Engineering 39 (2024) 84–93

Contents lists available at ScienceDirect

Engineering

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

Research
AI in Chemical Engineering—Article

A Case Study Applying Mesoscience to Deep Learning

Li Guo [a,b,*], Fanyong Meng [a], Pengfei Qin [a,b], Zhaojie Xia [a,b], Qi Chang [a], Jianhua Chen [a], Jinghai Li [a]

[a] *State Key Laboratory of Multiphase Complex Systems, Institute of Process Engineering, Chinese Academy of Sciences, Beijing 100190, China*
[b] *School of Chemical Engineering, University of Chinese Academy of Sciences, Beijing 100049, China*

ARTICLE INFO

Article history:
Received 29 April 2023
Revised 24 December 2023
Accepted 12 January 2024
Available online 19 January 2024

Keywords:
Mesoscience
Deep learning
Complex system
Gas–solid system
Bubbling bed

ABSTRACT

In this paper, we propose mesoscience-guided deep learning (MGDL), a deep learning modeling approach guided by mesoscience, to study complex systems. When establishing sample dataset based on the same system evolution data, different from the operation of conventional deep learning method, MGDL introduces the treatment of the dominant mechanisms of complex system and interactions between them according to the principle of compromise in competition (CIC) in mesoscience. Mesoscience constraints are then integrated into the loss function to guide the deep learning training. Two methods are proposed for the addition of mesoscience constraints. The physical interpretability of the model-training process is improved by MGDL because guidance and constraints based on physical principles are provided. MGDL was evaluated using a bubbling bed modeling case and compared with traditional techniques. With a much smaller training dataset, the results indicate that mesoscience-constraint-based model training has distinct advantages in terms of convergence stability and prediction accuracy, and it can be widely applied to various neural network configurations. The MGDL approach proposed in this paper is a novel method for utilizing the physical background information during deep learning model training. Further exploration of MGDL will be continued in the future.

1. Deep learning modeling methods using physical mechanisms

Complex systems typically exhibit nonlinear evolution and multilevel structures, with each level being made up of a large number of interacting components. Modeling for a specific complex system using only a data-driven approach based on mathematics and statistics—without considering the influence of the physical mechanisms embedded in the system—can result in drawbacks, such as the need for a large training dataset, a lengthy iteration time, humble model-generalization capability, and inadequate interpretability of the model and modeling process. Many researchers have therefore attempted to incorporate physical knowledge of the system into data-driven modeling [1]. The most well-known method is the physics-informed neural networks (PINNs) proposed by Raissi et al. [2]. PINN focuses on solving partial differential equations (PDEs) using neural networks and has been applied in several fields due to the universality of the method and its relative ease of implementation. Examples include exploring the impact of a rugby player's mouth guard on the brain [3], the nonlinear dynamics of optical fibers [4], the stress-transfer mechanisms in single-turn joints [5], and high-speed fluid modeling [6]. Researchers have also investigated the combination of PINN with other learning strategies, including transfer learning [7] and meta-learning [8].

In a PINN, the residuals of the system's governing PDEs and the initial value/boundary conditions of the PDEs are integrated into the loss function using differential operators, which is equivalent to integrating the constraints of the PDEs into the training process of the neural network. Thus, the resulting model satisfies the constraints of the physical laws represented by the governing PDEs, thereby mitigating the aforementioned problems to a certain extent. However, this treatment still requires attention to be paid to the accuracy of the PDEs. The PDEs are not always reliable due to the averaging process (e.g., in cases that involve handling turbulence and multiphase flows).

PINNs can be applied in systems in which mathematical equations with strong correlations between the different physical quantities have been established. However, PINNs cannot work if

[*] Corresponding author.
E-mail address: lguo@ipe.ac.cn (L. Guo).

https://doi.org/10.1016/j.eng.2024.01.007
2095-8099/© 2024 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company.

① such equations among the key physical quantities of the system have not been established, and ② it has been found that the intrinsic mechanism of the system is determined by other relationships between the physical quantities. To model such systems, novel methods must be developed.

Recently, mesoscience [9,10] has been proposed as a methodology for coping with multilevel complexities. This approach focuses on the study of mesoscale problems at different levels and correlates the macroscale behavior and intrinsic mechanisms of a system by means of the principle of compromise in competition (CIC) between dominant mechanisms. Mesoscience has been defined as "the science of the universality of mesoscale phenomena at different levels" or, more simply, as "the science of the in-between" [9]. To promote the development and application of mesoscience, the International Panel of Mesoscience (IPM) was established in 2018.

When solving a complex problem, mesoscience involves identifying the level, analyzing the dominant mechanisms and their compromise, determining their mathematical formulation, and finally solving a multi-objective variational (MOV) problem. For example, suppose that there are two dominant mechanisms in a system, A and B, where A tends to be the maximum and B tends to be the minimum. A and B compete with and restrain each other as time goes by and eventually reach a mutual compromise by which neither A nor B may always achieve their own objectives, but the system reaches stability in the end. Mesoscience suggests that the CIC between A and B is the essential driving force of the evolution of this complex system. A mesoscience framework can be applied to various disciplinary fields [11]. Based on this premise, it has been proposed that mesoscience theory can be incorporated into the process of deep learning modeling for complex systems [12]. This work illustrates how the concept of mesoscience theory can be implemented to guide deep learning modeling.

This paper proposes a novel deep learning modeling method called mesoscience-guided deep learning (MGDL). Analogous to a PINN, MGDL incorporates the physical information of the system into the modeling process. Unlike a PINN, which utilizes the residuals of the system's governing equations with the convergence criterion that the absolute values of the residuals tend to zero, MGDL employs an alternative strategy. It first identifies the dominant mechanisms of the system and their CIC behavior based on mesoscience; then, the CIC between dominant mechanisms is used as a critical reference for deep learning model building.

2. An MGDL modeling approach

According to mesoscience, although the dominant mechanism varies, complex systems all adhere to the same principle of CIC to build the system's dynamic structure. This complexity can be treated as a mathematical MOV problem to be solved [9] by analyzing the dominant mechanisms and the CIC between them. For example, in gas–solid two-phase flow systems, particle clusters are important mesoscale structures that have a significant impact on the system's mass and heat transfer. The energy-minimization multiscale (EMMS) method [13] employs this mesoscale structure to investigate the CIC between two dominant mechanisms, and it has been proved to be quite effective.

The principle of mesoscience has been successfully applied to complex systems in the chemical industry, such as gas–solid two-phase flow [14], turbulent tube flow [15], and heterogeneous catalysis [16], and is gradually being applied to complex systems in a variety of other fields [17]. In this paper, we propose the MGDL method, which integrates the dominant mechanisms of a system's spatiotemporal variation into the model training process by adding mesoscience constraints to the loss function; in this way, the proposed method improves the physical interpretability of the model-training process and extends the application of the physical-knowledge-informed modeling method to systems in which the variables have not yet been mathematically correlated very rigorously. We then validate MGDL using the example of a bubbling fluidized bed, which is a typical gas–solid system.

2.1. A sample dataset-building method with embedded spatiotemporal information

Before applying mesoscience concepts to deep learning model training, it is necessary to collect system evolution data containing the aforementioned dominant mechanisms. This paper proposes a method for constructing a sample dataset with embedded spatiotemporal properties.

