

Physics-informed neural network with equation adaption for ^{220}Rn progeny concentration prediction

Authors: Shaohua Hu, Qi Qiu, De-Tao Xiao, Xiang-Yuan Deng, Xiang-Yu Xu, Peng-hao Fan, Lei Dai, Zhi-Wen Hu, Tao Zhu, Qing-Zhi Zhou, Tao Zhu, Qing-Zhi Zhou

Date: 2025-07-26T16:51:52+00:00

Abstract

Physics-informed neural networks (PINNs) play a vital role in machine learning and demonstrate significant advantages in addressing complex physical problems. The PINN method enables rapid prediction of ^{220}Rn progeny concentration and holds significant importance for the regulation and measurement of this property. To construct a PINN model, training data typically undergo preprocessing; however, this process alters the physical characteristics inherent in the data, with the preprocessed data potentially no longer directly satisfying the original physical equations. As a result, the original physical equations cannot be directly applied in the PINN. Therefore, an effective approach for transforming physical equations becomes crucial for accurately constraining PINNs to model the ^{220}Rn progeny concentration prediction. This study proposes an equation adaptation approach for neural networks, specifically designed to enhance the prediction of ^{220}Rn progeny concentration. Five neural network models based on three distinct architectures are developed: a classical network, a physics-informed network without equation adaptation, and a physics-informed network with equation adaptation. The transport equation governing ^{220}Rn progeny concentration is transformed through equation adaptation and integrated with the PINN model. The compatibility and robustness of the model incorporating equation adaptation are subsequently analyzed. The results show that PINNs with equation adaptation exhibit convergence behavior consistent with classical neural networks with respect to both training and validation loss, and achieve comparable levels of prediction accuracy. These results demonstrate that the proposed method can be effectively integrated into the neural network architecture. Moreover, the prediction performance of classical neural networks declines significantly when confronted with interference data,

whereas the PINNs with equation adaptation maintain stable prediction accuracy. This performance demonstrates that the proposed method successfully leverages the constraining capability of physical equations, thereby significantly enhancing the robustness of the resultant PINN models. Consequently, employing a physics-informed network with equation adaptation ensures accurate prediction of ^{220}Rn progeny concentration.

Full Text

Preamble

Physics-Informed Neural Network with Equation Adaption for ^{220}Rn Progeny Concentration Prediction*

Shao-Hua Hu,¹ Qi Qiu,² De-Tao Xiao,¹ Xiang-Yuan Deng,¹ Xiang-Yu Xu,¹ Penghao Fan,¹ Lei Dai,¹ Zhi-Wen Hu,³ Tao Zhu,^{2,†} and Qing-Zhi Zhou^{1,‡}

¹School of Nuclear Science and Technology, University of South China, Hunan Hengyang 421001, China

²School of Computer/School of Software, University of South China, Hunan Hengyang 421001, China

³College of Computer Science and Technology, Hengyang Normal University, Hengyang, 421002, China

Physics-informed neural networks (PINNs) represent a vital advancement in machine learning, exhibiting significant advantages when addressing complex physical problems. The PINN method enables rapid prediction of ^{220}Rn progeny concentration, which is crucial for regulatory and measurement applications. To construct a PINN model, training data are typically preprocessed; however, this approach alters the physical characteristics of the data, such that the preprocessed data may no longer directly satisfy the original physical equations. Consequently, the original physical equations cannot be directly employed in the PINN framework. An effective method for transforming physical equations is therefore essential for accurately constraining PINNs in modeling ^{220}Rn progeny concentration prediction. This study presents an equation adaption approach for neural networks designed to improve prediction accuracy for ^{220}Rn progeny concentration. Five neural network models based on three architectures were established: a classical network, a physics-informed network without equation adaptation, and a physics-informed network with equation adaptation. The transport equation for ^{220}Rn progeny concentration was transformed via equation adaption and integrated into the PINN model. The compatibility and robustness of the model with equation adaption were then analyzed. The results demonstrate that PINNs with equation adaption converge consistently with classical neural networks in terms of training and validation loss, achieving comparable prediction accuracy. This outcome indicates that the proposed method can be successfully integrated into neural network architectures. Moreover, while the prediction performance of classical neural networks declines sig-

nificantly when encountering interference data, PINNs with equation adaption exhibit stable prediction accuracy. This performance demonstrates that the proposed method effectively harnesses the constraining power of physical equations, substantially enhancing the robustness of the resultant PINN models.

Thus, employing a physics-informed network with equation adaption can guarantee accurate prediction of ^{220}Rn progeny concentration.

Keywords: Machine learning, Physics-informed neural networks, Equation adaption, ^{220}Rn progeny

Introduction

Deep learning has profoundly impacted numerous areas of modern society [1–4], with significant applications in image recognition [1], natural language processing [2], cognitive science [3], and genomics [3, 4]. As a core technology of machine learning, neural networks play pivotal roles in these fields. However, traditional neural network methods require substantial volumes of training data when analyzing complex physical, biological, or engineering systems. In specialized cases, data collection costs are often high, and uncertainties exist regarding measurement accuracy, posing significant challenges for deep learning applications [5]. Furthermore, when provided with only partial datasets, most advanced machine learning techniques lack robustness and cannot draw reliable conclusions or make decisions.

In recent years, a new deep neural network (DNN) framework known as physics-informed neural networks (PINNs) has been developed. A PINN incorporates physical laws into a neural network (e.g., an artificial, recurrent, or convolutional neural network (ANN, RNN, or CNN, respectively)), constituting a data-physics dual-drive approach. This feature differentiates PINNs from traditional neural networks that rely solely on data-driven methods [1, 5]. Through the use of physical information as prior knowledge, PINNs can be trained with very few or no labeled data as alternative models for accurately solving partial differential equations [5, 6], while also incorporating complex physical laws (that are difficult to describe via theoretical equations) in data form. Thus, PINNs possess both physics- and data-driven components.

