

# Time/Space Separation-Based Physics-Informed Machine Learning for Spatiotemporal Modeling of Distributed Parameter Systems

Bing-Chuan Wang<sup>✉</sup>, Cong-Ling Dai, Yong Wang<sup>✉</sup>, Senior Member, IEEE, and Han-Xiong Li<sup>✉</sup>, Fellow, IEEE

**Abstract**—This article introduces a novel time/space separation-based physics-informed machine learning (T/S-PIML) modeling method by making full use of the complementary strengths of the physics-informed neural network (PINN) and the time/space separation methodology. T/S-PIML is the first attempt to seamlessly integrate structural (including spatial and temporal) physical information with data for effective spatiotemporal modeling of distributed parameter systems (DPSs). With the help of the spectral method, spatial basis functions are first extracted to capture spatial physical information. Subsequently, a reduced-order system is derived to characterize the corresponding temporal physical information. Upon the structural physical information, PINN is developed for temporal modeling. Following the time/space synthesis, a small amount of sensing data is utilized to calibrate system errors. Experiments on a benchmark DPS and the thermal process of a lithium-ion battery demonstrate the effectiveness of T/S-PIML.

**Index Terms**—Distributed parameter systems, spatiotemporal modeling, time/space separation, physics-informed machine learning.

## NOMENCLATURE

AD	Automatic differentiation.
DNN	Deep neural network.
DPS	Distributed parameter system.
FCNN	Fully connected neural network.
IC	Initial condition.
KL	Karhunen-Loève.
ODE	Ordinary differential equation.
PDE	Partial differential equation.
PIML	Physics-informed machine learning.
PINN	Physics-informed neural network.

Manuscript received 29 March 2024; accepted 9 August 2024. Date of publication 17 October 2024; date of current version 18 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62106287, Grant U23A20347, and Grant 62476290; in part by the GRF Project from RGC of Hong Kong under Grant CityU 11206623; in part by the Hunan Provincial Natural Science Foundation under Grant 2024JJ4072; and in part by the High Performance Computing Center of Central South University. This article was recommended by Associate Editor H.-N. Wu. (*Corresponding author: Yong Wang*)

Bing-Chuan Wang, Cong-Ling Dai, and Yong Wang are with the School of Automation, Central South University, Changsha 410083, China (e-mail: bchwang@csu.edu.cn; 224612140@csu.edu.cn; ywang@csu.edu.cn).

Han-Xiong Li is with the Department of Systems Engineering, City University of Hong Kong, Hong Kong, China (e-mail: mehxli@cityu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TSMC.2024.3443601>.

Digital Object Identifier 10.1109/TSMC.2024.3443601

PRNN	Physics-reinforced neural network.
RMSE	Root mean square error.
s-PINNs	Spectrally adapted physics-informed neural networks.
SBF	Spatial basis function.
T/S-PIML	Time/space separation-based physics-informed machine learning.

## I. INTRODUCTION

MANY industrial processes are distributed parameter systems (DPSs) governed by partial differential equations (PDEs) with mixed or homogeneous boundary conditions [1]. An appropriate mathematical model of a DPS is required in many fields, including system analysis, numerical simulation, and control design. The primary feature of DPSs is that the outputs, process states, and parameters vary in space and time [2]. The time-space coupled nature of DPSs implies that these systems are infinite-dimensional. In summary, it is difficult and complicated to model and control a DPS owing to its time-space coupled dynamics and infinite-dimensional properties.

In recent decades, there has been rapid development in the research on methods for modeling DPSs. However, traditional full-order numerical simulation methods, like some mesh-based methods [3], [4], are computationally expensive, and the complexity of mesh generation for the computational domain poses further challenges for most of these methods when dealing with complex DPSs [5]. Physics-informed machine learning (PIML) [6] is a kind of intelligent method emerging and becoming popular recently. This kind of method combines physical information with data through neural networks or other kernel-based regression networks. The physics-informed neural network (PINN) proposed by Raissi et al. [7] is the most commonly adopted PIML method. It approximates the solution of PDEs by a deep neural network (DNN). Physical constraints, including the governing PDE, boundary conditions, and initial conditions (ICs), are incorporated into the DNN through a physical loss function to facilitate the combination of physical information and data. Note that, many state-of-the-art studies [8], [9] that employ PINN often refer to it as PIML. This kind of method flexibly incorporates underlying physical information and empirical data in a compact way. In addition, unlike traditional numerical methods that require mesh generation, PINN is completely meshless. However, due to the complex characteristics of DPSs, the spatial and

temporal information in the physical information are coupled. Therefore, it is challenging for PINN to obtain the structural information<sup>1</sup> of the DPS under consideration, which causes difficulties in the training of PINN [10]. In particular, when solving complex PDEs, achieving convergence with PINN can be challenging, which requires expanding the scale of a DNN. However, the expansion will result in an increasing computing load and a long training time for a DNN.

Fortunately, the framework of time/space separation can extract spatial physical information from a DPS and construct the corresponding temporal model for the reduced-order/finite-dimensional system [11], [12], [13], [14]. Subsequently, through time/space synthesis, spatial and temporal physical information can be effectively used to recover the spatiotemporal system. The Karhunen–Loëve (KL) method [15], [16], [17], as a purely data-driven time/space separation-based approach, obtains empirical eigenfunctions from numerical or empirical data. Indeed, its accuracy is highly dependent on the data quality and quantity. Thus, a lot of data is usually required to ensure sufficient modeling accuracy. However, acquiring a large volume of data can be challenging in practice. As another representative of time/space separation-based methods, the spectral method [18] is physics-driven, as it extracts spatial basis functions (SBFs) from the governing PDE, boundary conditions, and other relevant conditions. Typical SBFs include Fourier series [19], eigenfunctions [20], [21], orthogonal polynomials [22], etc. However, in practical engineering applications, the actual situation would not precisely align with the first-principles PDE due to environmental disturbances. This will sacrifice the spectral method's accuracy to some degree. For both the KL method and the spectral method, a temporal model can be built by the collocation method [23], [24], Galerkin method [25], [26], and other methods [27]. Recently, machine learning techniques are also often used for temporal model building [28], [29], [30]. As we know, errors can occur between the original system and the reduced-order system approximated through the synthesis of SBFs and the temporal model [8]. In summary, despite effectively extracting structural physical information, time/space separation-based methods exhibit limited modeling accuracy.

Both PINN and time/space separation-based methods possess their respective advantages and disadvantages, which can be effectively leveraged to complement each other. On the one hand, the capability of time/space separation to extract spatial and temporal physical information can significantly ease the training process of PINN; on the other hand, PINN can effectively integrate physical information with data to mitigate system errors. Chen et al. [8] designed the physics-reinforced neural network (PRNN) which combines the projection-based reduced basis method with PINN. However, PRNN requires obtaining high-fidelity snapshots for basis generation and projecting these snapshots into the reduced-order space to obtain labeled coefficients for network training. This process can be

computationally expensive, and in practice, acquiring a large volume of high-fidelity snapshots is nontrivial. Xia et al. [9] proposed spectrally adapted PINNs (s-PINNs) in which the loss function associated with the physical information includes the residuals of PDEs. Although appropriate basis functions based on the form of PDE solutions are embedded into s-PINNs, the information provided by this loss function is still infinite-dimensional. Thus, it is difficult to sufficiently characterize the physical information of PDEs. In addition, the SBFs in this method are not obtained through specific knowledge about the system, making the physical information of DPSs under utilized. Moreover, it is worth noting that both methods primarily focus on solving PDEs rather than specifically addressing the complexity of modeling DPSs, where considerations for control input signals and their impact on the system dynamics become crucial [31].

Based on the aforementioned literature review, several notable insights and the principal research gaps can be summarized as follows.

- 1) The training of the conventional PINN can be challenging due to the difficulty in obtaining the structural physical information of the DPS under consideration.
- 2) Both data-driven and physics-driven time/space separation-based methods exhibit limitations in effectively integrating physical information with data.
- 3) Most of the existing methods that attempt to combine PINN with the time/space separation methodology cannot fully utilize the complementary strengths of these two methodologies. Besides, they have not taken the complexity of modeling DPSs into account.

In summary, the effective integration of PINN with the time/space separation methodology is promising for spatiotemporal modeling of DPSs yet remains an open question.

Based on the above analysis, a novel time/space separation-based PIML (T/S-PIML) modeling method is designed. It seeks to integrate PINN with the time/space separation methodology for spatiotemporal modeling of DPSs. The chosen time/space separation-based method is the spectral method. As described above, the aim of the time/space separation is to extract structural physical information. The spectral method, in contrast to data-driven methods, is able to make full use of the known physical information without requiring data. It is because its SBFs are derived from the governing PDE, boundary conditions, and other relevant conditions. This is a good fit with PINN, as PINN integrates physical information with data efficiently. By embedding the SBFs into the network, it can further reduce the data requirement of PINN. Specifically, T/S-PIML obtains the SBFs of a DPS through the spectral method and reduces the governing PDE to a finite-dimensional temporal system, including time coefficients described by ordinary differential equations (ODEs). In this way, spatial and temporal dynamics are decoupled and a DNN is trained for the time coefficients instead of the infinite-dimensional PDE, which makes training T/S-PIML easier than the conventional PINN. Then, the loss terms associated with physical information are represented by the residuals of the ODEs. The time-related derivatives are obtained by automatic differentiation (AD), and the space-related derivatives are derived directly from the

<sup>1</sup>The structural physical information in this article refers to the spatial and temporal physical information obtained by decoupling the physical information through certain means.