Many machine learning techniques, including deep learning, can be conceptualized as optimization problems. In general, the construction of the majority of these machine learning models can be summed up as follows: First, based on an analysis of the problem, an optimization objective is determined (e.g., minimizing the mean squared difference between the predicted and actual values); then, the model parameters are iteratively updated using various optimization algorithms to gradually approach the optimization objective. A classical optimization problem consists of the objective function of the vector of the decision variables, as well as a number of constraints. The objective of the optimization is to find the optimal solution for the vector of the decision variables, such that the objective function is minimized in the decision variables' domain of definition. On the vector of the decision variables, the constraints are divided into two categories: equation constraints and inequality constraints.

From a mesoscience perspective, in comparison, the process of reaching the steady state of a system can be viewed as a physical system optimization process: The constraints may not be equations or inequalities directly related to the input and output variables but may rather be trend constraints after the variables' functional transformation (i.e., the application of physical principles). The MGDL modeling approach proposed in this paper is based on the numerical simulation of complex systems. A formal expression of the supervised learning sample data-building method with embedded spatiotemporal information is given in Eqs. (1)–(4) below.

$$\left[\left(x_1, x_2, \ldots, x_{m_x}\right) \Rightarrow \left(y_1, y_2, \ldots, y_{m_y}\right)\right]_{s \sim s+(T-1)} \tag{1}$$

$$\left(g_{i,s}, g_{i,s+1}, \ldots, g_{i,s+(T-1)}\right) \to \max, \quad i = 1, 2, \ldots, m_g \tag{2}$$

$$\left(h_{i,s}, h_{i,s+1}, \ldots, h_{i,s+(T-1)}\right) \to \min, \quad i = 1, 2, \ldots, m_h \tag{3}$$

$$\mathrm{CIC}_i\left(l_{i,1}, l_{i,2}\right), \quad i = 1, 2, \ldots, m_{\mathrm{CIC}} \tag{4}$$

where x_i is the ith input physical quantity; m_x is the total number of input physical quantities; y_i is the ith physical quantity to be predicted; m_y is the total number of physical quantities to be predicted; $g_{i,j}$ is the data in time step j of the ith dominant mechanism, which tends to be a maximum; m_g is the total number of dominant mechanisms tending to be a maximum; $h_{i,j}$ is the data in time step j of the ith dominant mechanism, which tends to be a minimum; m_h is the total number of dominant mechanisms tending to be a minimum; CIC_i is the ith CIC relationship; $l_{i,1}$, $l_{i,2}$ are the ith dominant mechanism pair with a CIC relationship; m_{CIC} is the total number of CICs.

Using Eqs. (1)–(4), a data sample is built, where the starting simulation time step is s and the time window width is T. Eq. (1) generates a data sample for general supervised learning, while Eqs. (2) and (3) are system time-series data with trends, the correlations of which are defined by Eq. (4). More specifically, Eqs. (2) and (3) are the simulation data of the system's dominant mechanism, classified based on evolutionary trends in the time window from s to $s+(T-1)$, where Eq. (2) is for the maximum trend and Eq. (3) is for the minimum trend. Eq. (4) globally describes the CIC relationships between the system's dominant mechanisms.

The application of constraints in Eqs. (2)–(4) differs from a standard optimization problem. In a classical optimization, all constraints (both equations and inequalities) are rigid; that is, all constraints must be satisfied by the optimization solution. However, constraints based on mesoscience are not necessarily rigid at a particular time, while the temporal trends of the dominant mechanisms must be examined. In other words, these constraints satisfy the CICs between different dominant mechanisms based on the principles of mesoscience and describe the dynamic properties of the system [18]. Each constraint competes with and restrains the others as time progresses and eventually reaches a mutual compromise in a particular state; the constraints may not always achieve their own objectives, but the system reaches stability in the end.

2.2. MGDL model training

The MGDL model-training method proposed in this paper adds mesoscience-based constraints to the error back-propagation of the network training—that is, it adds an additional item to the original loss function in order to reflect whether the current data state is consistent with the mesoscience principles. More specifically, in the model-training process, the time-series data defined by Eqs. (2) and (3) is analyzed to determine how well the current data conforms to the trend of each dominant mechanism and whether the CICs defined in Eq. (4) are satisfied. Fig. 1 depicts the training framework for the MGDL modeling.

The standard loss used in artificial neural network (ANN) or deep neural network modeling is referred to as the "mathematical loss" or "regular loss," whereas the results calculated using mesoscience principles are referred to as "meso-constraints." The two are combined to update the network weight parameters. Consequently, two crucial issues must be resolved: how to calculate the meso-constraints and how to integrate them with the regular loss. Various methods have been proposed for solving multi-objective optimization in mesoscience [17]; for example, Zhang et al. [19] developed a mathematical equivalent solution for the multi-objective optimization problem of the EMMS model for gas–solid fluidization. In the present work, a straightforward approach is adopted: examining the evolution trend of each system's dominant mechanism and degree of compliance with CIC, then weighting and summing them up as meso-constraints for deep learning modeling.

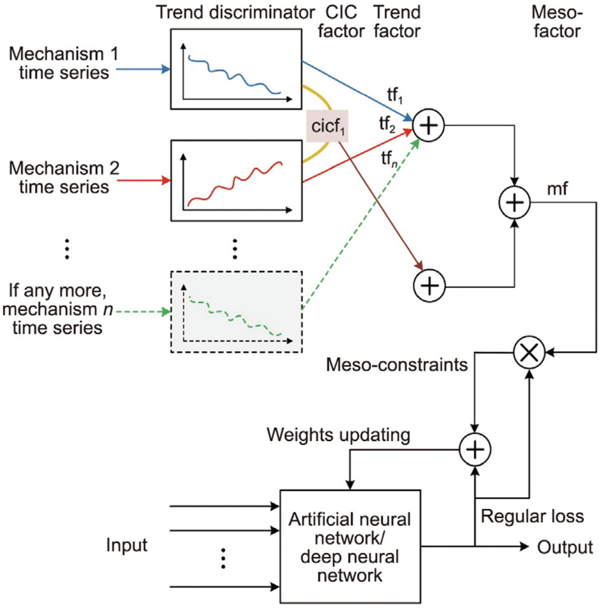

Fig. 1. Model-training framework for MGDL.

In Fig. 1, for a given time t, tf_i is the trend factor of the ith dominant mechanism (for one of the dominant mechanisms listed in Eqs. (2) and (3)). It is used to quantify the degree to which the trend at time t of this dominant mechanism coincides with the trend currently indicated by mesoscience. $\mathrm{tf}_i \in [-1.0, 1.0]$ is positive when the trend matches and negative otherwise. Various methods, such as linear regression, the Mann–Kendall trend test [20,21], and the Cox–Stuart trend method [22], can be utilized to determine the trend of fluctuation in a data series within a given time window.

In Fig. 1, for a given time t, cicf_i is the CIC factor of ith CIC pair. It is used to describe whether this dominant mechanism pair possesses CIC as specified by mesoscience at time t. For example, if CIC_i defines the CIC relationship between dominant mechanisms $l_{i,1}$ and $l_{i,2}$ and the trends of them are maximum and minimum respectively, then cicf_i at time t is 1 if the product of $\Delta l_{i,1}$ and $\Delta l_{i,2}$ is less than 0, where $\Delta l_{i,1}$ and $\Delta l_{i,2}$ are the respective differences of $l_{i,1}$ and $l_{i,2}$ between time t and $t-1$, which means that $l_{i,1}$ and $l_{i,2}$ are increasing or decreasing non-synchronously from time $t-1$ to t, and there is some kind of competition between them at time t; otherwise, cicf_i at time t is –1.