The main traditional neural network models include ANNs, RNNs, and CNNs [5]. Many advanced algorithms have been developed to optimize their performance, such as TRANSFORM-ANN [8], which can simultaneously fine-tune the neural network architecture, adjust the training dataset size, and select appropriate activation functions. To mitigate overfitting risk, TRANSFORM-ANN integrates three strategies for determining training set size, all based on Sobol sampling, making it suitable for constructing accurate and concise ANN models. However, limitations exist. Because TRANSFORM-ANN employs multi-objective optimization, it incurs high computational cost, especially for high-dimensional datasets where computational complexity increases significantly. In addition to TRANSFORM-ANN, progressive neural architecture search [9] and

one-shot neural architecture search (OSNAS) [10] represent important methods in model optimization. Boundary-integrated neural networks (BINNs) [11] are similar to PINN models. A BINN combines a boundary integral equation (BIE) with a neural network to solve acoustic radiation and scattering problems efficiently and accurately. A BINN requires only boundary-node information as input, greatly reducing computational cost and making it particularly suitable for infinite domain problems. The semi-analytical characteristics of the BIE improve BINN prediction accuracy, though application to more challenging problems in high-frequency and nonlinear acoustics and complex geometries requires further exploration.

Training data supplied to neural networks typically span physical quantities with multiple dimensions that often exhibit significantly different orders of magnitude; therefore, appropriate data preprocessing is crucial [12]. For example, physical quantities encountered in medical microdosimetry [13] and radioactive detection [14] may span hundreds of orders of magnitude or more. Even datasets with narrower ranges contain key information. To improve neural network sensitivity to data with wide ranges and ensure models fully capture key information, data must be appropriately processed using preprocessing functions. For instance, logarithmic functions can effectively scale data to an appropriate range. While preprocessing functions are essential and universal for neural network training, their application changes the dimensions of various features in the training data [1, 15]. Such alterations adversely affect PINN networks because the key concept of PINNs involves combining physical equations to guide neural network training, ensuring that model predictions conform to both data distribution and specific physical laws. When preprocessing functions change data feature dimensions, the physical equations in PINNs are no longer directly effective because equation parameters and variables are usually closely related to the dimensions of the original data.

A ^{220}Rn chamber is an essential scientific device for accurately measuring radiation dose levels of ^{220}Rn and its progeny [7, 40, 41]. The exhaust pipe is a core component of this device. When concentrations of ^{220}Rn and its progeny in the ^{220}Rn chamber must be reduced, clean air is injected into the chamber to dilute the indoor radioactive gas concentration, with excess radioactive gas released into the atmosphere via the exhaust pipe. The emitted radioactive gas concentration can reach several thousand becquerels per cubic meter; thus, effective monitoring of the emitted gas concentration distribution is necessary. To achieve precise control and accurate measurement of ^{220}Rn progeny concentration [7], a rapid prediction model must be established, which is feasible using the PINN approach. However, when developing a ^{220}Rn concentration prediction model using preprocessed training data, the preprocessing function alters the characteristics of physical quantities such as time, space, and concentration. To ensure normal functionality of physical equations in the PINN framework, these equations must be transformed according to specific preprocessing functions.

Since PINNs were proposed in 2019 [16], they have been applied to various fields.

In fluid mechanics [17–23], PINNs have proven valuable for overcoming limitations of traditional numerical simulation methods, particularly for noisy data, complex grid generation challenges, and high-dimensional flow problems. In medical diagnostics [24, 25], PINNs precisely simulate biomechanics and biofluid mechanics, elucidating complex biological fluid phenomena and aiding disease diagnosis, treatment optimization, and medical device design. In materials science [26–33], PINNs have greatly enhanced prediction accuracy of key physical quantities such as material stress and strain, especially with limited data resources. In the power industry [34–39], PINNs have been used for power-system optimization and stability analysis, combining physical laws with data analysis to accurately predict system behavior, optimize energy distribution, enhance grid stability, and improve overall energy efficiency. These applications demonstrate the excellent adaptability and reliability of PINNs. However, previous studies generally adopted conventional data processing methods without exploring the issue of physical-equation deformation and incorporation into neural networks after preprocessing. In particular, for effective ^{220}Rn progeny concentration prediction, developing a PINN with equation adaption is highly significant.

This study introduces an equation adaption approach for neural networks that can accurately transform physical equations for application in PINN model training. The compatibility of this method with neural networks and the robustness of the resultant model are explored. The remainder of this paper is organized as follows. In Section II, five neural network models are established based on three architectures: a classical network, a physics-informed network without equation adaptation, and a physics-informed network with equation adaptation. The equation adaption process is then applied, and the transformation of physical equations based on specific preprocessing functions is demonstrated. Section III focuses on the compatibility of the neural-network equation adaption approach and the robustness of the PINN network after equation adaption, based on the five models established previously. Section IV concludes the work.

II. Methodology

When training data are processed by a preprocessing function, physical equations must be transformed before incorporation into the PINN architecture. This section first establishes five prediction models for ^{220}Rn concentration based on three network architectures. Then, the proposed equation adaptation method is introduced, with the equations being incorporated into the PINN model.

A. Physical Object

In this study, the exhaust pipe of the ^{220}Rn chamber served as the research object, and the concentration distribution of the emitted radioactive gas was predicted. The device was cylindrical with a diameter of $\Phi = 10$ cm and a length L of 40 cm [Figure 1: see original paper]. Gas entered through the inlet and exited through the outlet, primarily comprising a mixture of ^{220}Rn progeny

and air. The inlet wind speed was used as the model boundary condition, with an adjustment range of 0–0.1 m/s [7]. The main decay products of ^{220}Rn are ^{216}Po , ^{212}Pb , and ^{212}Bi . Because ^{216}Po has a short half-life of only 0.145 s, its migration and diffusion capabilities are minimal. In contrast, the latter two have more prolonged half-lives of 10.64 h and 50.55 min, respectively, making their migration and diffusion more impactful. Thus, ^{212}Pb and ^{212}Bi are the focus of research attention [7, 40].

