized LKM-PINN was used for breakthrough prediction, with the same column size and loading concentration, by changing the loading flow rate, the LKM-PINN model presents better performance than numerical method CADET. The normalized prediction RMSE for the loading flow rate of 1.8 m/s was 0.045, which was much lower than that predicted with CADET with both limited or unlimited q_{max} (0.071 and 0.084, respectively). However, the results turned to be slightly worst when transferred to a difference column size or a different loading concentration, where it could be found that the RMSE of PINN was close to or higher than the RMSE of CADET. The different performance would be contributed to three factors. The first one is that the impact of column packing. As two different columns were used, though the column size was considered, the porosity would also be impacted by column packing. This problem would be solved by adding additional experiments to test the column porosity, but it will increase the workload and cost for wet run, which is a kind of trade off. The second one would be the selection of training data. As the data used for LKM parameters estimation are with the same column size and loading concentration, it would not be sufficient for the model to capture their variance and provide a reliable result. This problem could be solved by adjusting the training data used for parameters estimation. The third factor would be the training procedure when fitting the LKM parameters, and it is expected to obtain better performance if the fitting algorithm could be optimized.

Compared to the PINN reported in other literatures for the applications in chromatography, this research provided a more comprehensive method for PINN development, and a versatile LKM-PINN model involving all LKM parameters in PINN. Unlike other PINN models reported in the literatures which were trained for specific material or process conditions, with all model parameters being involved in the PINN, the trained PINN in this research can be used as a surrogate solution for numerical method in different process conditions with different chromatography resin for different product.

6. Conclusions

A detailed methodology was provided for implementing the PINN for chromatography process simulation with the LKM. A hybrid approach with both numerical solutions and grid points was applied to improve the training of PINN. Different model structures of PINN were firstly screened as the foundation, and the in-series structure proposed in this paper has the best performance and was selected. Then the PINN model performance was improved from several aspects: adding a monotonic constraint, increasing data density, adjusting data points distribution and increasing model complexity. After these improvements, a final LKM-PINN model was trained for practical use. The final LKM-PINN model was used in the prediction of the breakthrough curve of chromatography process successfully, and a better performance was obtained compared to traditional numerical method solved via CADET.

The results indicate that the PINN is capable to be used for the simulation of chromatography process, and would be more suitable for real-time simulation and digital twin compared to numerical methods. It's inherent features that satisfying physics laws and having analytical expression make it an excellent tool to capture the process information and make decisions in real-time for a dynamic control of industrial chromatography process.

Though the proof-of-concept has been demonstrated in this study successfully, there are still some improvements that can be investigated in future. Firstly, The PINN model developed in this study was based on LKM, and the mass transfer in resin particle was not considered. Thus, there are still some minor gaps between the experimental data and model fitting data. Typically, a general rate model (GRM) would have better performance than LKM, but the additional dimension on resin particle radius and more model parameters would increase the difficulty for model training. Meanwhile, though the model complexity can be increased to improve the model performance, the model complexity also significantly impacted the calculation speed when it was used for data fitting and prediction. Thus, an optimal model complexity should be developed to balance the model performance and calculation speed. Finally, the algorithm used for LKM-PINN model fitting is a very preliminary SGD without any optimization. It is expected that the fitting algorithm can be further optimized to improve the fitting performance. In all, it is feasible and efficient to use PINN model for process simulation and digital twin in place of numerical methods.

CRediT authorship contribution statement

Si-Yuan Tang: Conceptualization, Methodology, Software, Investigation, Writing – original draft, Project administration. **Yun-Hao Yuan:** Software, Data curation, Visualization. **Yu-Cheng Chen:** Writing – review & editing. **Shan-Jing Yao:** Writing – review & editing. **Ying Wang:** Writing – review & editing, Supervision. **Dong-Qiang Lin:** Writing – review & editing, Supervision.

Declaration of Competing Interest

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

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Key R&D Program of China (2021YFE0113300) and National Natural Science Foundation of China (22078286).

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.chroma.2023.464346.

References

[1] C. Shi, S. Vogg, D.Q. Lin, M. Sponchioni, M. Morbidelli, Analysis and optimal design of batch and two-column continuous chromatographic frontal processes for monoclonal antibody purification, Biotechnol. Bioeng. 118 (2021) 3420–3434, https://doi.org/10.1002/bit.27763.