The calculation of the mesoscience impact factor (or meso-factor), mf, at time t, constitutes the crux of the meso-constraints.

$$\mathrm{mf} = \sum_{i=1}^{n} a_i \cdot \mathrm{tf}_i + \sum_{i=1}^{m} b_i \cdot \mathrm{cicf}_i \tag{5}$$

where $n = m_g + m_h$, $m = m_{\mathrm{CIC}}$, and a_i and b_i are weighting factors for each tf_i and cicf_i at time t.

In this paper, we propose two methods for combining the regular loss with the meso-constraints:

(1) Consider the mesoscience constraints as a regularization term of the regular loss, and compute the total loss at time t using Eq. (6):

$$L_t = \alpha \cdot L_{\mathrm{r},t} + (1-\alpha) \cdot \mathrm{mf}_t \cdot \left| L_{\mathrm{r},t} - L_{\mathrm{r},t-1} \right| \tag{6}$$

where L_t is the total loss at time t; $L_{\mathrm{r},t}$ and $L_{\mathrm{r},t-1}$ are the regular losses at time t and $t-1$, respectively; mf_t is the meso-factor at time t; and α is the combined coefficient of the regular loss and meso-constraints, $\alpha \in [0,\ 1.0]$.

Eq. (6) can be qualitatively interpreted as follows: When calculating the total loss at time t, in addition to the regular loss, it is necessary to examine the degree of conformity of the dominant mechanism data of the system with the mesoscience principles, in order to decide whether to reward ($\mathrm{mf}_t > 0$) or suppress ($\mathrm{mf}_t < 0$) the updating of the network weight parameters. The magnitude of reward and punishment is determined by the absolute value of the product of mf_t and the difference between the regular loss at time t and $t-1$. Clearly, the total loss of time t is related to the time-series data of both time t and $t-1$.

(2) Correct the learning rate with mesoscience constraints. When calculating the total loss L_t at time t according to Eq. (6), the regular loss at time $t-1$ is used for the meso-constraints. It is necessary to save a large number of network parameters at time $t-1$ in order to back-propagate the error and update the network parameters at time t, which introduces programming implementation challenges. To avoid this situation, we assume that the meso-

constraints part is only related to the regular loss at time t, so the total loss at time t is

$$L_t = L_{\mathrm{r},t} + \alpha \cdot \mathrm{mf}_t \cdot L_{\mathrm{r},t} \tag{7}$$

Eq. (7) can be written in the following form:

$$L_t = L_{\mathrm{r},t} + \alpha \cdot \mathrm{mf}_t \cdot L_{\mathrm{r},t} = \left(1 + \mathrm{mf}'_t\right) L_{\mathrm{r},t} \tag{8}$$

Eq. (8) demonstrates that L_t and $L_{\mathrm{r},t}$ can be considered to have a multiplicative relationship, and $\mathrm{mf}'_t \propto \mathrm{mf}_t$. Eq. (8) can be viewed as an additional correction for the learning rate provided by the addition of the meso-constraints, allowing the trend of the system's dominant mechanisms and their CICs to affect the model-training process. Only the regular loss $L_{\mathrm{r},t}$ is used to calculate the total loss L_t at time t, which simplifies the programming implementation.

Traditional methods dealing with complex systems usually use averaging techniques to tackle the mesoscale components and cannot handle the fluctuating phenomena of the system structure properly. In the case of a small dataset used for modeling, there is not enough training data to build an end-to-end deep learning model, so the convergence of the modeling process and the predictive performance of the model are not very good. Even applying the governing equations with the averaging concept to the modeling process does not help much. In mesoscience, the CIC between a system's dominant mechanisms is treated as the intrinsic mechanism of the system's evolution, which can reflect the inherent dynamics of the system structure very well. Therefore, MGDL is expected to have better convergence and prediction performance than conventional deep learning modeling, especially when the size of the training dataset is small.

3. Validation of MGDL

3.1. Deep learning prediction of a gas–solid two-phase flow

Gas–solid two-phase flows are widely used in industry, including in petroleum refining [23], the metallurgical industry [24], and other process industries. Taking a chemical reactor as an example, alterations to the flow structure can have a significant effect on the heat and mass transfer within the reactor, thereby influencing the overall performance of the reactor. A gas–solid two-phase flow is an example of a complex system that is characterized by nonlinear non-equilibrium; a small change in operating conditions can result in a significant change in the flow-field structure of the entire reactor or even the entire system.

In the field of gas–solid two-phase flows, the application of artificial intelligence technology represented by deep learning is still in an exploratory phase. Lu et al. [25] built a graphics processing unit (GPU)-based convolutional neural network (CNN) model to accelerate the particle–particle and particle–boundary collisions calculations of granular flows in a discrete element modeling (DEM) simulation. The model was validated using a rotating drum and a hopper, and demonstrated accuracy and efficiency. Using an ANN, Yang et al. [26] proposed a generic EMMS-ANN drag model for dense fluidization beds. The model's parameters were optimized to balance the training precision and computational time. Tested on five fluidized-bed simulations under different operating conditions and material properties, reasonable predictions could be obtained with the EMMS-ANN drag model. Bazai et al. [27] used an encoder–decoder architecture to predict the solid holdup patterns in a pseudo-two-dimensional (2D) gas–solid fluidized bed. A CNN was employed to extract the features of the solid holdup pattern frames in order to encode them into a latent vector, which could be used to predict the next solid holdup pattern frame by means of a decoder—another CNN. Ouyang et al. [28] coupled computational fluid dynamics (CFD) with deep learning to close a filtered two-fluid model (TFM) for gas–particle coarse-grid simulations at the reactor scale. Mesoscale drag and solid stress were predicted by a TensorFlow deep learning model. The results suggested that—without increasing the computation cost—a deep learning model is much more advantageous in flows with high superficial gas velocities, such as turbulent fluidization regimes. Qin et al. [29] treated the voidage distribution of a gas–solid system as a 2D image, used a multi-scale CNN network to predict the voidage distribution at steady state for a bubbling bed using a short-term CFD + DEM simulation of the early stage, and achieved both good prediction accuracy and good generalizability. Upadhyay et al. [30] developed an ANN model to predict circulating fluidized bed (CFB) riser axial solid holdup. The prediction results had mean square error (MSE) values on the order of 10^{-3}, compared with the experimental data.

3.2. Simulation of a bubbling bed and acquisition of flow-field data

In this paper, MGDL was validated using a bubbling bed case study. Numerical simulations are required to collect data for constructing a sample dataset. The TFM [31], Eulerian–Lagrangian method [32], and particle-resolved direct numerical simulation (PR-DNS or DNS) [33] can be utilized to obtain detailed flow-field information of gas–solid systems in order to analyze the dynamic behavior of fluid and solid particles through simulation. In DNS, the solid phase is a particle with volume, and the fluid grid is much smaller than the particle size, so it is not necessary to introduce an empirical model to deal with the interphase forces; rather, it is sufficient to integrate the viscous force and pressure on the particle surface. There is no need to close the governing equations and introduce a trajectory model; it is possible to obtain detailed information about the fluid surrounding the particles for highly accurate simulations. Thus, DNS is appropriate for investigating the underlying mechanisms of fluidized systems. In this study, DNS is used to improve the simulation accuracy in order to effectively obtain sample data for a widely distributed gas–solid flow field in a bubbling fluidized bed.