As ^{212}Pb and ^{212}Bi exhibit highly similar migration and diffusion patterns in this context, this study considered only the ^{212}Pb concentration distribution.

B. Establishment of Neural Network

Computational fluid dynamics (CFD) was used to establish a numerical simulation of the physical object (i.e., the exhaust pipe) discussed in Section II A, thereby obtaining the database needed to train the neural network. This database was used to train and validate subsequent neural network models. The physical equations were incorporated into the loss function to jointly constrain training of the neural network model. Finally, the data and physical-equation loss function were combined to train the neural network model. The model construction flowchart is shown in [Figure 2: see original paper].

The physical structure considered in this study was a cylinder, which is highly symmetrical. Therefore, in experiments, only the ^{212}Pb concentration distribution in the xoy plane area is often required. In addition, because the physical structure is highly symmetrical and fluid velocity is minimal, the Reynolds number is far less than 2000, establishing laminar flow characterized by a linear and uniform velocity distribution with a stable pressure gradient. Considerable similarity exists between two- and three-dimensional flows; that is, the characteristics of a three-dimensional flow field can be described by a two-dimensional simulation, thus avoiding the difficulty of establishing a three-dimensional model [43]. Therefore, the neural network model established in this study was designed to predict the ^{212}Pb concentration distribution on the xoy plane.

(1) Data Collection PINN models exhibit high robustness and can effectively handle data with significant errors [44, 45]. In this study, to test the robustness of the PINN model established using the proposed method, two databases were created: one without and one with interference (labeled Data-01 and Data-02, respectively). Data-02 comprised Data-01 with the addition of a small volume of data containing significant errors. The databases contained ^{212}Pb concentrations spanning an extensive range, from 10^{-4} Bq/m^3 to 10^3 Bq/m^3 . Data-01 comprised $N_{\text{data}} = 987,135$ normal data points, whereas Data-02 comprised $N_{\text{data}} = 987,135$ normal data points and $N_{\text{error}} = 50$ erroneous data points. A random 0.2% of the data from Data-01 was selected as the validation set ($N_{\text{validation}} = 2,000$ data points), labeled “Data-validation.” Finally, these two databases were used to separately train three networks: NN, PINN-EA, and PINN-f, yielding five models: NN, NN-ERR, PINN-EA, PINN-EA-

ERR, and PINN-f. The correspondence between the different training databases and training models is shown in [Figure 3: see original paper]. NN, PINN-EA, and PINN-f are introduced in detail in the following subsection. Note that Data-02 had 50 additional data points compared to Data-01, accounting for 0.005% of the total data—a tiny proportion. Therefore, the data volumes of Data-02 and Data-01 were considered identical, and the influence of the additional 50 data points on model training was ignored.

(2) Neural Networks A PINN is essentially a DNN that can approximate a solution determined from data and PDEs [46]; its architecture is shown in [Figure 4: see original paper]. In this study, three neural networks were constructed: NN, PINN-EA, and PINN-f. NN was a classical neural network without physical laws, whereas PINN-EA and PINN-f were PINNs with physical laws. PINN-EA and PINN-f differed in terms of their preprocessing functions. The equation adaption approach proposed in this study was adopted for PINN-EA; that is, the physical law $F(X, Y, U, V, C)$ was used in the network. No preprocessing function was used for PINN-f, with the physical law $f(x, y, u, v, c)$ being directly integrated into the network. The physical laws $F(X, Y, U, V, C)$ and $f(x, y, u, v, c)$ are explained in detail in Section II C.

A residual neural network [47, 48] was adopted in this study, for which the relationship between inputs and outputs can be expressed as $(u, v, c) = \text{TNN}(x, y, t, q; \Theta)$. Here, TNN represents the neural network, with inputs being the space coordinates (x and y), time (t), and wind speed (q). The neural network outputs are the velocity vectors (u and v) and concentration (c). The parameter Θ represents the trainable variables. Through this network, the relationship between inputs and outputs is constructed. The k -th hidden layer for the residual neural network is expressed as $H_k = (H_{k-1}W_k + b_k)^{\sigma}$, where W and b are the weights and biases, respectively; H represents the output of each hidden layer; and σ is the activation function.

In this work, the partial derivatives $\frac{\partial}{\partial x}$, $\frac{\partial}{\partial y}$, and $\frac{\partial}{\partial t}$ were computed based on the chain rule, which has previously been implemented via automatic differentiation in both TensorFlow and PyTorch [46]. This study employed PyTorch for this computation. Note that higher-order derivatives can be estimated through multiple calls to this function. Based on the 220Rn progeny transport equation detailed in Eqs. (9) and (18) of Section II C, the residual equation of the transport equation can be derived as follows: $-De(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2})c + \frac{\partial c}{\partial t}$.