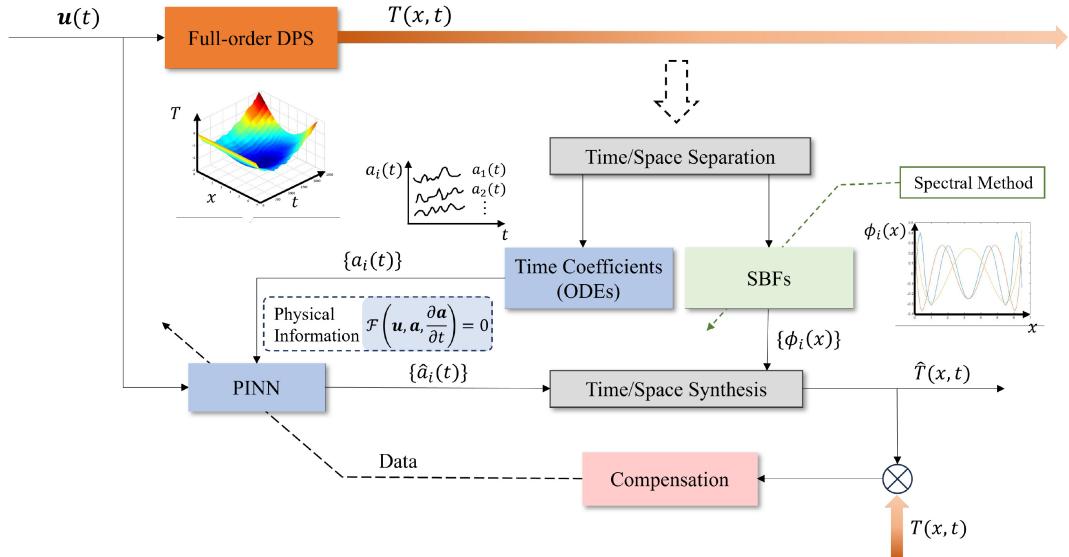


Fig. 1. Framework of T/S-PIML.

SBFs. These derivatives are embedded into the loss terms to train T/S-PIML. Finally, time/space synthesis is conducted and a small amount of sensing data is used for system calibration. The contributions of T/S-PIML are highlighted below.

- 1) T/S-PIML can compactly integrate the complementary strengths of PINN and the time/space separation methodology.
- 2) T/S-PIML can significantly ease the training of the conventional PINN by effectively utilizing the structural physical information of the DPS under consideration.
- 3) T/S-PIML can effectively integrate physical information with data to compensate for system errors.
- 4) T/S-PIML is less dependent on data compared with data-driven methods, making it more practical in real-world applications.

The structure of the article is as follows. The DPSs considered in this study are introduced in Section II. T/S-PIML is elaborated in Section III. Experiments and discussions are carried out in Section IV. In Section V, some concluding remarks are summarized. Furthermore, a succinct proof of the bound of T/S-PIML is provided in the Appendix of the supplementary file.

## II. PROBLEM DESCRIPTION

The classic dissipative DPSs described by the following parabolic PDE are considered:

$$\frac{\partial T(x, t)}{\partial t} = \mathcal{L}\left(T, \frac{\partial T}{\partial x}, \frac{\partial^2 T}{\partial x^2}\right) + \tau b^T(x)u(t) + f(T(x, t)) \quad (1)$$

subject to the mixed-type boundary conditions

$$\mathcal{B}\left(T, \frac{\partial T}{\partial x}, \frac{\partial^2 T}{\partial x^2}\right)\Big|_{x=x_a \text{ or } x=x_b} = 0 \quad (2)$$

and the IC

$$T(x, 0) = T_0(x) \quad (3)$$

where  $x \in \Omega = [x_a, x_b]$  is the spatial coordinate,  $t \in [0, \infty)$  is the temporal variable,  $T(x, t)$  denotes the spatiotemporal output,  $\mathcal{L}$  is a function containing nonlinear spatial differential operators,  $u(t) = [u_1(t), \dots, u_m(t)]^T$  denotes the control input signals distributed by the spatial functions  $b(x) = [b_1(x), \dots, b_m(x)]^T$ ,  $\tau$  is a constant,  $m$  is the number of actuators,  $f(T(x, t))$  represents a function of  $T(x, t)$ ,  $\mathcal{B}$  is a nonlinear function representing the mixed-type boundary conditions, and  $T_0(x)$  is a smooth function to describe the initial output. The PDE in (1) can describe a set of dissipative DPSs, which involve many industrial processes.

In this study, we present a T/S-PIML method for spatiotemporal modeling of DPSs, enabling the effective estimation of the entire spatiotemporal distribution by integrating physical information with only a small amount of sensing data.

## III. METHODOLOGY

### A. Framework

The framework of T/S-PIML is shown in Fig. 1. The time/space separation of the full-order DPS (i.e.,  $T(x, t)$ ) is first performed to extract spatial physical information and temporal physical information. The extraction of spatial physical information is achieved by employing the spectral method on the DPS to obtain a set of SBFs (i.e.,  $\{\phi_i(x)\}$ ). As a result, the full-order DPS is reduced to a reduced-order system consisting of a series of time coefficients (i.e.,  $\{a_i(t)\}$ ) described by a set of ODEs so that temporal physical information is extracted. Once the ODEs have been obtained, the extracted structural physical information can be utilized to develop a temporal model through the application of PINN. Specifically, the inputs of PINN, including  $t$  and  $u(t)$ , are time-dependent only and the outputs are the corresponding time coefficients. The loss terms associated with physical information are the residuals of the ODEs instead of the full-order PDE, as typically used in the conventional PINN. Indeed, this modification effectively addresses the convergence

issue encountered during the training of PINN. Afterward, the output of the full-order DPS is obtained through synthesis of the SBFs and estimations of the temporal model (i.e.,  $\{\hat{a}_i(t)\}$ ). Finally, a small amount of sensing data is compactly used by PINN to mitigate system errors, including errors caused by the gap between the first-principles model and the actual situation, as well as model reduction errors caused by time/space separation.

In the following subsections, we introduce the key components of T/S-PIML, which include the extraction of spatial and temporal physical information, the construction of PINN, and the data compensation for system calibration.

### B. Extraction of Spatial and Temporal Physical Information

According to the time/space separation methodology [32], the spatiotemporal output of a DPS can be expanded by a set of SBFs

$$T(x, t) = \sum_{i=1}^{\infty} a_i(t) \phi_i(x) \quad (4)$$

where  $a_i(t)$  and  $\phi_i(x)$  denote the  $i$ th time coefficient and SBF, respectively. Similar to the Fourier series, the SBFs are commonly arranged from slow to fast in the spatial frequency domain. The spectral method distinguishes between fast and slow modes by constructing a linear or linearized spatial operator based on the form of DPSs, thereby formulating the eigenvalue problem associated with the space [20]. As shown in Fig. 2, according to the magnitudes of its eigenvalues, the eigenspectrum of the spatial operator can be separated into a finite-dimensional part consisting of the first  $N$  slow modes and an infinite-dimensional part containing the remaining fast modes. Then, the Sturm–Liouville theorem [26], [33] is applied to the eigenvalue problem to generate SBFs. Since the fast modes contribute little to the system, only the first  $N$  slowest modes need to be retained in practice. Thus,  $T(x, t)$  can be easily approximated by the truncation as follows:

$$T_N(x, t) = \sum_{i=1}^N a_i(t) \phi_i(x). \quad (5)$$

Eq. (5) is suitable for parabolic PDE systems, since their eigenspectra show a large separation between slow and fast modes. This separation makes it relatively easy to truncate these systems with high accuracy.

In particular, it is assumed that the boundary conditions of the DPS are homogeneous; otherwise, a nonhomogeneity separation [34] is performed as a preliminary step before applying time/space separation, which is introduced in detail in Section IV-B.

1) *Extraction of Spatial Physical Information:* The homogeneous version of  $T(x, t)$  in (1) can be expressed as

$$\frac{\partial T}{\partial t} - K_1 \frac{\partial^2 T}{\partial x^2} - K_2 \frac{\partial T}{\partial x} + K_3 T = 0 \quad (6)$$

where  $K_1$  and  $K_2$  are the constants or space-related functions and  $K_3$  is a constant.

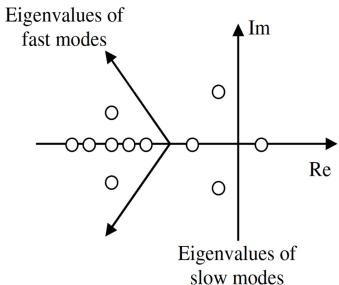


Fig. 2. Separation of eigenspectrum [11].