The simulation algorithms and program used in this study are inherited from Ref. [34]. The simulation object is a 2D bubbling bed with a width of 69.4 times and a height of 290.0 times the diameter of the solid particles. Class A particles of the solid phase initially accumulate at the bottom of the bed. There is a gas distribution plate at the bottom of the bed to induct air with a flow rate of U_{g}, while the air boundary conditions on the left and right walls are nonslip. Using a soft sphere model, collisions among particles and between particles and walls were calculated. Table 1 provides the physical parameters used for the example DNS.

The Navier–Stokes equations were used to solve the motion of an incompressible, viscous Newtonian fluid in the simulations, while the Newton equations of rigid motion were used to describe the translation and rotation of the particles. The fluid–particle coupling was addressed using the immersed boundary method (IBM) [35].

Table 1
Physical parameters of the DNS.

Parameter	Value	Dimension
Domain width, W	5.816×10^{-3}	m
Domain height, H	24.302×10^{-3}	m
Grid size, Δ	5.2375×10^{-6}	m
Time step, Δt	5.5×10^{-6}	s
Gas velocity, U_{g}	1.2517	$\mathrm{m \cdot s^{-1}}$
Gas viscosity, μ	1.8872×10^{-5}	Pa·s
Gas density, ρ_{g}	1.1795	$\mathrm{kg \cdot m^{-3}}$
Particle density, ρ_{p}	508.0	$\mathrm{kg \cdot m^{-3}}$
Particle diameter, d_{p}	8.38×10^{-5}	m
Particle number, n_{p}	2070	—

In the DNS method, the fluid mesh size is typically one order of magnitude smaller than the particle size. In this work, 5.2 million fluid meshes were used. Due to the large number of fluid meshes that must be processed, DNS is extremely resource intensive, so high-performance parallel computing is typically employed to improve the simulation efficiency. The DNS was run on three parallel computing nodes with 3 × 56 = 168 Intel Xeon E5-2680 v4 @ 2.40 GHz central processing unit (CPU) cores, 3 × 128 = 384 GB random access memory (RAM), and 3 × 4 = 12 NVIDIA K80 GPUs, as well as a message-passing interface (MPI) for parallel communication. The simulation required approximately 2.5 days to reach a steady state. A total of 200 DNS cases were conducted using various combinations of three parameters: namely, the gas velocity, solid particle density, and Reynolds number.

To validate the proposed MGDL, we developed a deep learning model to predict the future 2D particle velocity field of a bubbling bed based on multiple frames of a DNS-simulated continuous 2D particle velocity field. Fig. 2 is a schematic diagram of the particle velocity field prediction. The left side is the input data of the prediction model. A constant image_step was set in advance. From a starting time step t_0, a series of simulation results stepped by the image_step were obtained, and their instantaneous velocity field was captured. These continuous velocity fields were used as the input data for the deep learning model. The right side of Fig. 2 shows the predicted velocity field. In this study, the number of velocity-field input data samples is 10, while the distance between the predicted velocity field and the input continuous velocity fields is image_step × 10.

3.3. Constructing a sample dataset for MGDL

To create the required dataset for MGDL validation, the sample dataset-building method described in Section 2.1 was applied to the deep learning model training of the bubbling bed in Section 3.2. According to the mesoscience analysis of the gas–solid two-phase flow [14], the gas-dominated tendency of the energy consumption rate per unit volume of particles, W_{st}, is to be minimal ($W_{\mathrm{st}} \rightarrow \min$), and the particle-dominated tendency of the local average voidage, ε, is to be minimal ($\varepsilon \rightarrow \min$); these behave as the two dominant mechanisms. These two extreme trends cannot occur simultaneously at the microscale; that is, there is no microscale stability condition. At the mesoscale, there is a spatiotemporal compromise between the particles and the fluid with a dynamic evolution trend of $(N_{\mathrm{st}})_{\mathrm{meso}} = W_{\mathrm{st}}/(1-\varepsilon)\rho_{\mathrm{p}} \rightarrow \min$, while $W_{\mathrm{st}} \rightarrow \min$ and $\varepsilon \rightarrow \min$ are achieved through CIC. Thus, the mesoscale stability condition can be expressed as $(N_{\mathrm{st}})_{\mathrm{meso}} \rightarrow \min$. A complex spatiotemporal structure is produced by the dynamic compromise process between the particles and fluid. Due to the reactor sidewall effect, the enhancement of $\varepsilon \rightarrow \min$ and the suppression of $W_{\mathrm{st}} \rightarrow \min$ at the sidewall result in the emergence of a macroscopic radial core–annulus structure and the emergence of a tendency toward minimization of the spatial average quantity at the macroscale; thus, the stability condition is $(\overline{N_{\mathrm{st}}})_{\mathrm{macro}} \rightarrow \min$.

In gas–solid flow simulations, it is difficult to obtain W_{st} and N_{st} values directly, because the heterogeneous structures are too complicated to be partitioned into specific homogeneous phases—namely, the dilute phase, the dense phase, and the interphase. After refining the analysis of the gas–solid system, it was possible to identify three alternative indicators: the fluid velocity of suspending and transporting particles U_{st}, the particle acceleration a, and the voidage ε [34]. Their evolution trends were minimum, minimum, and maximum, respectively, whereas the particle acceleration a and voidage ε had a CIC relationship; that is, at a given time, the gradient of variation of each a and ε should have a different sign. The variation profiles of the three physical indicators in a DNS example are depicted in Fig. 3.

In the DNS results, a particular time step was used as the starting point. Nine other frames were taken every image_step time step for a total of 10 as input, and another data frame was taken at image_step × 10 time-step intervals as the prediction target. All the 11 frames were counted to obtain the 2D velocity fields of solid particles, thereby constituting a data sample defined in Eq. (1).

The construction of a deep learning model requires a sufficient quantity of training data. In this study, substantial amounts of computational resources and time were used to conduct 200 DNSs. If only one data sample is generated from a simulation, the amount of sample data in the resulting dataset is obviously not enough, and it is a tremendous waste of computational resources. For this reason, a sufficiently long step-interval can be set up and, every other interval, a 11-velocity-field data sample shown in Fig. 2 can be extracted repeatedly from the DNS result. Using the results of 200 DNSs, this data-augmentation method can generate a sufficiently large sample dataset.

To obtain the spatiotemporal information of the bubbling bed defined in Eqs. (2) and (3), the data of U_{st}, a, and ε must be calculated and counted in relevant time steps of the DNS, and the CIC relations of a and ε must be specified.

By following the preceding two steps, a sample dataset of the bubbling bed containing embedded spatiotemporal information was created. Then, the training dataset and the testing dataset were built with a size ratio of 10:1 based on different simulations.