The loss function within the PINN is defined as $L = L_{\text{data}} + \lambda L_{\text{eqns}}$, where λ is a weighting coefficient whose value is discussed in Section III C. Further, L_{data} and L_{eqns} are computed as:

$$L_{\text{data}} = (1/N_{\text{data}}) \sum (|u_{\text{data}} - u_{\text{pred}}| + |v_{\text{data}} - v_{\text{pred}}| + |c_{\text{data}} - c_{\text{pred}}|)$$

$$L_{\text{eqns}} = (1/N_{\text{data}}) \sum |e|$$

Here, u_i_data , v_i_data , and c_i_data are the measured data; u_i_pred , v_i_pred , and c_i_pred are the predicted data; L_{data} represents the loss between measured and predicted data; and e represents e_f or e_F . Recall that in PINN-F, $e = e_F$, whereas in PINN-f, $e = e_f$. The variables Θ are optimized by minimizing the loss function. In this study, training variables were updated using the adaptive moment estimation (Adam) optimizer with an initial learning rate of 0.001. The learning rate decreased stepwise every 40 epochs to 0.9 times the original learning rate. Five models were obtained by training the three neural networks with different databases. The hyperparameter settings of the five models are detailed in . Note that the specified number of iterations was sufficient to decrease the model training error to a stable condition. The numbers of hidden units and layers (N_{cell} and N_{layer} , respectively) were determined as discussed in Section III C.

(3) Activation Functions The activation function is pivotal to a neural network's ability to approximate data. Without an activation function, the network would perform only linear transformations [46]. Given that a PINN incorporates a derivation process, selection of an appropriate activation function is essential for practical model training. Five representative activation functions were used to construct and optimize the model training process in this study: Sigmoid, Tanh, ReLU, Leaky ReLU, and Hardswish. These activation functions exhibit unique nonlinear mapping characteristics, spanning both saturated (Sigmoid, Tanh) and unsaturated (ReLU, Leaky ReLU, Hardswish) types, as well as parameterized (e.g., the negative slope parameter of Leaky ReLU) and nonparametric designs. These five activation functions are representative and widely used in the field of neural networks [49, 50].

C. Equation Adaption

The migration and diffusion behaviors of ^{212}Pb within the device follow the transport equation:

$$f(x, y, t, u, v, c) = c/t + (uc)/x + (vc)/y - De(\partial^2 c/\partial x^2 + \partial^2 c/\partial y^2) - \lambda c = 0$$

where x and y are spatial coordinates; u and v are the flow-field velocities at these coordinates; c is the ^{212}Pb concentration at these coordinates; De is the diffusion coefficient; and λ is the ^{212}Pb decay constant.

First, the $f(x, y, u, v, c)$ expression is established. The preprocessing function and its inverse are represented by N and FN , respectively. Further, X , Y , T , Q , U , V , and C are the parameters obtained by transforming x , y , t , q , u , v , and c through the preprocessing function. Their relationships can be expressed as:

$$\begin{aligned} x &= FN(X) \\ y &= FN(Y) \\ t &= FN(T) \\ q &= FN(Q) \\ u &= FN(U) \end{aligned}$$

$$\begin{aligned}v &= \text{FN}(V) \\c &= \text{FN}_c(C)\end{aligned}$$

From these relationships, the physical Eq. (9) can be transformed as follows:

$$\begin{aligned}F(X, Y, T, U, V, C) &= \text{FN}_c(C)/\text{FN}(T) + [\text{FN}(U)\text{FN}_c(C)]/\text{FN}(X) + \\&[\text{FN}(V)\text{FN}_c(C)]/\text{FN}(Y) - \text{De}[\text{FN}_c(C)/\text{FN}(X)^2 + \text{FN}_c(C)/\text{FN}(Y)^2] \\&- \text{FN}_c(C) = 0\end{aligned}$$

To improve transparency of the subsequent equation derivation, this expression is decomposed into four parts: A, B, E, and G:

$$\begin{aligned}A &= \text{FN}_c(C)/\text{FN}(T) \\B &= [\text{FN}(U)\text{FN}_c(C)]/\text{FN}(X) + [\text{FN}(V)\text{FN}_c(C)]/\text{FN}(Y) \\E &= \text{De}[\text{FN}_c(C)/\text{FN}(X)^2 + \text{FN}_c(C)/\text{FN}(Y)^2] \\G &= -\text{FN}_c(C)\end{aligned}$$

This transformed equation represents the physical Eq. (9) in the preprocessed space. If the preprocessing function is determined, a further precise transformation can be performed. The normalization function was employed for preprocessing in this study. The normalization function and its inverse are respectively expressed as:

$$\begin{aligned}N() &= (- \text{min})/(\text{max} - \text{min}) \\FN() &= (\text{max} - \text{min}) + \text{min}\end{aligned}$$

where represents the parameters x, y, t, q, u, v , and c ; min and max represent the minimum and maximum parameter values, respectively; and represents the normalized parameters X, Y, T, Q, U, V , and C .

Therefore, based on these definitions, the transformed equation can be expressed as:

$$F(X, Y, T, U, V, C) = A + B - E + G$$

By applying the chain rule for differentiation of composite functions [42], this can be expanded to:

$$\begin{aligned}A' &= (C(c_{\text{max}} - c_{\text{min}}) + c_{\text{min}})/(t_{\text{max}} - t_{\text{min}}) \\B' &= [(c_{\text{max}} - c_{\text{min}})/(x_{\text{max}} - x_{\text{min}})]U(u_{\text{max}} - u_{\text{min}}) + [(c_{\text{max}} - c_{\text{min}})/(y_{\text{max}} - y_{\text{min}})]V(v_{\text{max}} - v_{\text{min}}) \\E' &= \text{De}(c_{\text{max}} - c_{\text{min}})[1/(x_{\text{max}} - x_{\text{min}})^2 + 1/(y_{\text{max}} - y_{\text{min}})^2] \\G' &= -(C(c_{\text{max}} - c_{\text{min}}) + c_{\text{min}})\end{aligned}$$

Thus, the final transformed equation becomes:

$$F(X, Y, T, U, V, C) = A' + B' - E' + G' = 0$$

Therefore, after the preprocessing function is determined, the original physical Eq. (9) is transformed into this adapted form, enabling implementation of the equation adaption technique.