Combining (4) with (6) gives the sufficient condition

$$\frac{\dot{a}_i(t)}{K_1 a_i(t)} + \frac{K_3}{K_1} = \frac{\ddot{\phi}_i(x)}{\phi_i(x)} + K \frac{\dot{\phi}_i(x)}{\phi_i(x)} = \bar{\lambda}_i = -\lambda_i \quad (7)$$

where  $K = (K_2/K_1)$ ,  $\bar{\lambda}_i < 0$  denotes the eigenvalue, and  $\bar{\Lambda} = \{\bar{\lambda}_i\}$  with  $\bar{\lambda}_1 > \bar{\lambda}_2 > \dots > \bar{\lambda}_N > \dots > \bar{\lambda}_{\infty}$ . (7) can be further transformed into

$$\ddot{\phi}_i(x) + K \dot{\phi}_i(x) + \lambda_i \phi_i(x) = 0. \quad (8)$$

It is the Sturm–Liouville eigenvalue problem [26], [33], and the general solution can be expressed as

$$\phi_i(x) = \gamma_1 H_{1i}(x) + \gamma_2 H_{2i}(x) \quad (9)$$

where  $\gamma_1$  and  $\gamma_2$  are the coefficients,  $H_{1i}(x)$  and  $H_{2i}(x)$  are the functions related to  $\lambda_i$ , and  $\Phi(x) = \{\phi_i(x)\}$  consists of orthogonal eigenfunctions (i.e., SBFs). For a set of desired orthogonal eigenfunctions, further normalization is required. That is to say,  $\Phi(x)$  should satisfy the following conditions:

$$\int \phi_i(x) \phi_j(x) dx = \begin{cases} 0, & i \neq j \\ 1, & i = j. \end{cases} \quad (10)$$

According to (4), (9), (10), and the homogeneous boundary conditions, the values of  $\gamma_1$ ,  $\gamma_2$ , and  $\bar{\Lambda}$  can be found. Finally, the SBFs conveying spatial physical information are obtained.

2) *Extraction of Temporal Physical Information:* For the Sturm–Liouville eigenvalue problem, the eigenvalue  $\bar{\lambda}_i$  represents the frequency corresponding to the eigenfunction  $\phi_i(x)$ . Given  $r$ , a value of  $N$  can be calculated such that it satisfies the following equation:

$$\frac{|\text{Re}(\bar{\lambda}_1)|}{|\text{Re}(\bar{\lambda}_N)|} = \frac{|\text{Re}(\lambda_1)|}{|\text{Re}(\lambda_N)|} \leq r. \quad (11)$$

Then,  $T(x, t)$  can be truncated to a rank- $N$  approximation as shown in (5).

By substituting (5) into (1), we can obtain the residual

$$R(x, t) = \frac{\partial T_N}{\partial t} - \left( \mathcal{L}\left(T_N, \frac{\partial T_N}{\partial x}, \frac{\partial^2 T_N}{\partial x^2}\right) + \tau b^T(x) u(t) + f(T_N) \right). \quad (12)$$

From the Galerkin method, the inner product of the residual and the SBFs is supposed to be zero

$$\int_{\Omega} R(x, t) \phi_i(x) dx = 0, \quad i = 1, \dots, N. \quad (13)$$

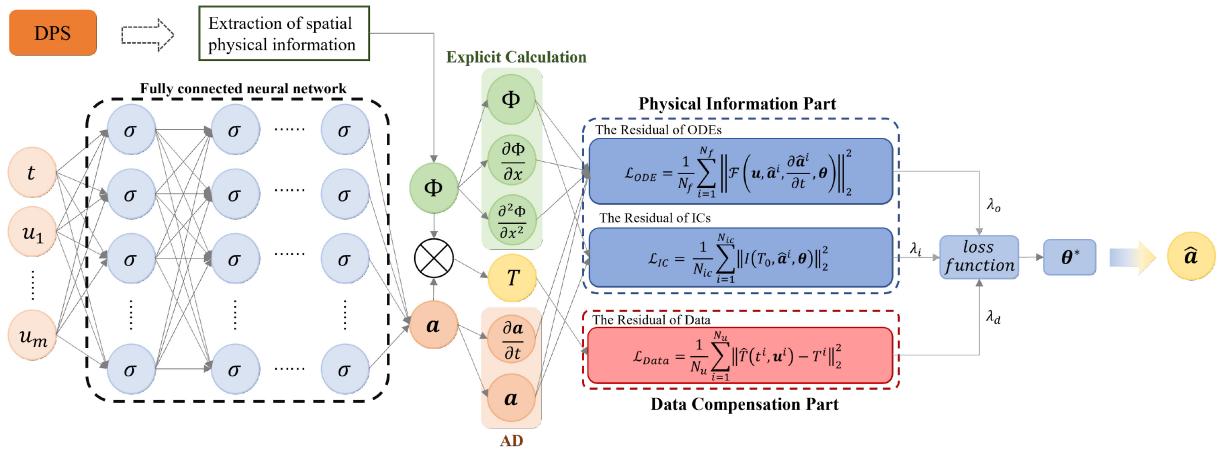


Fig. 3. Schematic of PINN for temporal modeling.  $\sigma$  denotes the nonlinear activation function,  $\lambda_o$ ,  $\lambda_i$ , and  $\lambda_d$  are three weights,  $\theta^*$  denotes the optimized parameters of FCNN, and  $\hat{\mathbf{a}}$  denotes the approximation of  $\mathbf{a}$ .

Subsequently, we yield the ODEs associated with time coefficients. Besides, the IC associated with time coefficients is obtained by substituting (5) into (3). Finally, a reduced-order system for time coefficients can be developed as follows:

$$\mathcal{F}\left(\mathbf{u}, \mathbf{a}, \frac{\partial \mathbf{a}}{\partial t}\right) = 0, \quad t \in [0, \infty) \quad (14)$$

$$\mathcal{I}(T_0, \mathbf{a}) = 0, \quad t = 0 \quad (15)$$

where  $\mathbf{a} = \{a_i(t)\}$  consists of the time coefficients,  $\mathbf{u}$  denotes the time-dependent control input signals,  $T_0$  is the initial value, and  $\mathcal{F}$  and  $\mathcal{I}$  represent the residuals of ODEs and ICs, respectively. Finally, the ODEs can be utilized to establish a temporal model through PINN.

### C. Construction of PINN

The merits of PINN are used to model the ODEs, and the details of this process are illustrated in Fig. 3.

1) *Network Architecture*: It can be seen from Fig. 3 that PINN consists of two components: a) the fully connected neural network (FCNN) part and b) the loss function part. The loss function is composed of a physical information part and a data compensation part. Note that, the main feature of PINN lies in its ability to seamlessly integrate physical information with data in a compact manner, which is not easily achieved by a traditional time/space separation-based method alone. This unique characteristic allows PINN to effectively compensate for system errors. The details of the data compensation will be elaborated in the next section.

- a) In the FCNN part, we leverage the well-known universal approximation capability of DNNs [35]. The inputs of FCNN include the temporal variable (i.e.,  $t$ ) and the time-dependent control signals (i.e.,  $\mathbf{u}$ ). The outputs are the time coefficients (i.e.,  $\mathbf{a} = \{a_i\}$ ).
- b) In the physical information part, the physical information is provided by the residuals of the ODEs and ICs. These residuals are calculated by substituting the SBFs, the outputs of FCNN (i.e., the approximated time coefficients), and their derivatives into the ODEs and ICs. The derivatives of the time coefficients are readily

obtained through AD attributed to the backpropagation rule. Additionally, the SBFs involved in the ODEs have been obtained using the spectral method, allowing their derivatives to be calculated explicitly with high accuracy.

2) *Physical Loss Function*: The outputs of FCNN should satisfy governing ODEs and ICs

$$\mathcal{F}\left(\mathbf{u}, \hat{\mathbf{a}}, \frac{\partial \hat{\mathbf{a}}}{\partial t}, \theta\right) = 0, \quad t \in [0, \infty) \quad (16)$$

$$\mathcal{I}(T_0, \hat{\mathbf{a}}, \theta) = 0, \quad t = 0 \quad (17)$$

where  $\theta$  denotes the hyperparameters of FCNN and  $\hat{\mathbf{a}}$  denotes the approximation of  $\mathbf{a}$ . To this end, the physical loss function (denoted as  $\mathcal{L}_{\text{physics}}$ ) is derived from the residuals of the ODEs and ICs

$$\mathcal{L}_{\text{physics}} = \lambda_o \mathcal{L}_{\text{ODE}} + \lambda_i \mathcal{L}_{\text{IC}} \quad (18)$$

$$\mathcal{L}_{\text{ODE}} = \frac{1}{N_f} \sum_{i=1}^{N_f} \left\| \mathcal{F}\left(\mathbf{u}, \hat{\mathbf{a}}^i, \frac{\partial \hat{\mathbf{a}}^i}{\partial t}, \theta\right) \right\|_2^2 \quad (19)$$

$$\mathcal{L}_{\text{IC}} = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} \left\| \mathcal{I}\left(T_0, \hat{\mathbf{a}}^i, \theta\right) \right\|_2^2 \quad (20)$$

where  $\mathcal{L}_{\text{ODE}}$  and  $\mathcal{L}_{\text{IC}}$  denote the ODE loss and the IC loss, respectively,  $\lambda_o$  and  $\lambda_i$  denote the weights of these two losses, respectively,  $\hat{\mathbf{a}}^i$  includes the outputs of FCNN at the  $i$ th collocation point,  $N_f$  and  $N_{ic}$  represent the number of collocation points selected from the computational domains of the ODEs and ICs, respectively, and  $\|\cdot\|_2$  denotes the  $L_2$  norm operator.