[2] V. Kumar, S. Leweke, W. Heymann, E. von Lieres, F. Schlegel, K. Westerberg, A. M. Lenhoff, Robust mechanistic modeling of protein ion-exchange chromatography, J. Chromatogr. A 1660 (2021), 462669, https://doi.org/10.1016/j.chroma.2021.462669.

[3] G. Sandoval, B.A. Andrews, J.A. Asenjo, Elution relationships to model affinity chromatography using a general rate model, J. Mol. Recognit. 25 (2012) 571–579, https://doi.org/10.1002/jmr.2223.

[4] C. Shi, Z.Y. Gao, Q.L. Zhang, S.J. Yao, N.K.H. Slater, D.Q. Lin, Model-based process development of continuous chromatography for antibody capture: a case study with twin-column system, J. Chromatogr. A 1619 (2020), 460936, https://doi.org/10.1016/j.chroma.2020.460936.

[5] P. Deulgaonkar, R. Bhambure, B. Prasad, A. Mishra, S. Tiwari, R. Mody, Mechanistic modeling of continuous capture step purification of biosimilar

monoclonal antibody therapeutic, J. Chem. Technol. Biotechnol. 97 (2022) 2404–2419, https://doi.org/10.1002/jctb.6922.
[6] C. Shi, Q.L. Zhang, B. Jiao, X.J. Chen, R. Chen, W. Gong, S.J. Yao, D.Q. Lin, Process development and optimization of continuous capture with three-column periodic counter-current chromatography, Biotechnol. Bioeng. 118 (2021) 3313–3322, https://doi.org/10.1002/bit.27689.
[7] Y.N. Sun, C. Shi, Q.L. Zhang, S.J. Yao, N.K.H. Slater, D.Q. Lin, Model-based process development and evaluation of twin-column continuous capture processes with Protein A affinity resin, J. Chromatogr. A 1625 (2020), 461300, https://doi.org/10.1016/j.chroma.2020.461300.
[8] R. Chen, X.J. Chen, C. Shi, B. Jiao, Y. Shi, B. Yao, D.Q. Lin, W. Gong, S. Hsu, Converting a mAb downstream process from batch to continuous using process modeling and process analytical technology, Biotechnol. J. 17 (2022), 2100351, https://doi.org/10.1002/biot.202100351.
[9] F.L. Vetter, J. Strube, Need for a next generation of chromatography models—Academic demands for thermodynamic consistency and industrial requirements in everyday project work, Processes 10 (2022) 715, https://doi.org/10.3390/pr10040715.
[10] D.Q. Lin, Q.L. Zhang, S.J. Yao, Model-assisted approaches for continuous chromatography: current situation and challenges, J. Chromatogr. A 1637 (2021), 461855, https://doi.org/10.1016/j.chroma.2020.461855.
[11] D.Q. Lin, C. Shi, S.J. Yao, Method for realizing multi-column continuous flow chromatography design and analysis, US Patent US20220381751A1 (2019).
[12] H. Narayanan, M. Luna, M. Sokolov, P. Arosio, A. Butté, M. Morbidelli, Hybrid models based on machine learning and an increasing degree of process knowledge: application to capture chromatographic step, Ind. Eng. Chem. Res. 60 (2021) 10466–10478, https://doi.org/10.1021/acs.iecr.1c01317.
[13] L. Lu, X. Meng, Z. Mao, G.E. Karniadakis, DeepXDE: a deep learning library for solving differential equations, SIAM Rev. 63 (2021) 208–228, https://doi.org/10.1137/19M1274067.
[14] J. Yu, L. Lu, X.H. Meng, G.E. Karniadakis, Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems, Comput. Methods Appl. Mech. Eng. 393 (2022), 114823, https://doi.org/10.1016/j.cma.2022.114823.
[15] L. Lu, R. Pestourie, W. Yao, Z. Wang, F. Verdugo, S.G. Johnson, Physics-informed neural networks with hard constraints for inverse design, SIAM J. Sci. Comput. 43 (2021) B1105–B1132, https://doi.org/10.1137/21M1397908.
[16] M. Mouellef, F.L. Vetter, J. Strube, Benefits and limitations of artificial neural networks in process chromatography design and operation, Processes 11 (2023) 1115, https://doi.org/10.3390/pr11041115.