D. Model Evaluation Indexes

In this study, three key indicators were used to gauge the performance of trained models: training loss, validation loss, and relative standard deviation (RSD). The RSD measures the relative discrepancy between predicted and true values, calculated as:

$$\text{RSD} = (1/N_{\text{data}}) \sum |y_{\text{tr}} - y_{\text{pre}}| / y_{\text{tr}} \times 100\%$$

where y_{tr} and y_{pre} are the true and predicted values, respectively.

To facilitate analysis and comparison of different models, the final training and validation losses (FT and FV, respectively) were adopted as metrics. Consequently, the FT and FV for the NN and PINN models were denoted as FT_NN, FV_NN, FT_PINN, and FV_PINN, respectively. The ratios between the two models were defined as $K_{\text{FT}} = \text{FT_NN} / \text{FT_PINN}$ and $K_{\text{FV}} = \text{FV_NN} / \text{FV_PINN}$.

III. Results and Discussion

Section II presented the basic conditions for establishing the PINN model. This section details the optimization of model parameters and subsequent performance analysis of the optimal model. The performance analysis is reported first in Sections III A and III B, followed by discussion of the basis for determining model parameters in Section III C.

A. Convergence and Predictive Performance of PINNs Without Equation Adaption

This section discusses the convergence and predictive performance of the PINN network without equation adaption trained on the Data-01 database. The PINN model discussed corresponds to the PINN-f model in Section II B. [Figure 5: see original paper] shows the training and validation loss convergence during training of the PINN-f model. Following application of the Leaky ReLU and Hardswish activation functions, the model training and validation losses exceeded orders of magnitude of 10^{23} and 10 , respectively, and did not converge to smaller values. However, for models with ReLU, Sigmoid, and Tanh activation functions, these values converged to 10^{-3} . Therefore, compared to Leaky ReLU and Hardswish, the ReLU, Sigmoid, and Tanh activation functions yielded better convergence of model training and validation losses.

Although training and validation losses are fundamental metrics for evaluating model training effectiveness, the accuracy of physical quantity predictions is also crucial. In this study, model accuracy was assessed using predictive RSD, which measures the relative standard deviation between predicted and true values as expressed in Eq. (19). [Figure 6: see original paper] illustrates the RSD between predicted and true values for the x- and y-velocity components and ^{212}Pb concentration for the PINN-f model. Models employing Leaky

ReLU and Hardswish activation functions exhibited predictive RSDs for the three considered physical quantities that significantly exceeded 10^{-1} , with the predictive RSD for ^{212}Pb concentration reaching the order of 10^2 . Models using ReLU, Sigmoid, and Tanh activation functions exhibited smaller predictive RSDs. However, the values for the three physical quantities remained relatively high, all exceeding 1, with the predictive RSD for ^{212}Pb concentration reaching the order of 10^{11} . Therefore, PINN-f models trained with these five activation functions did not achieve satisfactory prediction accuracy.

B. Convergence and Predictive Performance of PINNs With Equation Adaption

As indicated in Section III A, the neural network method without equation adaption failed to accurately predict the x- and y-velocity components and ^{212}Pb concentration. This section compares the neural network model with equation adaption to that without equation adaption and verifies the compatibility and robustness of the proposed method within neural networks.

1. Comparative Analysis of PINN and NN Models Without Interference Data

(1) Neural-Network Convergence This section discusses the training and predictive performance of the PINN network with equation adaption and the classical neural network, both trained without interference data (i.e., on the Data-01 database). These models correspond to the PINN-EA and NN models described in Section II B, respectively. [Figure 7: see original paper] shows the training and validation loss convergence for both models under different activation function conditions. Without interference data, the training and validation loss convergence patterns of the two models were consistent, reaching the order of 10^{-1} . Compared to results reported in Section III A, these outcomes indicate that database preprocessing effectively reduces model training difficulty and facilitates training and validation loss convergence, further demonstrating the necessity of equation adaption in PINN model training.

[Figure 8: see original paper] compares the magnitudes to which the two models' training and validation losses converged after 1000 training epochs under different activation function conditions. As the activation function changed, the two models exhibited consistent trends in both FT and FV, with K values fluctuating around 1. This indicates almost no difference between the FT values of the two models and similarly that the FV values were almost identical. Therefore, without interference data, comprehensive analysis of FT, FV, and K values suggests that the neural network model with equation adaption (PINN-EA) and the classical neural network model (NN) exhibit good consistency.

(2) Model Prediction Accuracy [Figure 9: see original paper] illustrates the change pattern of predictive RSD for the x-velocity and y-velocity compo-

nents and ^{212}Pb concentration as the NN and PINN-EA models underwent continuous training. When the ReLU activation function was used, the RSD for the x-velocity component exceeded 100%. For other activation functions, however, RSD values remained below 100%. The lowest RSD values were obtained for the y-velocity component, with all values below 10%, whereas those for x-velocity and ^{212}Pb concentration fell between 10% and 100%. [Figure 9: see original paper] also displays the RSD change pattern for the PINN-EA model, which was essentially the same as for the NN model for all three physical properties.

To more intuitively illustrate the relationship between ^{212}Pb concentrations predicted by the NN and PINN-EA models and the true values, [Figure 10: see original paper] shows scatter plots of predicted versus true values. The true and predicted values were normalized using the `_min` and `_max` from Eq. (15). The closer scatter points are to the $y = x$ line, the closer predicted values are to true values, indicating better prediction accuracy. The scatter points in [Figure 10: see original paper] are essentially on the $y = x$ line, suggesting both NN and PINN-EA models had good prediction accuracy.

In summary, analysis of training and validation loss convergence patterns and model predictive accuracy revealed that, without interference data, the NN and PINN-EA models exhibit consistent convergence during training. Moreover, compared to models in Section III A, these two models achieved higher prediction accuracy for ^{212}Pb concentration. These results demonstrate that the equation adaption technique can be effectively integrated into neural networks without conflict, showing good compatibility.