In summary, by training FCNN using the physical loss function, the physical information is effectively incorporated into the model.

### D. Data Compensation for System Calibration

In the time/space separation framework, the approximation of the full-order DPS is achieved through time/space synthesis of the SBFs and the outputs of FCNN. However, not only is there a gap between the original system and the actual situation but this approximation will also suffer from model reduction errors. Generally speaking, sampled data of the DPS can be used for compensation. PINN is well-known for its capability

to effectively combine physical information with data through the formulation of a loss function. This particular advantage is used here for system calibration. Specifically, the data loss function (denoted as  $\mathcal{L}_{\text{Data}}$ ) is formulated as follows:

$$\mathcal{L}_{\text{Data}} = \frac{1}{N_u} \sum_{i=1}^{N_u} \|\widehat{T}(t^i, \mathbf{u}^i) - T^i\|_2^2 \quad (21)$$

$$\widehat{T}(t^i, \mathbf{u}^i) = \sum_{j=1}^N \hat{a}_j(t^i, \mathbf{u}^i) \phi_j(x) \quad (22)$$

where  $N_u$  denotes the number of the sampled data points,  $T^i$  is the  $i$ th sampled data point,  $\hat{a}_j(t^i, \mathbf{u}^i)$  is the  $j$ th output of FCNN at the  $i$ th sampled data point, and  $\widehat{T}(t^i, \mathbf{u}^i)$  denotes the corresponding spatiotemporal output.

Subsequently, (21) can be embedded into the physical loss function for system calibration

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{\text{physics}} + \lambda_d \mathcal{L}_{\text{Data}} \\ &= \lambda_o \mathcal{L}_{\text{ODE}} + \lambda_i \mathcal{L}_{\text{IC}} + \lambda_d \mathcal{L}_{\text{Data}} \end{aligned} \quad (23)$$

where  $\mathcal{L}$  is the loss function and  $\lambda_d$  is the weight of the data compensation part (i.e., the data loss function).

In summary, PINN effectively integrates physical information with data by constructing an appropriate loss function. Finally, by minimizing the loss function, the parameters of FCNN (i.e.,  $\theta$ ) are obtained and the spatiotemporal output can be estimated.

### E. Main Advantages of T/S-PIML

- 1) Compared with data-driven time/space separation-based methods (e.g., the KL method), T/S-PIML can achieve higher accuracy even though less data is used for model training. The main reason is that T/S-PIML is able to effectively extract physical information for model training, which can significantly alleviate the dependence on data.
- 2) Compared with physics-driven time/space separation-based methods (e.g., the spectral method), T/S-PIML can obtain better performance when the same model order is used. It is because T/S-PIML can compactly incorporate data loss into physical loss to compensate for system errors.
- 3) Compared with PINN, T/S-PIML achieves higher accuracy and faster convergence. The key advantage lies in T/S-PIML's ability to extract spatial and temporal physical information through time/space separation, facilitating efficient network training. In contrast, PINN struggles to obtain the same level of structural physical information.
- 4) Compared with the existing methods (e.g., s-PINNs) that attempt to combine PINN with the time/space separation methodology, T/S-PIML achieves higher accuracy with less data. It is due to the fact that T/S-PIML can fully exploit the complementary strengths of these two methodologies. In particular, the SBFs derived from the spectral method can utilize the specific system knowledge to further promote the utilization of physical information, thus reducing data requirement.

In summary, the proposed method can achieve better accuracy and efficiency. Note that, we give a concise proof of the bound of T/S-PIML in the Appendix of the supplementary file.

### IV. EXPERIMENTS

The proposed T/S-PIML constructed an FCNN to model DPSs. Specific network settings were summarized as follows.

- 1) The network had five hidden layers, each with 80 neurons.
- 2) The hyperbolic tangent activation function (i.e., tanh) was used as the activation function.
- 3) As the same in [7], all the experiments used L-BFGS and Adam as optimizers, and the learning rates were set to 1.0 and 0.001, respectively.

Two complex DPSs were simulated to verify the effectiveness of T/S-PIML. The first DPS was studied as a benchmark example, representing a standard problem for performance evaluation. The second DPS was the thermal process of a lithium-ion battery, representing a real-world application with practical significance. The performance of T/S-PIML was compared with that of four widely used spatiotemporal modeling methods: 1) the KL method [36], 2) the spectral method [37], 3) the conventional PINN [38], and 4) s-PINNs [9]. The order  $M$  of the KL method was defined by selecting the first  $M$  largest eigenvalues, such that the ratio of these eigenvalues to the sum of all eigenvalues is as large as possible. The experiments in this article assumed that the ratio was 99.99%, which was considered to be sufficient for capturing the essential spatial modes that represent the original system well. The order of the spectral method was consistent with T/S-PIML. The network structures of both the conventional PINN and s-PINNs, as well as the amount of data in PINN, were the same as in T/S-PIML. In addition, the root mean-square error (RMSE) was adopted for performance comparison

$$\text{RMSE} = \sqrt{\frac{1}{N_s} \sum_{i=1}^{N_s} (\mathcal{D}_i^{\text{ref}} - \widehat{\mathcal{D}}_i)^2} \quad (24)$$

where  $\mathcal{D}_i^{\text{ref}}$  and  $\widehat{\mathcal{D}}_i$  are the  $i$ th sampled and estimated data points, respectively, and  $N_s$  denotes the size of the data set. Note that, sensor placement is an important topic in the field of spatiotemporal modeling [39]. In our experiments, we adopted uniform placement as it is considered as a simple and effective way to capture the dynamic characteristics [40]. The further discussions were conducted as follows.

#### A. Case 1: Benchmark DPS

We considered a nonlinear one-dimensional thermal process with the following PDE:

$$\frac{\partial T(x, t)}{\partial t} = \frac{\partial^2 T(x, t)}{\partial x^2} + \beta_u [\mathbf{b}^T(x) \mathbf{u}(t) - T(x, t)] \quad (25)$$

subject to the Dirichlet boundary conditions

$$T(0, t) = 0, T(\pi, t) = 0 \quad (26)$$

and the IC

$$T(x, 0) = 0 \quad (27)$$

where  $x \in \Omega = [0, \pi]$  gives the spatial coordinate,  $T(x, t)$  gives the output temperature ( $^{\circ}\text{C}$ ),  $\beta_u$  denotes the dimensionless heat transfer coefficient,  $\mathbf{u}(t) = [u_1(t), u_2(t), u_3(t), u_4(t)]^T$

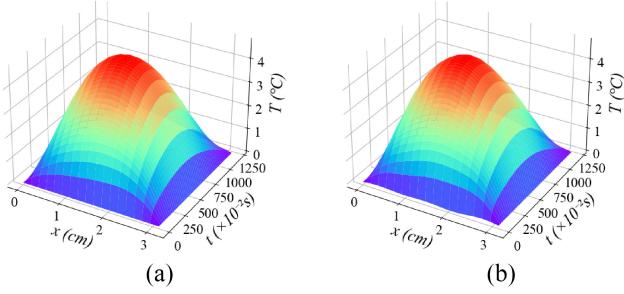


Fig. 4. Estimated results of T/S-PIML for modeling the benchmark DPS. (a) Measured temperature. (b) Estimated temperature.

with  $u_i(t) = 1.1 + 5 \sin(t/4 + i/10)$  ( $i = 1, 2, 3, 4$ ) includes the control input signals, and  $\mathbf{b}(x) = [b_1(x), b_2(x), b_3(x), b_4(x)]^T$  denotes a set of the spatial distribution functions, with  $b_i(x) = H(x - (i-1)\pi/4) - H(x - i\pi/4)$  and  $H(\cdot)$  being the standard Heaviside step function.

Eighteen sensing points were evenly located along the spatial coordinate for data collection, each sampling 1200 snapshots. The sampling interval was  $\Delta t = 0.01$  s. Note that, only the data from two specific sensing points was selected for training, while the data from the remaining sensing points was used for validation.

1) *Model Construction:* Following the design method of SBFs outlined in Section III-B, the general solution form was first derived. In Case 1,  $K_2 = 0$ , then  $K$  became zero. As a result, the general solution was as follows:

$$\phi_i(x) = \gamma_1 \cos(\sqrt{\lambda_i}x) + \gamma_2 \sin(\sqrt{\lambda_i}x). \quad (28)$$

Subsequently, the SBFs were derived as follows:

$$\begin{cases} \phi_i(x) = \sqrt{\frac{2}{\pi}} \sin \sqrt{\lambda_i}x, & i = 1, 2, \dots, \infty. \\ \lambda_i = i^2 \end{cases} \quad (29)$$

Let  $r = 0.05$ , and according to (11), the order  $N$  was calculated as  $N = 5$ .