3.4. Training the model using MGDL

In this study, the MGDL validation has two steps: First, the classical deep learning method is used to model the bubbling bed as a benchmark; then, mesoscience concepts are incorporated into the model training process as described in Section 2.2, and the results are compared with those from the benchmark. To build a deep learning model to predict the particle velocity field in a bubbling

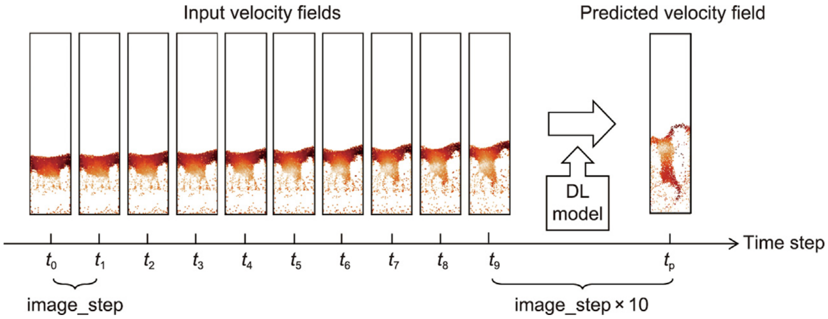

Fig. 2. Diagrammatic representation of the 2D particle velocity field prediction of the bubble bed. DL: deep learning.

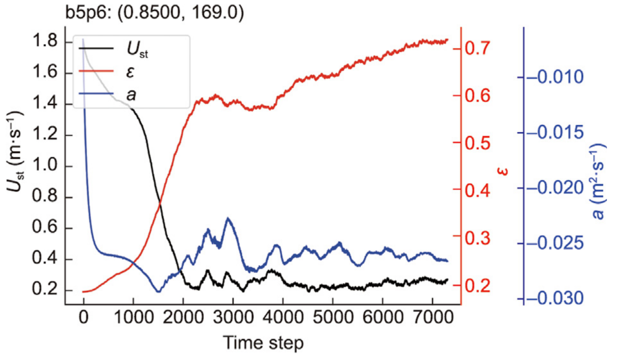

Fig. 3. U_{st}, a, and ε curves of the DNS example.

bed, the particle velocity field data in the sample dataset built in Section 3.3 must be preprocessed. The particle velocity is converted to a floating-point number in the interval [0, 1] after normalization. The 2D particle velocity field can be converted into a 240 × 56 grayscale image. Therefore, the problem of this paper can be transformed into a video prediction problem—that is, predicting a future video frame using a series of consecutive video frames.

Numerous researchers have conducted studies of video prediction and have proposed some successful deep learning models, such as long-term recurrent convolutional networks (LRCNs) [36], convolutional long short-term memory (LSTM) networks (ConvLSTMs) [37], multiscale CNNs [38], and so forth. In this study, ConvLSTM is used for modeling video prediction. ConvLSTM combines the benefits of LSTM in processing temporal data and those of convolutional networks in capturing the spatial features of images, and is suited for solving spatiotemporal sequence problems. The video prediction model was built with an encoding–forecasting structure, as illustrated in Fig. 4 [37].

In this article, the encoding network consists of three convolutional blocks and three ConvLSTM blocks. The forecasting network consists of three deconvolutional blocks, three ConvLSTM blocks, and a convolutional block. The kernel size and input/output channel number of each network block are shown in Table 2. An Adam optimizer was chosen for model training, in addition to the application of batch normalization.

In this study, the structural similarity index measure (SSIM) loss function [39] of the structural similarity based on human eye perception was not used in the training process. As shown in Eqs. (9)–(11), the MSE, which reflects the pixel difference between the predicted and actual images, and the gradient difference loss (GDL) [38], which reflects the sharpness difference between images, were combined as the regular loss. The peak signal-to-noise ratio (PSNR) [38] between the predicted image and the actual image was used as the model-predicting evaluation metric, as shown in Eq. (12). The greater the PSNR, the more accurate the prediction.

Table 2
The ConvLSTM network parameters of this study.

Block	Kernel	Input channel number	Output channel number
Encoding network			
Convolution 1	3 × 3	1	32
ConvLSTM 1	3 × 3	32	32
Convolution 2	3 × 3	32	64
ConvLSTM 2	3 × 3	64	64
Convolution 3	3 × 3	64	128
ConvLSTM 3	3 × 3	128	128
Forecasting network			
Deconvolution 1	4 × 4	128	64
ConvLSTM 1	3 × 3	128	64
Deconvolution 2	4 × 4	64	32
ConvLSTM 2	3 × 3	64	32
Deconvolution 3	4 × 4	32	32
ConvLSTM 3	3 × 3	32	32
Convolution 1	1 × 1	32	1

$$L_{\mathrm{MSE}} = \frac{1}{N}\sum_{i,j}\left(Y_{i,j} - Y'_{i,j}\right)^2 \tag{9}$$

$$\begin{aligned} L_{\mathrm{GDL}} = \sum_{i,j}\left|\left|Y_{i,j} - Y_{i-1,j}\right| - \left|Y'_{i,j} - Y'_{i-1,j}\right|\right|^{\beta} \\ + \left|\left|Y_{i,j-1} - Y_{i,j}\right| - \left|Y'_{i,j-1} - Y'_{i,j}\right|\right|^{\beta} \end{aligned} \tag{10}$$

$$L = L_{\mathrm{MSE}} + L_{\mathrm{GDL}} \tag{11}$$

$$\mathrm{PSNR} = 10\log_{10}\frac{\max_{Y'}^2}{\frac{1}{N}\sum_{i,j}\left(Y_{i,j} - Y'_{i,j}\right)^2} \tag{12}$$

In Eqs. (9)–(12), N is the total number of pixels in the velocity field image of a bubbling bed; $Y_{i,j}$ and $Y'_{i,j}$ are the target and predicted pixel values with coordinates (i, j), respectively; and β is an integer greater than 1.

After the model was constructed and tested using the conventional ConvLSTM network, mesoscience constraints were introduced during model training. MGDL was validated and compared with the conventional approach using the learning rate correction method mentioned in Section 2.2.

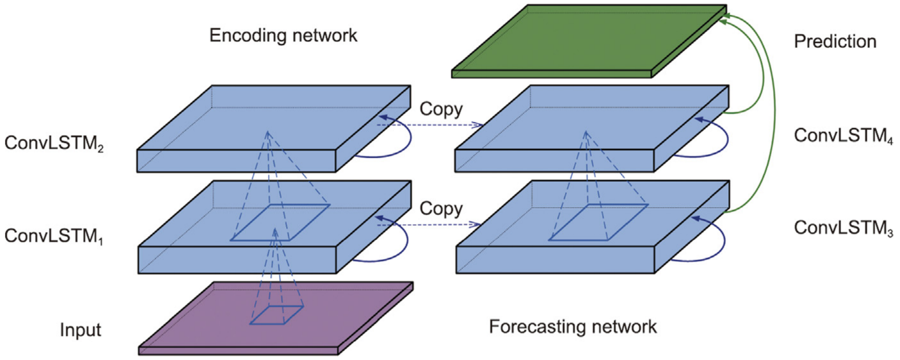

Fig. 4. ConvLSTM encoding–forecasting architecture for a spatiotemporal sequence problem. Reproduced from Ref. [37] with permission.

The stability of the training process's convergence is displayed in Fig. 5, which depicts the loss curves for various training dataset sizes after 100 epochs when the image_step is 1, 2, 3, and 5 and the batch size is 7. The blue curve represents the conventional loss as a benchmark. The loss curve is red with the addition of meso-constraints to the training process. For comparability, the red curves are drawn using the conventional mathematical loss values rather than the MGDL loss values (which include the meso-constraints).

After the model was constructed, it was necessary to evaluate its predictive ability. In this study, the size of the testing dataset is 10% of the size of the training dataset, and the testing dataset is extrapolated to the training dataset using the variable parameters of the DNS.