2. Comparative Analysis of PINN and NN Models With Interference Data To verify the robustness of the PINN model following adoption of the equation adaption technique, both the classical neural network and the PINN network were trained using data containing interference (the Data-02 database). The training effects were examined for both models, corresponding to the NN-ERR and PINN-EA-ERR models described in Section II B.

(1) Convergence of Neural Networks [Figure 11: see original paper] shows training and validation loss convergence with epochs for the two models under different activation functions. When model training stabilized, although the NN-ERR training loss was considerably smaller than that of PINN-EA-ERR, the PINN-EA-ERR validation loss was smaller than that of NN-ERR by approximately half. Further, both NN-ERR and PINN-EA-ERR models exhibited overfitting at approximately 200 epochs. reports the magnitudes to which training and validation losses of the classical neural network and PINN network converged after 2000 training epochs. The FT values of both models were within the range of 1×10^{-4} to 1×10^{-3} , and the FV values exceeded 1×10^{-1} . Under conditions without interference data, as shown in [Figure 8: see original paper], the FT and FV values of the two models were within the range of 1×10^{-4}

to 1×10^{-6} . This indicates that under the influence of interference data, FT increased by two to three orders of magnitude, and FV increased by six orders of magnitude. Therefore, interference data has a significant adverse effect on model training.

[Figure 12: see original paper] shows the ratios of FT or FV values between NN-ERR and PINN-EA-ERR under different activation functions. The FT ratios of the two models were less than 1, whereas the FV ratios were larger than 1. This result indicates that the PINN-EA-ERR model, with equation adaption constraint, can recognize erroneous data, further enhancing neural network robustness.

(2) Model Prediction Accuracy [Figure 13: see original paper] shows RSD values between predicted and true values for the x- and y-velocity components and 212Pb concentration for the NN-ERR and PINN-EA-ERR models under different activation function conditions. Comparison reveals that for the x and y fluid velocity vectors, the RSD of PINN-EA-ERR model predictions were half those of the NN-ERR model. Regarding 212Pb concentration prediction, the RSD of PINN-EA-ERR was approximately 1/400th that of NN-ERR. This indicates that PINN-EA-ERR had higher prediction accuracy than NN-ERR, especially for 212Pb concentration. The main reason is that the physical equation used is the 212Pb concentration transport equation, which effectively constrained 212Pb concentration prediction. This equation also incorporates flow-field physical quantities (x- and y-velocity components), yielding some improvement in their prediction accuracy, though the constraining force was insufficient, resulting in a stark contrast in accuracy.

[Figure 14: see original paper] depicts scatter plots of 212Pb concentrations predicted by the NN-ERR and PINN-EA-ERR models against true values. The true and predicted values were uniformly normalized as in [Figure 10: see original paper]. Figure 14: see original paper-(e) and (f)-(j) show scatter plots for NN-ERR and PINN-EA-ERR models, respectively. In Figure 14: see original paper-(e), most scatter points lie on $y = x$, except those close to 0, indicating significant deviation in NN-ERR predictions for values near 0, which markedly decreased prediction accuracy. In contrast, PINN-EA-ERR predicted scatter points Figure 14: see original paper-(j) lie mainly on the $y = x$ line, and the model maintained good prediction accuracy for values close to 0. This further demonstrates the effectiveness of the practical constraint provided by equation adaption on neural network training.

C. Ndata, Ncell, and Parameters

Parameter optimization is typically performed for neural networks with different activation functions. Based on Sections III A and III B, optimal performance was obtained with the tanh activation function. Therefore, this section takes the tanh activation function model as an example and discusses the parameter

optimization process in detail, primarily considering the influence of Ndata, Ncell, and σ on prediction accuracy without interference data.

The ^{212}Pb concentration predicted by the PINN for different Ndata values is presented in [Figure 15: see original paper]. The RSD of ^{212}Pb concentration prediction was used to characterize model prediction accuracy. The layer number was fixed at 5 for all cases, and the number of neurons in each layer was varied from 16 to 128.

Three conclusions can be drawn from [Figure 15: see original paper]. First, regarding data dependence: With increasing Ndata, prediction accuracy for ^{212}Pb concentration improved significantly, but when Ndata reached 10^3 , prediction accuracy did not change considerably. Second, regarding Ncell determination: With increasing Ncell, prediction accuracy increased gradually; the accuracies for Ncell = 64 and 128 were close. Considering computational resources, the optimal Ncell for this model was 64. Third, regarding σ determination: In the training experiment for the PINN neural network model, the order of magnitude of Leqns was approximately 10^{-3} , and the order of magnitude of Ldata was approximately 10^{-1} . To facilitate supervision of neural network training, similar orders of magnitude were set for Leqns and Ldata, so prediction accuracy changes were analyzed when σ was 10^2 – 10^2 . As apparent from [Figure 15: see original paper], with sufficient training data and increasing σ , model prediction accuracy first increased rapidly then decreased slightly, with optimal performance obtained when σ was 10^2 . Therefore, Ncell = 64 and $\sigma = 10^2$ were selected for the optimal model.

This study had some limitations. High accuracy was not achieved for flow field prediction when interference data were considered (Section III B 2), indicating that physical equation conditions should be adjusted to constrain the model and enhance prediction accuracy. Regarding noise in the training stage, training data were preprocessed to a certain extent, yielding a good signal-to-noise ratio.