Next, the residual was expressed as

$$R(x, t) = \frac{\partial T_N(x, t)}{\partial t} - \left( \frac{\partial^2 T_N(x, t)}{\partial x^2} + \beta_u (\mathbf{b}^T(x) \mathbf{u}(t) - T_N(x, t)) \right). \quad (30)$$

With the Galerkin method, it could be obtained

$$\dot{a}_i(t) = \bar{P}_{0i} a_i(t) + \bar{P}_{1i} u_1(t) + \bar{P}_{2i} u_2(t) + \bar{P}_{3i} u_3(t) + \bar{P}_{4i} u_4(t) \quad (31)$$

where  $\bar{P}_{0i} = \bar{\lambda}_i - \beta_u = -\lambda_i - \beta_u$ ,  $\bar{P}_{1i} = \int \beta_u b_1(x) \phi_i(x) dx$ ,  $\bar{P}_{2i} = \int \beta_u b_2(x) \phi_i(x) dx$ ,  $\bar{P}_{3i} = \int \beta_u b_3(x) \phi_i(x) dx$ , and  $\bar{P}_{4i} = \int \beta_u b_4(x) \phi_i(x) dx$  ( $i = 1, 2, \dots, N$ ).

Finally, the reduced-order system described by ODEs could be expressed as

$$\dot{\mathbf{a}}(t) = \bar{\mathbf{P}}_0 \mathbf{a}(t) + \bar{\mathbf{P}}_1 u_1 + \bar{\mathbf{P}}_2 u_2 + \bar{\mathbf{P}}_3 u_3 + \bar{\mathbf{P}}_4 u_4 \quad (32a)$$

$$T_N(x, t) = \Phi \mathbf{a}(t) \quad (32b)$$

where  $\Phi = [\phi_1(x), \phi_2(x), \dots, \phi_N(x)]$ ,  $\mathbf{a}(t) = [a_1(t), a_2(t), \dots, a_N(t)]^T$ ,  $\bar{\mathbf{P}}_0 = \text{diag}(\bar{P}_{01}, \bar{P}_{02}, \dots, \bar{P}_{0N})$ ,  $\bar{\mathbf{P}}_1 = [\bar{P}_{11}, \bar{P}_{12}, \dots]$

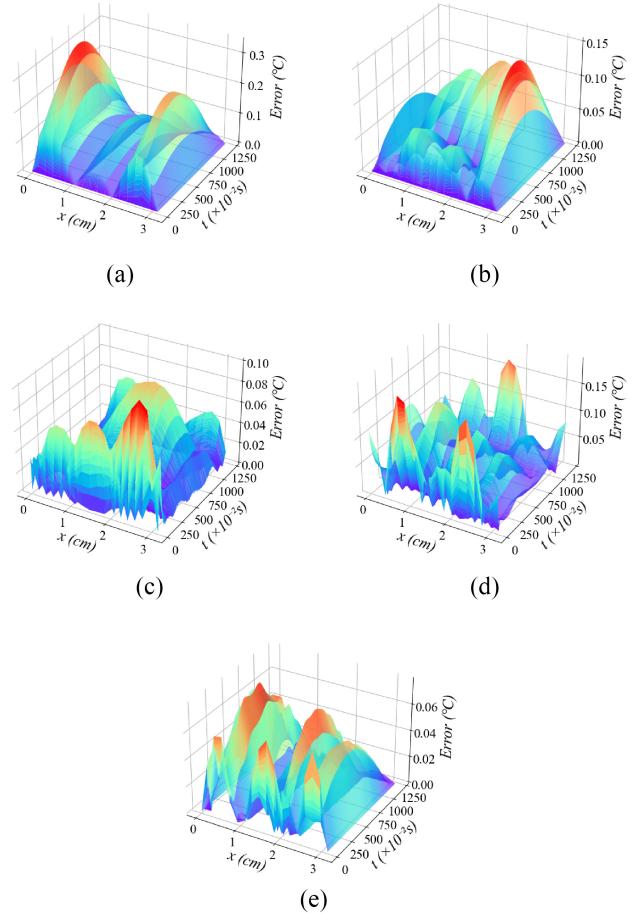


Fig. 5. Estimation errors of different methods for modeling the benchmark DPS. (a) The KL method. (b) the spectral method. (c) PINN. (d) s-PINNs. (e) T/S-PIML.

$\dots, \bar{P}_{1N}]^T$ ,  $\bar{\mathbf{P}}_2 = [\bar{P}_{21}, \bar{P}_{22}, \dots, \bar{P}_{2N}]^T$ ,  $\bar{\mathbf{P}}_3 = [\bar{P}_{31}, \bar{P}_{32}, \dots, \bar{P}_{3N}]^T$ , and  $\bar{\mathbf{P}}_4 = [\bar{P}_{41}, \bar{P}_{42}, \dots, \bar{P}_{4N}]^T$ .

As described in Sections III-C and III-D, the reduced-order system was modeled by PINN, and the data from two sensing points was selected for system calibration.

2) *Results of Spatiotemporal Modeling:* After the training process, T/S-PIML can produce a spatially continuous output, as shown in Fig. 4(b). It agrees well with the measured temperature described in Fig. 4(a). It can be seen from Fig. 5(e) that the estimation error oscillates in a small range with the peak not exceeding 0.06. Table I shows that the RMSE of T/S-PIML in the whole domain is 0.033.

3) *Performance Comparison:* To further verify the performance of T/S-PIML, we compared it with four existing methods: 1) the KL method, 2) the spectra method, 3) the conventional PINN, and 4) s-PINNs. The KL method had an order of 3 and used data from five sensing points for training. The order of the spectral method was 5. Chebyshev polynomials were selected as the SBFs in s-PINNs, so the sampling points needed to be calculated according to the roots of the polynomials. Note that, s-PINNs used data from three sensing points for training. The estimation errors and RMSEs are given in Fig. 5 and Table I, respectively. As shown in Table I, T/S-PIML demonstrates an accuracy improvement

TABLE I  
RMSES OF DIFFERENT METHODS FOR MODELING  
THE BENCHMARK DPS

	KL	Spectral	PINN	s-PINNs	T/S-PIML
RMSE	0.136	0.064	0.052	0.045	<b>0.033</b>

TABLE II  
TIME COSTS OF DIFFERENT METHODS FOR MODELING  
THE BENCHMARK DPS

	KL	Spectral	PINN	s-PINNs	T/S-PIML
Running (ms)	170.953	12881.781	4.891	2.374	<b>2.161</b>

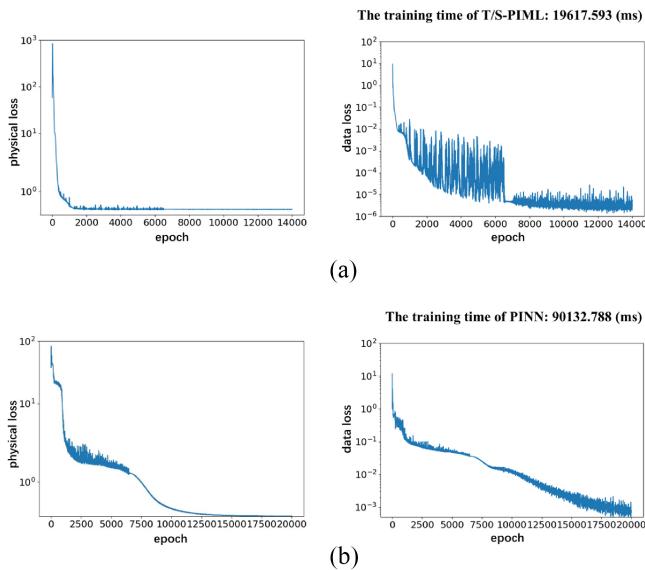


Fig. 6. Convergence curves of training losses for modeling the benchmark DPS. (a) Physical loss and data loss of T/S-PIML. (b) Physical loss and data loss of PINN.

of 75.74%, 48.44%, 36.54%, and 26.67% over the KL method, the spectral method, PINN, and s-PINNs, respectively. Thus, T/S-PIML exhibits higher accuracies than the other four methods. The running time in Table II shows that T/S-PIML is much faster than the KL method and the spectral method, and slightly faster than PINN and s-PINNs. The reason is that T/S-PIML, PINN, and s-PINNs adopt the deep learning structure. Once trained, these methods run faster than the KL and spectral methods. Additionally, T/S-PIML uses less data than the KL method and s-PINNs but shows superior performance, thus demonstrating the applicability of this method in scenarios with limited data availability. Moreover, we compared the convergence performance of T/S-PIML with that of PINN. Fig. 6 shows that T/S-PIML converges faster than PINN in terms of both physical loss and data loss, and the training time required for T/S-PIML is much less than that of PINN. In conclusion, the results indicate that T/S-PIML is efficient and effective for modeling the benchmark DPS.