On the different testing datasets, the prediction performance of the models created by the two model-training methods was evaluated. Fig. 6 depicts the PSNR curves of various training dataset sizes after 100 epochs when the image_step is 1, 2, 3, and 5 and the batch size is 7. The blue curve represents the PSNR of the conventional method used as a benchmark. The PSNR curve is red with the addition of meso-constraints during the model-training process.

The example particle velocity field of the DNS for the bubbling bed is shown on Fig. 7(a), while Fig. 7(b) is the particle velocity field predicted by the model constructed using ConvLSTM + MGDL.

When meso-constraints are added to the model-training process, the batch size becomes a crucial parameter. If the batch size is excessively large, the prediction error of the training data within the batch is averaged and back-propagated. This is unfavorable for

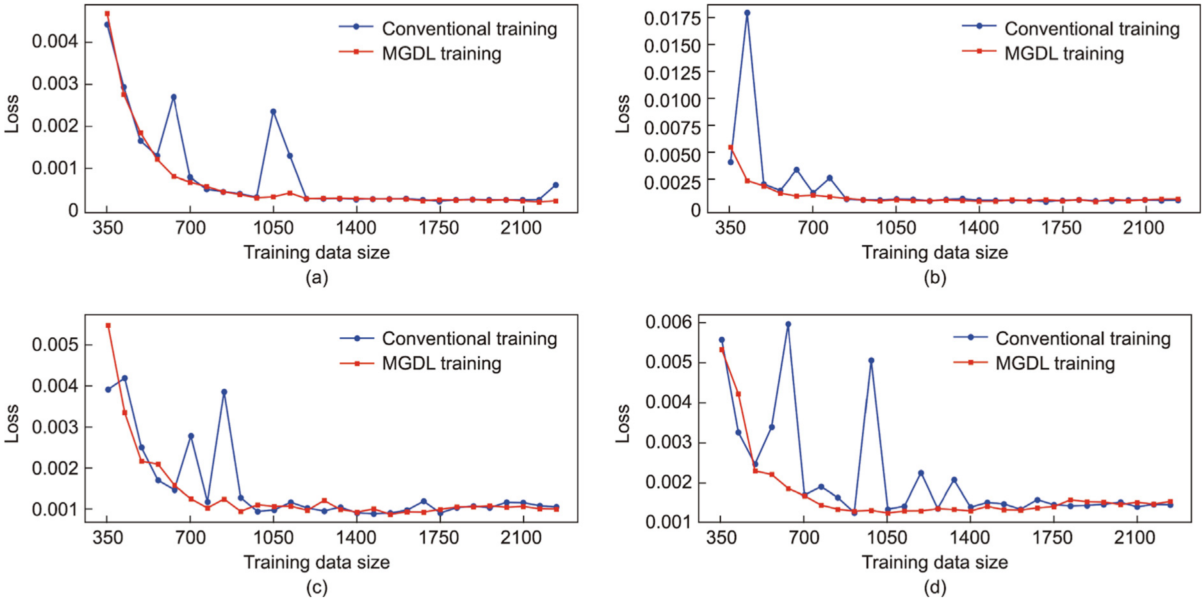

Fig. 5. Loss curves with and without meso-constraints for various training dataset sizes with different image_steps: (a) image_step = 1; (b) image_step = 2; (c) image_step = 3; and (d) image_step = 5.

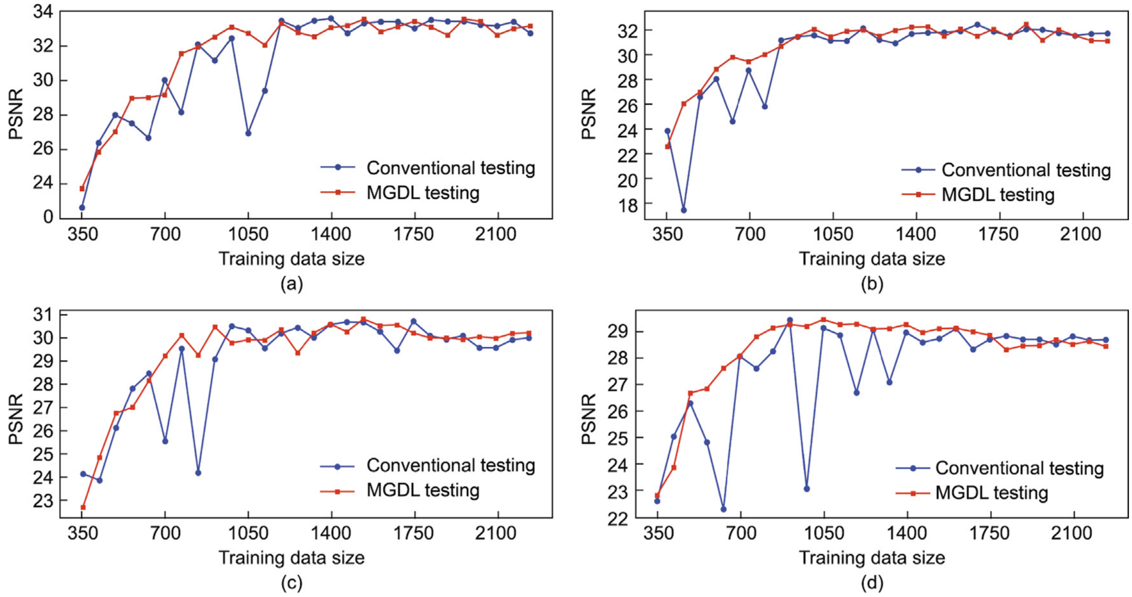

Fig. 6. PSNR curves with and without meso-constraints for various training dataset sizes with different image_steps: (a) image_step = 1; (b) image_step = 2; (c) image_step = 3; and (d) image_step = 5.

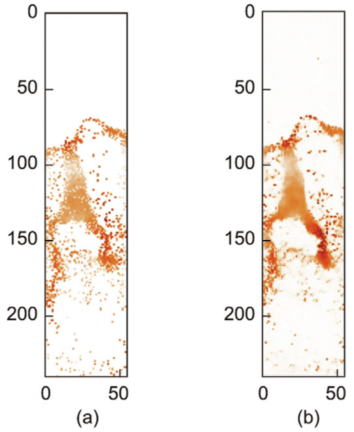

Fig. 7. (a) Particle velocity field of a bubbling bed simulated by DNS; (b) particle velocity field predicted using ConvLSTM + MGDL.

the application of the meso-constraints because it will cause the positive and negative meso-constraints within the batch to cancel each other out, thereby reducing the effectiveness of the meso-constraints during the model-training process. For this reason, this paper examines the effect of batch size on the prediction performance of the MGDL model. Fig. 8 depicts the prediction performance of the models constructed by MGDL on the testing datasets of image_steps 1, 2, 3, and 5 for varying batch sizes. The PSNR values displayed in each graph represent the mean PSNR for 20 model predictions with training dataset sizes ranging from 350 to 1680 at batch sizes 1–7. In the process of MGDL training, Fig. 8 demonstrates that an optimal batch size exists. When the training dataset is small, the best prediction accuracy is achieved with a batch size of 3.

3.5. Discussion

Incorporating mesoscience guidance into the modeling process of a bubbling bed is beneficial for the convergence. As shown in Fig. 5, the convergence curve of the model-training process with meso-constraints is more stable and smoother than that of conventional model training for smaller training datasets. For example, in Fig. 5(d), when image_step = 5, the conventional method tends to converge steadily and consistently once the size of the training dataset exceeds approximately 1400. However, the same threshold for MGDL is 840. Moreover, the convergent loss value of the conventional method oscillates when the size of the training dataset is between 840 and 1400. This phenomenon can be explained as follows: The modeling process with meso-constraints has additional physical information—that is, the essential characteristics of the evolution of a bubbling bed according to mesoscience—to guide the updating of the network parameters, aside from the training data itself.