IV. Conclusion

In this study, a PINN with equation adaption was established for ^{220}Rn progeny concentration prediction. A PINN without equation adaption was examined and failed to yield desired outcomes, underscoring the critical role of equation adaption in training neural networks. For training without interference data, the PINN with equation adaption exhibited performance consistent with that of a classical neural network model, achieving high accuracy when predicting ^{220}Rn concentrations. This outcome emphasizes the excellent compatibility of the equation adaption technique with neural networks. When interference data were considered, the PINN model with equation adaption retained good prediction accuracy, especially for ^{220}Rn concentration prediction. This highlights the effectiveness of equation adaption in constraining neural networks with physical equations, thereby improving model robustness.

In future work, different types of noise will be added to the model based on fac-

tors such as background radioactivity level and detection method. Additionally, the equation adaption technique will be used to model specific physical objects with PINNs; for example, precise and rapid prediction of ^{220}Rn and its progeny concentrations will be explored. Overall, the equation adaptation approach presented in this study has good universality and provides a theoretical foundation for widespread application of neural networks in various fields.

References

- [1] A. Krizhevsky, I. Sutskever, G.E. Hinton et al., ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90 (2017). doi:10.1145/3065386
- [2] Y. LeCun, Y. Bengio, G. Hinton et al., Deep learning. *Nature* 521, 436–444 (2015). doi:10.1038/nature14539
- [3] B.M. Lake, R. Salakhutdinov, J.B. Tenenbaum et al., Human-level concept learning through probabilistic program induction. *Science* 350, 1332–1338 (2015). doi:10.1126/science.aab3050
- [4] B. Alipanahi, A. Delong, M.T. Weirauch et al., Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nat. Biotechnol.* 33, 831–838 (2015). doi:10.1038/nbt.3300
- [5] G.E. Karniadakis, I.G. Kevrekidis, L. Lu et al., Physics-informed machine learning. *Nat. Rev. Phys.* 3, 422–440 (2021). doi:10.1038/s42254-021-00314-5
- [6] L. Yuan, Y.Q. Ni, X.Y. Deng et al., A-PINN: Auxiliary physics informed neural networks for forward and inverse problems of nonlinear integro-differential equations. *J. Comput. Phys.* 462, 111260 (2022). doi:10.1016/j.jcp.2022.111260
- [7] S.H. Hu, Y.J. Ye, Z.Z. He et al., Analysis and optimization of performance parameters of the ^{220}Rn chamber in flow-field mode using computational fluid dynamics method (2024). arXiv:202406.0035504525. doi:10.12074/202406.00355V1
- [8] S.S. Miriyala, V.R. Subramanian, K. Mitra, TRANSFORM-ANN for online optimization of complex industrial processes: Casting process as case study. *Eur. J. Oper. Res.* 264, 294–309 (2018). doi:10.1016/j.ejor.2017.05.026
- [9] C. Liu, B. Zoph, M. Neumann et al., Progressive neural architecture search. In *Proceedings of the European conference on computer vision (ECCV)*, 19–34 (2018). doi:10.1007/978-3-030-01234-2_2
- [10] X. Xie, X. Song, Z. Lv et al., Efficient evaluation method for neural architecture search: A survey (2023). arXiv:2301.05919.
- [11] W. Qu, Y. Gu, S. Zhao et al., Boundary integrated neural networks and code for acoustic radiation and scattering. *Int. J. Mech. Syst. Dynam.* 4, 2767–1399 (2024). doi:10.1002/msd2.12109
- [12] S. García, J. Luengo, F. Herrera et al., *Data preprocessing in data mining*. (Springer, Switzerland, 2015), 59–139.
- [13] M.V. Nuland, H. Rosing, A.D. Huitema et al., Predictive value of microdose pharmacokinetics. *Clin. Pharmacokinet.* 58, 1221–1236 (2019). doi:10.1007/s40262-019-00769-x

- [14] N. Keat, J. Kenny, K. Chen et al., A microdose PET study of the safety, immunogenicity, biodistribution, and radiation dosimetry of ^{18}F -FB-A20FMDV2 for imaging the integrin $\alpha_v\beta_6$. *J. Nucl. Med. Technol.* 46, 136-143 (2018). doi:10.2967/jnmt.117.203547
- [15] L. Pouchard, K.G. Reyes, F.J. Alexander et al., A rigorous uncertainty-aware quantification framework is essential for reproducible and replicable machine learning workflows. *Digital Discovery* 2, 1251-1258 (2023). doi:10.1039/D3DD00094J
- [16] M. Raissi, P. Perdikaris, G.E. Karniadakis et al., Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686-707 (2019). doi:10.1016/j.jcp.2018.10.045
- [17] S. Cai, Z. Mao, Z. Wang et al., Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sinica-PRC* 37, 1727-1738 (2021). doi:10.1007/s10409-021-01148-1
- [18] Q. He, D. Barajas-Solano, G. Tartakovsky et al., Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Adv. Water Resour.* 141, 103610 (2020). doi:10.1016/j.advwatres.2020.103610
- [19] S. Falas, C. Konstantinou, M.K. Michael et al., Physics-informed neural networks for securing water distribution systems (2020). arXiv:2009.08842.
- [20] C. Cheng, G.T. Zhang, Deep learning method based on physics informed neural network with resnet block for solving fluid flow problems. *Water-Sui* 13, 423 (2021). doi:10.3390/w13040423
- [21] Z. Mao, A.D. Jagtap, G.E. Karniadakis et al., Physics-informed neural networks for high-speed flows. *Comput. Method. Appl. M.* 360, 112789 (2020). doi:10.1016/j.cma.2019.112789
- [22] Q. Zhu, Z. Liu, J. Yan et al., Machine learning for metal additive manufacturing: Predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Comput. Mech.* 67, 619-635 (2021). doi:10.1007/s00466-020-01952-9
- [23] J. Du, J. Zheng, Y. Liang et al., Deeppipe: A two-stage physics-informed neural network for predicting mixed oil concentration distribution. *Energy* 276, 127452 (2023). doi:10.1016/j.energy.2023.127452
- [24] A. Arzani, J.X. Wang, R.M. D' Souza et al., Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Phys. Fluids* 33, 071905 (2021). doi:10.1063/5.0055600
- [25] F.S. Costabal, Y. Yang, P. Perdikaris et al., Physics-informed neural networks for cardiac activation mapping. *Front. Phys-Lausanne* 8, 42 (2020). doi:10.3389/fphy.2020.00042
- [26] Y. Chen, L. Lu, G.E. Karniadakis et al., Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Opt. Express* 28, 11618-11633 (2020). doi:10.1364/OE.384875
- [27] S. Goswami, C. Anitescu, S. Chakraborty et al., Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theor. Appl. Fract. Mec.* 106, 102447 (2020). doi:10.1016/j.tafmec.2019.102447