TABLE III  
NOMINAL PARAMETERS OF THE 18650 LITHIUM-ION BATTERY

Parameter	Value
Length	65 mm
Diameter	18 mm
Voltage	3.2 V
Capacity	3.5 Ah
Internal Resistance	24 mΩ
Positive/negative electrode	Li(NiCoMn)O <sub>2</sub> /graphite

### B. Case 2: Thermal Process of the Lithium-Ion Battery

An 18650 cylindrical lithium-ion battery was used for the experimental study [41]. Table III lists the nominal parameters of the battery, which were obtained from the instructions provided by the battery manufacturer. Other parameters that are difficult to obtain directly (e.g., cell density, heat capacity, and convection coefficient) can be identified by experiments [42]. We considered the thermal process of the cylindrical lithium-ion battery, which is described by a parabolic PDE

$$\rho Cp \frac{\partial T(x, t)}{\partial t} = k_x \frac{\partial^2 T(x, t)}{\partial x^2} + q \quad (33)$$

subject to the Robin-type nonhomogeneous boundary conditions

$$-k_x \frac{\partial T}{\partial x} \Big|_{x=0} = -h_c(T - T_{air}) \quad (34a)$$

$$-k_x \frac{\partial T}{\partial x} \Big|_{x=x_0} = h_c(T - T_{air}) \quad (34b)$$

and the IC

$$T(x, 0) = T_{air} \quad (35)$$

where  $x \in \Omega = [0, x_0]$  gives the spatial coordinate with  $x_0$  as the length of the cylindrical battery,  $T(x, t)$  denotes the temperature distribution,  $k_x$ ,  $\rho$ , and  $Cp$  are the thermal conductivity, density, and heat capacity of the battery, respectively,  $h_c$  is the heat exchange coefficient,  $T_{air}$  is the ambient temperature, and  $q$  denotes the heat generation rate and is approximated as

$$q = b(x)I \left( E_{oc} - V - T \cdot \frac{\partial E_{oc}}{\partial T} \right). \quad (36)$$

In (36),  $I$ ,  $V$ , and  $E_{oc}$  are the current, terminal voltage, and open circuit potential, respectively,  $b(x)$  denotes the spatial distribution of the current density, and  $(\partial E_{oc}/\partial T)$  is the entropy coefficient that is assumed to be a constant.

A simulated model was first constructed in COMSOL multiphysics. Next, 20 sensing points were evenly distributed on the battery for spatiotemporal data acquisition. For each sensing point, 2400 temperature snapshots were collected with a sampling interval of  $\Delta t = 1$  s. The data from only two sensing points was selected for training, while the remaining data was used for validation.

1) *Model Construction:* Since the boundary conditions of the thermal process were nonhomogeneous, nonhomogeneity separation [34] was first performed. For simplicity, the PDE and boundary conditions were organized as

$$\frac{\partial T(x, t)}{\partial t} = k_0 \frac{\partial^2 T(x, t)}{\partial x^2} + k_1 b(x) \cdot I\left(E_{oc} - V - T \cdot \frac{\partial E_{oc}}{\partial T}\right) \quad (37)$$

$$\left[ \frac{\partial T}{\partial x} - hT \right]_{x=0} = -hT_{air} \quad (38a)$$

$$\left[ \frac{\partial T}{\partial x} - hT \right]_{x=x_0} = hT_{air} \quad (38b)$$

where  $k_0 = (k_x/\rho Cp)$ ,  $k_1 = (1/\rho Cp)$ , and  $h = (h_c/k_x)$ .

In order to separate the interference of nonhomogeneity from (33),  $T(x, t)$  was decomposed into  $U(x, t)$  and an auxiliary function  $v(x, t)$

$$T(x, t) = U(x, t) + v(x, t) \quad (39)$$

$$\left[ \frac{\partial v}{\partial x} + \beta_1 v \right]_{x=0} = \eta_1, \left[ \frac{\partial v}{\partial x} + \beta_2 v \right]_{x=x_0} = \eta_2 \quad (40a)$$

$$\left[ \frac{\partial U}{\partial x} + \beta_1 U \right]_{x=0} = 0, \left[ \frac{\partial U}{\partial x} + \beta_2 U \right]_{x=x_0} = 0 \quad (40b)$$

where  $\beta_1 = -h$ ,  $\beta_2 = h$ ,  $\eta_1 = -hT_{air}$ , and  $\eta_2 = hT_{air}$ .

We assumed that  $v(x, t)$  exhibited the following form:

$$v(x, t) = A_v(t)x + C_v. \quad (41)$$

Substituting (41) into (40a) yielded the following system of equations:

$$\begin{cases} A_v + \beta_1 C_v = \eta_1 \\ (1 + \beta_2 x_0)A_v + \beta_2 C_v = \eta_2 \end{cases} \quad (42)$$

with solutions  $A_v = 0$  and  $C_v = T_{air}$ . As a result,  $v(x, t)$  could be derived

$$v(x, t) = T_{air}. \quad (43)$$

By substituting (39) and (43) into (37), the original PDE could be transformed into the following new PDE with homogeneous boundary conditions:

$$\frac{\partial U(x, t)}{\partial t} = k_0 \frac{\partial^2 U(x, t)}{\partial x^2} + k_1 b(x) \cdot I\left(E_{oc} - V - (U + T_{air}) \cdot \frac{\partial E_{oc}}{\partial T}\right). \quad (44)$$

For time/space separation,  $U(x, t)$  could be expanded by a set of SBFs

TABLE IV  
VALUES OF  $\lambda_i$

$i$	1	2	3	4	5	6	...
$\lambda_i$	0.0395	0.3174	1.0240	2.1934	3.8292	5.9318	...

$$U(x, t) = \sum_{i=1}^{\infty} a_i(t) \phi_i(x) \quad (45)$$

and could be further approximatively truncated as

$$U_N(x, t) = \sum_{i=1}^N a_i(t) \phi_i(x). \quad (46)$$

According to the design method of SBFs described in Section III-B, the general solution form was first derived. In Case 2,  $K_2 = 0$ , then  $K$  became zero. As a result, the general solution was as follows:

$$\phi_i(x) = \gamma_1 \cos(\sqrt{\lambda_i}x) + \gamma_2 \sin(\sqrt{\lambda_i}x). \quad (47)$$

By substituting (45) and (47) into (40b), the following equations were obtained:

$$\begin{cases} -h\gamma_1 + \sqrt{\lambda_i}\gamma_2 = 0 \\ (h \cos \sqrt{\lambda_i}x_0 - \sqrt{\lambda_i} \sin \sqrt{\lambda_i}x_0)\gamma_1 + (h \sin \sqrt{\lambda_i}x_0 + \sqrt{\lambda_i} \cos \sqrt{\lambda_i}x_0)\gamma_2 = 0. \end{cases} \quad (48)$$

According to (10), (47), and (48), the values of  $\gamma_1$  and  $\gamma_2$  could be obtained. Thus, the  $i$ th SBF could be derived

$$\phi_i(x) = \chi_i (\sqrt{\lambda_i} \cos \sqrt{\lambda_i}x + h \sin \sqrt{\lambda_i}x) \quad (49)$$

where  $\chi_i = \sqrt{(1/[h + (1/2)(h^2 + \lambda_i)x_0])} \neq 0$  is a constant coefficient.

To ensure the existence of nonzero solutions for (48), the determinant of coefficients in this equation should be equal to zero; thus, the following transcendental equation could be derived:

$$\cot \xi = \frac{1}{2} \left( \frac{\xi}{\mu} - \frac{\mu}{\xi} \right) \quad (50)$$

where  $\xi = \sqrt{\lambda_i}x_0$  and  $\mu = hx_0$ .

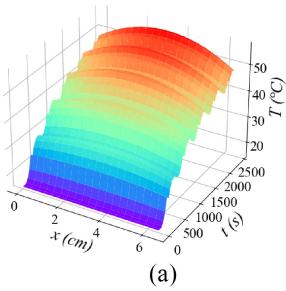
The values of  $\lambda_i$  could be found by dichotomy and are listed in Table IV. Let  $r = 0.01$ , and according to (11), the order  $N$  was calculated as  $N = 5$ . Next, the residual was expressed as

$$R(x, t) = \frac{\partial U_N(x, t)}{\partial t} - \left( k_0 \frac{\partial^2 U_N(x, t)}{\partial x^2} + k_1 b(x) \cdot I\left(E_{oc} - V - (U_N + T_{air}) \cdot \frac{\partial E_{oc}}{\partial T}\right) \right). \quad (51)$$

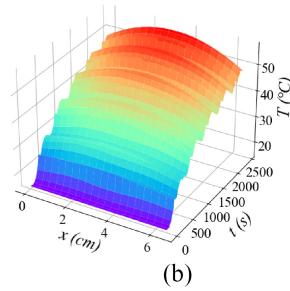
With the Galerkin method, it could be obtained

$$\dot{a}_i(t) = \bar{k}_{0i} a_i(t) + \bar{k}_{1i} I(t) (E_{oc} - V(t)) + \bar{k}_{2i} I(t) a_i(t) + \bar{k}_{3i} I(t) \quad (52)$$

where  $\bar{k}_{0i} = k_0 \bar{\lambda}_i = -k_0 \lambda_i$ ,  $\bar{k}_{1i} = \int k_1 b(x) \phi_i(x) dx$ ,  $\bar{k}_{2i} = -k_1 b(x) (\partial E_{oc}/\partial T)$ , and  $\bar{k}_{3i} = -\int k_1 b(x) T_{air} (\partial E_{oc}/\partial T) \phi_i(x) dx$ .



(a)



(b)

Fig. 7. Estimated result of T/S-PIML for modeling the thermal process of the lithium-ion battery. (a) Measured temperature. (b) Estimated temperature.

( $i = 1, 2, \dots, N$ ). Note that,  $b(x)$  was assumed to be a constant here for simplicity.