It can be seen in Fig. 6 that, on testing datasets, the model constructed using meso-constraints has a higher prediction accuracy than a conventional training method for smaller training datasets. Using image_step = 5 as an example again, the PSNR of the predictions made by a model constructed using MGDL does not increase significantly when the size of the training dataset is greater than approximately 840, whereas this threshold is approximately 1470 for the conventional method, as seen in Fig. 6(d). Prior to that, the prediction PSNR of the model using MGDL increased steadily with the size of the training dataset; in contrast, for the conventional method, the PSNR fluctuated when the size of the training dataset was between 840 and 1470. Only two of the 17 conventional models have marginally higher PSNR values than the MGDL models for the same-sized training dataset (training dataset size from 350 to 1470). Again, this result can be explained by the incorporation of mesoscience guidance into the model-training process.

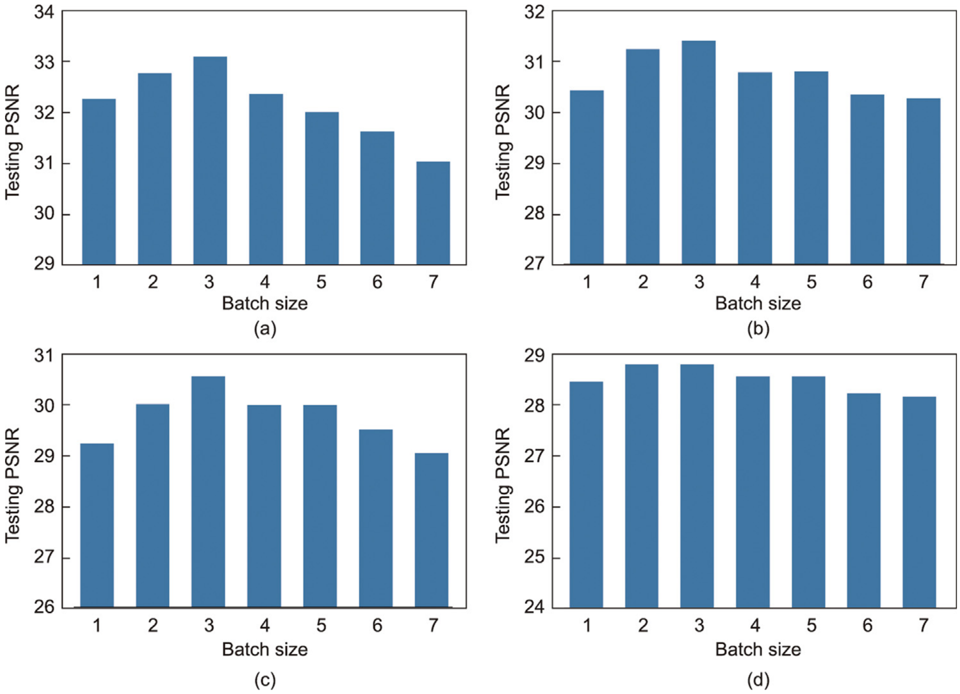

Fig. 8. Effect of batch size on the average PSNR for differently sized testing datasets of the model built with meso-constraints with different image_steps: (a) image_step = 1; (b) image_step = 2; (c) image_step = 3; and (d) image_step = 5.

To take it a step further, from Figs. 5 and 6, the inference at the end of Section 2.2 is confirmed. That is, with a relatively small training dataset, MGDL can still achieve more stable convergence and better prediction performance than conventional deep learning modeling because the mesoscale structure of the system is processed using the CIC between the system's dominant mechanisms according to mesoscience rather than using an averaging method.

Complex system numerical simulation is typically time consuming and computation intensive. Taking the small-scale bubbling bed used in this paper as an example, a DNS on three computing nodes with 12 NVIDIA K80 GPUs will take several days. Therefore, the creation of sample datasets for modeling complex systems with deep learning is expensive and time consuming. It is rational to aim to reduce the size of the training dataset without compromising the performance of the model prediction. MGDL is an effective method for achieving this objective.

Integrated deep learning modeling with physical knowledge is currently one of the hottest topics in the field, and the MGDL approach is a novel attempt. MGDL integrates the physical information embedded in the process dynamic data of the system's evolution into the modeling process, which differs from the conventional use of physical conservation or the residuals of the system's governing equations as modeling constraints. Thus far, other methods, such as PINN, must provide the governing equations or other strong mathematical correlations between system variables. If such equations are not available, it is challenging to apply these modeling methods. MGDL employs data-driven modeling while also incorporating the trends of the variables and the CICs between variables into the modeling process, because these trends and CICs often imply the intrinsic mechanisms and driving forces of a complex system's evolution. This makes MGDL a hybrid modeling technique that integrates physical information and mesoscience guidance into the data-driven model-training process. As a result, the MGDL modeling process has a certain physical interpretation, unlike purely mathematical algorithms applied on training data.

MGDL is not limited to the bubbling-bed example and the ConvLSTM employed here for demonstration; it can also be applied to solve other modeling problems of complex systems or using different network configurations, provided that the system is analyzed using mesoscience, the dominant mechanisms and their CIC relationships are identified, and the associated time-series data is collected. This flexibility is due to the fact that MGDL incorporates meso-constraints into the loss function and has no direct relationship with the type and configuration of the ANN or deep neural network employed. As long as the error back-propagation algorithm is used to optimize the network parameters, MGDL is applicable to any type of ANN or deep neural network.

According to mesoscience, complex systems are often composed of multiple levels, and mesoscales exist at each level [17,40]. The complexity of each level comes from its mesoscale, leading to the complexity of the entire system. The CIC principle between dominant mechanisms defines the structural dynamic features of mesoscale at each level, which is the key to deal with the specific complexity. It is possible to solve the problems of system-wide through the correlation between the levels. So, the application of CIC principle in single level is the crucial point of our work. The MGDL proposed in this paper applies single-level meso-constraints to improve the deep learning model training. The deep learning modeling of multilevel complex systems is a topic that needs to be studied in depth in the future.

4. Conclusions

At present, deep learning techniques are an important modeling tool for complex systems. Deep learning modeling that incorporates physical knowledge is currently a hot topic of study in academia, and a number of excellent techniques have emerged. Mesoscience is the study of mesoscale problems at various system levels, with the aim of addressing common challenges in various disciplinary fields. Understanding the macroscale behavior and intrinsic mechanisms of a system will be facilitated by the principle of CIC between the system's dominant mechanisms. The MGDL approach proposed in this paper integrates mesoscience into the deep learning modeling process, which is expected to solve the challenging problem of modeling systems that are driven by the CIC relationships between the primary physical quantities.

When tested with bubbling-bed modeling with small training datasets, the MGDL demonstrated advantages in terms of convergence stability and prediction performance. MGDL can be widely applied to various neural network configurations. With the profound study and widespread application of mesoscience, MGDL is expected to be extensively applied to the modeling of complex systems.