- [28] E. Zhang, M. Yin, G.E. Karniadakis et al., Physics-informed neural networks for nonhomogeneous material identification in elasticity imaging (2020). arXiv:2004.04525.
- [29] M. Yin, X. Zheng, J.D. Humphrey et al., Non-invasive inference of thrombus material properties with physics-informed neural networks. *Comput. Method. Appl. M.* 375, 113603 (2021). doi:10.1016/j.cma.2020.113603
- [30] R. Zhang, Y. Liu, H. Sun et al., Physics-informed multi-LSTM networks for metamodeling of nonlinear structures. *Comput. Method. Appl. M.* 369, 113226 (2020). doi:10.1016/j.cma.2020.113226
- [31] E. Haghighat, M. Raissi, A. Moure et al., A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Method. Appl. M.* 379, 113741 (2021). doi:10.1016/j.cma.2021.113741
- [32] Q. Zhang, Y. Chen, Z. Yang, Data-driven solutions and discoveries in mechanics using physics-informed neural network. *Preprints*, 060258 (2020). doi:10.20944/preprints202006.0258.v1
- [33] E. Zhang, M. Dao, G.E. Karniadakis et al., Analyses of internal structures and defects in materials using physics-informed neural networks. *Sci. Adv.* 8(7), eabk0644 (2022). doi:10.1126/sciadv.abk0644
- [34] T. Zhu, W. Luo, C. Bu et al., Accelerate population-based stochastic search algorithms with memory for optima tracking on dynamic power systems. *IEEE T. Power Syst.* 31(1), 268–277 (2015). doi:10.1109/TPWRS.2015.2407899
- [35] H. Gong, T. Zhu, Z. Chen et al., Parameter identification and state estimation for nuclear reactor operation digital twin. *Ann. Nucl. Energy* 180, 109497 (2023). doi:10.1016/j.anucene.2022.109497
- [36] Q.H. Ngo, B.L. Nguyen, T.V. Vu et al., Physics-informed graphical neural network for power system state estimation. *Appl. Energ.* 358, 122602 (2024). doi:10.1016/j.apenergy.2023.122602
- [37] M.E. Bento, Physics-guided neural network for load margin assessment of power systems. *IEEE T. Power Syst.* 39(1), 564–575 (2023). doi:10.1109/TPWRS.2023.3266236
- [38] R. Nellikkath, S. Chatzivasileiadis, Physics-informed neural networks for ac optimal power flow. *Electr. Pow. Syst. Res.* 212, 108412 (2022). doi:10.1016/j.epsr.2022.108412
- [39] P.R. Bana, M. Amin, Control for grid-connected VSC with improved damping based on physics-informed neural network. *IEEE J. Emerg. Sel. Top. Ind. Electron.* 4(3), 878–888 (2023). doi:10.1109/jestie.2023.3258339
- [40] Z. He, D. Xiao, L. Lv et al., Stable control of thoron progeny concentration in a thoron chamber for calibration of active sampling monitors. *Radiat. Meas.* 102, 27–33 (2017). doi:10.1016/j.radmeas.2017.02.013
- [41] W. Li, Q. Zhou, Z. He et al., Optimization of the thoron progeny compensation system of a thoron calibration chamber. *J. Radioanal. Nucl. Ch.* 324, 1255–1263 (2020). doi:10.1007/s10967-020-07180-y
- [42] L. Ambrosio, G. D. Maso, A general chain rule for distributional derivatives. *P. Am. Math. Soc.* 108, 691–702 (1990). doi:10.1090/S0002-9939-1990-0969514-3
- [43] W. Wang, M. Yang, The nonlinear flow characteristics within two-

- dimensional and three-dimensional counterflow models within symmetrical structures. *Energies*. 17, 3176 (2024). doi:10.3390/en17133176
- [44] M. Raissi, P. Perdikaris, G. E. Karniadakis et al., Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707 (2019). doi:10.1016/j.jcp.2018.10.045
- [45] C. Rackauckas, Y. Ma, J. Martensen et al., Universal differential equations for scientific machine learning. arXiv:2001.04385 (2020).
- [46] H. Wang, Y. Liu, S. Wang et al., Dense velocity reconstruction from particle image velocimetry/particle tracking velocimetry using a physics-informed neural network. *Phys. Fluids*. 34 (2022). doi:10.1063/5.0078143
- [47] S.H. Rudy, S.L. Brunton, J.L. Proctor et al., Data-driven discovery of partial differential equations. *Sci. Adv.* 3, e1602614 (2017). doi:10.1126/sciadv.1602614
- [48] K. He, X. Zhang, S. Ren et al., Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 770–778 (2016). doi:10.1109/CVPR.2016.90
- [49] P. Ramachandran, B. Zoph, Q.V. Le, Searching for activation functions. arXiv:1710.05941 (2017).
- [50] S. Markidis, The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Front. Big Data*. 4, 669097 (2021). doi:10.3389/fdata.2021.669097

Note: Figure translations are in progress. See original paper for figures.

Source: ChinaXiv –Machine translation. Verify with original.