[17] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, Nature Rev. Phys. 3 (2021) 422–440, https://doi.org/10.1038/s42254-021-00314-5.
[18] S. Cuomo, V.S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, F. Piccialli, Scientific machine learning through physics–informed neural networks: where we are and what's next, J. Sci. Comput. 92 (2022) 88, https://doi.org/10.1007/s10915-022-01939-z.
[19] V.V. Santana, M.S. Gama, J.M. Loureiro, A.E. Rodrigues, A.M. Ribeiro, F. W. Tavares, A.G. Barreto, I.B.R. Nogueira, A first approach towards adsorption-oriented physics-informed neural networks: monoclonal antibody adsorption performance on an ion-exchange column as a case study, ChemEngineering 6 (2022) 21, https://doi.org/10.3390/chemengineering6020021.
[20] S.F. Wang, X.L. Yu, P. Perdikaris, When and why PINNs fail to train: a neural tangent kernel perspective, J. Comput. Phys. 449 (2022), 110768, https://doi.org/10.1016/j.jcp.2021.110768.
[21] S.G. Subraveti, Z. Li, V. Prasad, A. Rajendran, Can a computer "learn" nonlinear chromatography?: Physics-based deep neural networks for simulation and optimization of chromatographic processes, J. Chromatogr. A 1672 (2022), 463037, https://doi.org/10.1016/j.chroma.2022.463037.
[22] P. Söderström, Physics-informed neural networks for liquid chromatography, Master's thesis, UMEÅ University, DiVA, 2022.
[23] P. Ramachandran, B. Zoph, Q. V. Le, Searching for activation functions, (2017). doi:10.48550/arXiv.1710.05941.
[24] S. Mishra, R. Molinaro, Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs, IMA J. Numer. Anal. 42 (2022) 981–1022, https://doi.org/10.1093/imanum/drab032.
[25] R.B. Gramacy, Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences, first ed., Chapman and Hall/CRC, New York, 2020 https://doi.org/10.1201/9780367815493.
[26] C. Shi, Model-assisted process development of continuous chromatography and its applications for antibody separation, Zhejiang University, CNKI, 2022, https://doi.org/10.27461/d.cnki.gzjdx.2021.001718.
[27] E. Haghighat, D. Amini, R. Juanes, Physics-informed neural network simulation of multiphase poroelasticity using stress-split sequential training, Comput. Methods Appl. Mech. Eng. 397 (2022), 115141, https://doi.org/10.1016/j.cma.2022.115141.
[28] G. Nuti, A.-I. Cross, P. Rindler, Evidence-based regularization for neural networks, Mach. Learn. Knowl. Extraction 4 (2022) 1011–1023, https://doi.org/10.3390/make4040051.
[29] A. Püttmann, S. Schnittert, U. Naumann, E. von Lieres, Fast and accurate parameter sensitivities for the general rate model of column liquid chromatography, Comput. Chem. Eng. 56 (2013) 46–57, https://doi.org/10.1016/j.compchemeng.2013.04.021.
[30] A. Püttmann, S. Schnittert, S. Leweke, E. von Lieres, Utilizing algorithmic differentiation to efficiently compute chromatograms and parameter sensitivities, Chem. Eng. Sci. 139 (2016) 152–162, https://doi.org/10.1016/j.ces.2015.08.050.
[31] S. Leweke, E. von Lieres, Chromatography analysis and design toolkit (CADET), Comput. Chem. Eng. 113 (2018) 274–294, https://doi.org/10.1016/j.compchemeng.2018.02.025.
[32] E. von Lieres, J. Andersson, A fast and accurate solver for the general rate model of column liquid chromatography, Comput. Chem. Eng. 34 (2010) 1180–1191, https://doi.org/10.1016/j.compchemeng.2010.03.008.
[33] E. Haghighat, R. Juanes, SciANN: a Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, Comput. Methods Appl. Mech. Eng. 373 (2021), 113552, https://doi.org/10.1016/j.cma.2020.113552.