Acknowledgments

We appreciate Prof. Wei Ge's and Mr. Lei Bi's assistance with the DNS of the bubbling bed. We are also grateful to Miss Kaijie Wen for literature collecting. This study was financially supported by the National Natural Science Foundation of China (62050226 and 22078327) and the International Partnership Program of Chinese Academy of Sciences (122111KYSB20170068).

Compliance with ethics guidelines

Li Guo, Fanyong Meng, Pengfei Qin, Zhaojie Xia, Qi Chang, Jianhua Chen, and Jinghai Li declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Karniadakis GE, Kevrekidis IG, Lu L, Perdikaris P, Wang S, Yang L. Physics-informed machine learning. Nat Rev Phys 2021;3(6):422–40.
[2] Raissi M, Perdikaris P, Karniadakis GE. 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.
[3] Raymond SJ, Cecchi NJ, Alizadeh HV, Callan AA, Rice E, Liu Y, et al. Physics-informed machine learning improves detection of head impacts. Ann Biomed Eng 2022;50(11):1534–45.
[4] Jiang X, Wang D, Fan Q, Zhang M, Lu C, Lau APT. Physics-informed neural network for nonlinear dynamics in fiber optics. Laser Photonics Rev 2022;16(9):2100483.
[5] Sharma S, Awasthi R, Sastry YS, Budarapu PR. Physics-informed neural networks for estimating stress transfer mechanics in single lap joints. J Zhejiang Univ Sci A 2021;22(8):621–31.
[6] Mao ZP, Jagtap AD, Karniadakis GE. Physics-informed neural networks for high-speed flows. Comput Methods Appl Mech Eng 2020;360:112789.
[7] Chakraborty S. Transfer learning based multi-fidelity physics informed deep neural network. J Comput Phys 2021;426:109942.
[8] Liu X, Zhang X, Peng W, Zhou W, Yao W. A novel meta-learning initialization method for physics-informed neural networks. Neural Comput Appl 2022;34(17):14511–34.
[9] Li JH, Huang WL. Towards mesoscience: the principle of compromise in competition. Heidelberg: Springer; 2014.
[10] Li JH, Huang WL, Chen JH, Ge W, Hou CF. Mesoscience based on the EMMS principle of compromise in competition. Chem Eng J 2018;333:327–35.
[11] Li JH. Exploring the logic and landscape of the knowledge system: multilevel structures, each multiscaled with complexity at the mesoscale. Engineering 2016;2(3):276–85.
[12] Guo L, Wu J, Li JH. Complexity at mesoscales: a common challenge in developing artificial intelligence. Engineering 2019;5(5):924–9.
[13] Li JH, Tung YK, Kwauk MS. Method of energy minimization in multi-scale modeling of particle-fluid two-phase flow. In: Basu P, Large JF, editors. Circulating fluidized bed technology II. New York: Pergamon Press; 1988. p. 89–103.
[14] Li JH, Zhang JY, Ge W, Liu XH. Multi-scale methodology for complex systems. Chem Eng Sci 2004;59(8–9):1687–700.
[15] Wang LM, Qiu XP, Zhang L, Li JH. Turbulence originating from the compromise-in-competition between viscosity and inertia. Chem Eng J 2016;300:83–97.
[16] Huang WL, Li JH. Mesoscale model for heterogeneous catalysis based on the principle of compromise in competition. Chem Eng Sci 2016;147:83–90.

[17] Chen JH, Ren Y, Huang WL, Zhang L, Li JH. Multilevel mesoscale complexities in mesoregimes: challenges in chemical and biochemical engineering. Annu Rev Chem Biomol Eng 2022;13(1):431–55.
[18] Zhang JY, Ge W, Li JH. Simulation of heterogeneous structures and analysis of energy consumption in particle–fluid systems with pseudo-particle modeling. Chem Eng Sci 2005;60(11):3091–9.
[19] Zhang L, Chen JH, Huang WL, Li JH. A direct solution to multi-objective optimization: validation in solving the EMMS model for gas–solid fluidization. Chem Eng Sci 2018;192:499–506.
[20] Mann HB. Nonparametric tests against trend. Econometrica 1945;13(3):245–59.
[21] Kendall MG. Rank correlation methods. 4th ed. London: Charless Griffin; 1975.
[22] Cox DR, Stuart A. Some quick sign tests for trend in location and dispersion. Biometrika 1955;42(1–2):80–95.
[23] Squires AM. The story of fluid catalytic cracking: the first "circulating fluidized beds". In: Basu P, editor. Circulating fluidized bed technology. New York: Pergamon Press; 1986. p. 1–19.
[24] Reh L. New and efficient high-temperature processes with circulating fluid-bed reactors. Chem Eng Technol 1995;18(2):75–89.
[25] Lu L, Gao X, Dietiker JF, Shahnam M, Rogers WA. Machine learning accelerated discrete element modeling of granular flows. Chem Eng Sci 2021;245:116832.
[26] Yang Z, Lu BN, Wang W. Coupling artificial neural network with EMMS drag for simulation of dense fluidized beds. Chem Eng Sci 2021;246:117003.
[27] Bazai H, Kargar E, Mehrabi M. Using an encoder–decoder convolutional neural network to predict the solid holdup patterns in a pseudo-2D fluidized bed. Chem Eng Sci 2021;246:116886.
[28] Ouyang B, Zhu LT, Su YH, Luo ZH. A hybrid mesoscale closure combining CFD and deep learning for coarse-grid prediction of gas–particle flow dynamics. Chem Eng Sci 2022;248(Pt B):117268.
[29] Qin PF, Xia ZJ, Guo L. A deep learning approach using temporal-spatial data of computational fluid dynamics for fast property prediction of gas–solid fluidized bed. Korean J Chem Eng 2023;40(1):57–66.
[30] Upadhyay M, Nagulapati VM, Lim H. Hybrid CFD–neural networks technique to predict circulating fluidized bed reactor riser hydrodynamics. J Clean Prod 2022;337:130490.
[31] Gidaspow D. Multiphase flow and fluidization: continuum and kinetic theory descriptions. New York: Academic Press; 1994.
[32] Tsuji Y, Tanaka T, Ishida T. Lagrangian numerical simulation of plug flow of cohesionless particles in a horizontal pipe. Powder Technol 1992;71(3):239–50.
[33] Tenneti S, Subramaniam S. Particle-resolved direct numerical simulation for gas–solid flow model development. Annu Rev Fluid Mech 2014;46(1):199–230.
[34] Cui HH, Chang Q, Chen JH, Ge W. PR-DNS verification of the stability condition in the EMMS model. Chem Eng J 2020;401:125999.
[35] Peskin CS. Numerical analysis of blood flow in the heart. J Comput Phys 1977;25(3):220–52.
[36] Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, et al. Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans Pattern Anal Mach Intell 2017;39(4):677–91.
[37] Shi XJ, Chen ZR, Wang H, Yeung DY, Wong WK, Woo WC. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Cortes C, Lee DD, Sugiyama M, Garnett R, editors. Proceedings of the 28th International Conference on Neural Information Processing Systems; 2015 Dec 7–12; Montreal, QC, Canada; 2015. p. 802–10.
[38] Mathieu MM, Couprie C, LeCun Y. Deep multi-scale video prediction beyond mean square error. 2015. arXiv:1511.05440.
[39] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13(4):600–12.
[40] Li JH. Insight: the journey ahead for AI [Internet]. Norwich: Business Chief; 2020 May 20 [cited 2024 Jan 16]. Available from: https://businesschief.asia/technology/insight-the-journey-ahead-for-ai.