Finally, the reduced-order system of the thermal process described by ODEs could be expressed as

$$\dot{\mathbf{a}}(t) = \bar{\mathbf{k}}_0 \mathbf{a}(t) + \bar{\mathbf{k}}_1 I(E_{\text{oc}} - V) + \bar{\mathbf{k}}_2 I \cdot \mathbf{a}(t) + \bar{\mathbf{k}}_3 I \quad (53a)$$

$$U_N(x, t) = \Phi \mathbf{a}(t) \quad (53b)$$

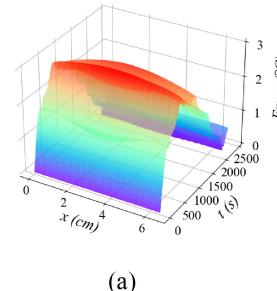
$$T_N(x, t) = U_N(x, t) + \omega(x, t) \quad (53c)$$

where  $\Phi = [\phi_1(x), \phi_2(x), \dots, \phi_N(x)]$ ,  $\mathbf{a}(t) = [a_1(t), a_2(t), \dots, a_N(t)]^T$ ,  $\bar{\mathbf{k}}_0 = \text{diag}(\bar{k}_{01}, \bar{k}_{02}, \dots, \bar{k}_{0N})$ ,  $\bar{\mathbf{k}}_1 = \text{diag}(\bar{k}_{11}, \bar{k}_{12}, \dots, \bar{k}_{1N})$ ,  $\bar{\mathbf{k}}_2 = \text{diag}(\bar{k}_{21}, \bar{k}_{22}, \dots, \bar{k}_{2N})$ , and  $\bar{\mathbf{k}}_3 = \text{diag}(\bar{k}_{31}, \bar{k}_{32}, \dots, \bar{k}_{3N})$ .

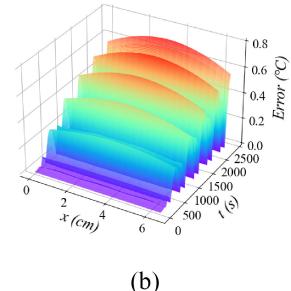
As described in Sections III-C and III-D, the reduced-order system was modeled by PINN, and the data from two sensing points was selected for system calibration.

*2) Results of Spatiotemporal Modeling:* Once T/S-PIML has been trained, it was used to estimate the spatiotemporal distribution of battery temperature, as shown in Fig. 7(b). The estimated temperature closely aligns with the measured temperature. As can be seen from Fig. 8(e), estimation errors at most locations are lower than 0.5. Moreover, as shown in Table V, the RMSE in the whole domain is 0.19, which is considered acceptable for practical utilization. It can be concluded that T/S-PIML is able to estimate the thermal process of the lithium-ion battery accurately.

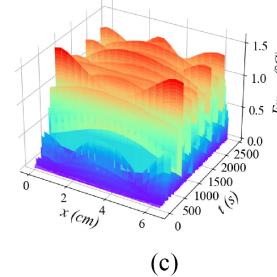
*3) Performance Comparison:* T/S-PIML was compared with the KL method, the spectral method, the conventional PINN, and s-PINNs. The estimation errors and RMSEs of these five methods are shown in Fig. 8 and Table V, respectively. As shown in Table V, T/S-PIML demonstrates an accuracy improvement of 89.62%, 36.67%, 53.66%, and 47.22% over the KL method, the spectral method, PINN, and s-PINNs, respectively. Compared with the other four methods, T/S-PIML exhibits lower estimation errors at most locations and achieves the lowest RMSE. The running time of different methods in Table VI shows that T/S-PIML is much faster than the KL method and the spectral method, and slightly faster than PINN and s-PINNs. The reason is that T/S-PIML, PINN, and s-PINNs adopt the deep learning structure. Once trained, these methods run faster than the KL method and the spectral method. Additionally, T/S-PIML uses less data than the KL method and s-PINNs but shows superior performance, thus



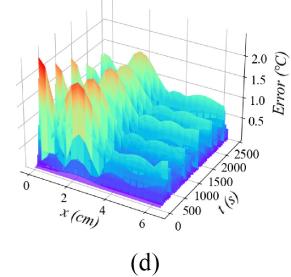
(a)



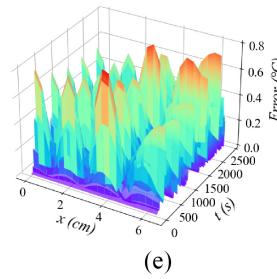
(b)



(c)



(d)



(e)

Fig. 8. Estimation errors of different methods for modeling the thermal process of the lithium-ion battery. (a) the KL method. (b) the spectral method. (c) PINN. (d) s-PINNs. (e) T/S-PIML.

TABLE V  
RMSES OF DIFFERENT METHODS FOR MODELING THE THERMAL PROCESS OF THE LITHIUM-ION BATTERY

	KL	Spectral	PINN	s-PINNs	T/S-PIML
RMSE	1.83	0.30	0.41	0.36	<b>0.19</b>

TABLE VI  
TIME COSTS OF DIFFERENT METHODS FOR MODELING THE THERMAL PROCESS OF THE LITHIUM-ION BATTERY

	KL	Spectral	PINN	s-PINNs	T/S-PIML
Running (ms)	857.406	987.457	77.159	3.587	<b>2.712</b>

demonstrating its potential for application in scenarios where only limited data is available. Finally, the convergence curves of T/S-PIML and PINN are described in Fig. 9 to compare their performance. It is obvious that T/S-PIML converges faster and requires less training time than PINN. In conclusion, the results indicate that T/S-PIML is efficient and effective for modeling the thermal process of the lithium-ion battery.

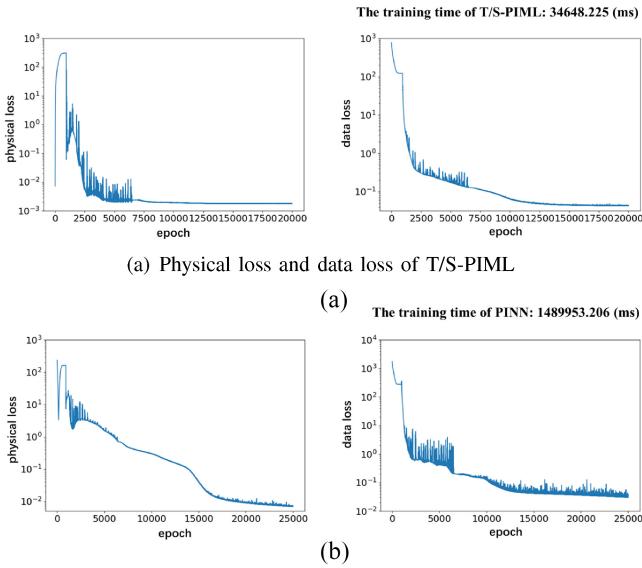


Fig. 9. Convergence curves of training losses for modeling the thermal process of the lithium-ion battery. (a) Physical loss and data loss of T/S-PIML. (b) Physical loss and data loss of PINN.

## V. CONCLUSION

We presented a novel approach called T/S-PIML that combines physical information with data through deep learning within the framework of time/space separation for spatiotemporal modeling of a DPS. First, we obtained the SBFs that convey the spatial physical information of the DPS using the spectral method. Consequently, the governing PDE was transformed into ODEs that include the temporal physical information of the system. Subsequently, PINN was constructed for temporal modeling. PINN was trained using the extracted structural (including spatial and temporal) physical information. Finally, time/space synthesis was performed, and a small amount of sensing data was used for system calibration. Thanks to the fusion of the complementary strengths of PINN and the time/space separation methodology, T/S-PIML is suitable for scenarios with limited data availability, and is characterized by fast convergence and high estimation accuracy. And for the scenario where physical information is available, the more the physical information, the less the data required. Additionally, no matter how complex the DPSs are, there will be some basic physical information. Combined with some data, an accurate model can also be obtained. Extensive experiments verified that T/S-PIML performed better than the KL method, the spectral method, the conventional PINN, and s-PINNs. In general, T/S-PIML showed great potential in modeling DPSs. Moreover, a succinct proof of the bound of T/S-PIML was given in the Appendix of the supplementary file.

Despite the above advantages of the proposed method, the chosen spectral method implements a truncation to approximate the original system by retaining the first  $N$  slowest modes. To ensure that the first  $N$  slowest modes representing the whole system can be found and  $N$  is not too big, T/S-PIML is suitable for a system with a clearly separated eigenspectrum, while it may face challenges

when applied to the systems without obvious separation of eigenspectra. In the future, efforts will be devoted to extending T/S-PIML to model DPSs governed by more complex PDEs.

Additionally, improving the performance of T/S-PIML by optimizing sensor placement is the next issue we have to consider.

## REFERENCES

- [1] K. S. Miller, *Partial Differential Equations in Engineering Problems*. Mineola, NY, USA: Dover Publ., 2020.
- [2] P. D. Christofides and J. Chow, "Nonlinear and robust control of PDE systems: Methods and applications to transport-reaction processes," *Appl. Mech. Rev.*, vol. 55, no. 2, pp. B29–B30, 2002.
- [3] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, vol. 22. New York, NY, USA: Springer, 2013.
- [4] T. Basak, A. K. Singh, R. Richard, and S. Roy, "Finite element simulation with heatlines and entropy generation minimization during natural convection within porous tilted square cavities," *Ind. Eng. Chem. Res.*, vol. 52, no. 23, pp. 8046–8061, 2013.
- [5] Y. Zhou, H.-X. Li, and S.-L. Xie, "Fast modeling of battery thermal dynamics based on spatio-temporal adaptation," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 337–344, Jan. 2022.
- [6] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," *Nat. Rev. Phys.*, vol. 3, no. 6, pp. 422–440, 2021.
- [7] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *J. Comput. Phys.*, vol. 378, pp. 686–707, Feb. 2019.
- [8] W. Chen, Q. Wang, J. S. Hesthaven, and C. Zhang, "Physics-informed machine learning for reduced-order modeling of nonlinear problems," *J. Comput. Phys.*, vol. 446, Dec. 2021, Art. no. 110666.
- [9] M. Xia, L. Böttcher, and T. Chou, "Spectrally adapted physics-informed neural networks for solving unbounded domain problems," *Mach. Learn., Sci. Technol.*, vol. 4, no. 2, 2023, Art. no. 025024.
- [10] S. Wang, X. Yu, and P. Perdikaris, "When and why PINNs fail to train: A neural tangent kernel perspective," *J. Comput. Phys.*, vol. 449, Jan. 2022, Art. no. 110768.
- [11] H.-X. Li and C. Qi, "Modeling of distributed parameter systems for applications—A synthesized review from time-space separation," *J. Process Control*, vol. 20, no. 8, pp. 891–901, 2010.
- [12] B.-C. Wang, H.-X. Li, and H.-D. Yang, "Spatial correlation-based incremental learning for spatiotemporal modeling of battery thermal process," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 2885–2893, Apr. 2020.
- [13] M. Wang, H.-X. Li, X. Chen, and Y. Chen, "Deep learning-based model reduction for distributed parameter systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 12, pp. 1664–1674, Dec. 2016.
- [14] K.-K. Xu, H.-X. Li, and H.-D. Yang, "Kernel-based random vector functional-link network for fast learning of spatiotemporal dynamic processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 5, pp. 1016–1026, May 2019.
- [15] D. Zhang and Z. Lu, "An efficient, high-order perturbation approach for flow in random porous media via karhunen-loeve and polynomial expansions," *J. Comput. Phys.*, vol. 194, no. 2, pp. 773–794, 2004.
- [16] Z. Wang and H.-X. Li, "Incremental spatiotemporal learning for online modeling of distributed parameter systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 12, pp. 2612–2622, Dec. 2019.
- [17] B. Xu and X. Lu, "A data-driven spatiotemporal model predictive control strategy for nonlinear distributed parameter systems," *Nonlin. Dyn.*, vol. 108, no. 2, pp. 1269–1281, Feb. 2022.
- [18] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics*. Berlin, Germany: Springer, 2007.
- [19] D. A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers*. Dordrecht, The Netherlands: Springer, 2009.
- [20] P. D. Christofides, "Robust control of parabolic PDE systems," *Chem. Eng. Sci.*, vol. 53, no. 16, pp. 2949–2965, 1998.

- [21] P. D. Christofides and J. Baker, "Robust output feedback control of quasi-linear parabolic PDE systems," *Syst. Control Lett.*, vol. 36, no. 5, pp. 307–316, 1999.
- [22] M. Y. Hussaini and T. A. Zang, "Spectral methods in fluid dynamics," *Annu. Rev. Fluid Mech.*, vol. 19, no. 1, pp. 339–367, 1987.
- [23] W. Raö, *Advanced Process Control*. New York, NY, USA: McGraw-Hill, 1981.
- [24] R. Russell and L. F. Shampine, "A collocation method for boundary value problems," *Numerische Mathematik*, vol. 19, pp. 1–28, Feb. 1972.
- [25] C. A. Fletcher, *Computational Galerkin Methods*. Berlin, Germany: Springer, 1984.
- [26] H. Sagan, *Boundary and Eigenvalue Problems in Mathematical Physics*. Chelmsford, MA, USA: Courier Corp., 1989.
- [27] J. P. Boyd, *Chebyshev and Fourier Spectral Methods*. Chelmsford, MA, USA: Courier Corp., 2001.
- [28] Q. Wang, J. S. Hesthaven, and D. Ray, "Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem," *J. Comput. Phys.*, vol. 384, pp. 289–307, May 2019.
- [29] X. Lu, B. Xu, and P. He, "A novel low-order spatiotemporal modeling method for nonlinear distributed parameter systems," *J. Process Control*, vol. 106, pp. 84–93, Oct. 2021.
- [30] Y. Zhou, H.-X. Li, and S.-L. Xie, "Space-decomposition-based spectral modeling for distributed battery thermal dynamics," *IEEE Trans. Transp. Electrif.*, vol. 8, no. 2, pp. 1634–1641, Jun. 2022.
- [31] J.-W. Wang, "Exponentially stabilizing observer-based feedback control of a sampled-data linear parabolic multiple-input–multiple-output PDE," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 9, pp. 5742–5751, Sep. 2019.
- [32] H.-X. Li and C. Qi, *Spatio-Temporal Modeling of Nonlinear Distributed Parameter Systems: A Time/Space Separation Based Approach*. Dordrecht, The Netherlands: Springer, 2011.
- [33] A. Zettl, *Sturm-Liouville Theory*. Providence, RI, USA: Am. Math. Soc., 2005.
- [34] Y. Zhou, H. Deng, H.-X. Li, and S. Xie, "Dual separation-based spatiotemporal modeling methodology for battery thermal process under nonhomogeneous boundary conditions," *IEEE Trans. Transp. Electrif.*, vol. 7, no. 4, pp. 2260–2268, Dec. 2021.
- [35] S. Liu, X. Huang, and P. Protopapas, "Residual-based error bound for physics-informed neural networks," 2023, *arXiv:2306.03786*.
- [36] W. Shen, K. Xu, L. Deng, and S. Zhang, "A Karhunen-Loeve Galerkin online modeling approach for the thermal dynamics of li-ion batteries," *IEEE Access*, vol. 8, pp. 187893–187901, 2020.
- [37] H. Deng, H.-X. Li, and G. Chen, "Spectral-approximation-based intelligent modeling for distributed thermal processes," *IEEE Trans. Control Syst. Technol.*, vol. 13, no. 5, pp. 686–700, Sep. 2005.
- [38] H.-P. Deng, Y.-B. He, B.-C. Wang, and H.-X. Li, "Physics-dominated neural network for spatiotemporal modeling of battery thermal process," *IEEE Trans. Ind. Informat.*, vol. 20, no. 1, pp. 452–460, Jan. 2024.
- [39] Z. Wang, H.-X. Li, and C. Chen, "Reinforcement learning-based optimal sensor placement for spatiotemporal modeling," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2861–2871, Jun. 2020.
- [40] P. Wei and H.-X. Li, "Spatial construction for modeling of unknown distributed parameter systems," *Ind. Eng. Chem. Res.*, vol. 60, no. 42, pp. 15184–15193, 2021.
- [41] Z. Liu and H.-X. Li, "A spatiotemporal estimation method for temperature distribution in lithium-ion batteries," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2300–2307, Nov. 2014.
- [42] Y.-B. He, B.-C. Wang, H.-P. Deng, and H.-X. Li, "Physics-reserved spatiotemporal modeling of battery thermal process: Temperature prediction, parameter identification, and heat generation rate estimation," *J. Energy Storage*, vol. 75, Jan. 2024, Art. no. 109604.



**Bing-Chuan Wang** received the B.E. degree in automation and the M.S. degree in control science and engineering from Central South University, Changsha, China, in 2013 and 2016, respectively, and the Ph.D. degree in systems engineering and engineering management from the City University of Hong Kong, Hong Kong.

Since 2020, he has been with the School of Automation, Central South University. His current research interests include intelligent modeling and evolutionary computation.



**Cong-Ling Dai** was born in ZheJiang, China. She received the B.E. degree in communication engineering from Xiangtan University, Xiangtan, China, in 2022. She is currently pursuing the M.E. degree in artificial intelligence with Central South University, Changsha, China.

Her current research interest includes spatiotemporal modeling of DPS with machine learning.



**Yong Wang** (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is currently a Professor with the School of Automation, Central South University. His current research interests include intelligent learning and optimization and their interdisciplinary applications.



**Han-Xiong Li** (Fellow, IEEE) received the B.E. degree in aerospace engineering from the National University of Defense Technology, Changsha, China, in 1982, the M.E. degree in electrical engineering from Delft University of Technology, Delft, The Netherlands, in 1991, and the Ph.D. degree in electrical engineering from the University of Auckland, Auckland, New Zealand, in 1997.

He is a Chair Professor with the Department of Systems Engineering, City University of Hong Kong, Hong Kong. His current research interests include process modeling and control, distributed parameter systems, and system intelligence.

Prof. Li was awarded the Distinguished Young Scholar (Overseas) by the China National Science Foundation in 2004, a Chang Jiang Professorship by the Ministry of Education, China, in 2006, and a National Professorship in China Thousand Talents Program in 2010. He serves as an Associate Editor for IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS and was an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS from 2002 to 2016 and IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS from 2009 to 2015. He serves as a Distinguished Expert for Hunan Government and China Federation of Returned Overseas Chinese.