Physics of Fluids

RESEARCH ARTICLE | DECEMBER 17 2024

Multiple scale method integrated physics-informed neural networks for reconstructing transient natural convection

Nagahiro Ohashi (大橋永寛) ; Nam Phuong Nguyen (Nguyễn Phương Nam); Leslie K. Hwang ; Beomjin Kwon (권범진)

Physics of Fluids 36, 123625 (2024)
https://doi.org/10.1063/5.0243548
CHORUS

AIP Publishing

Multiple scale method integrated physics-informed neural networks for reconstructing transient natural convection

Cite as: Phys. Fluids **36**, 123625 (2024); doi: 10.1063/5.0243548
Submitted: 11 October 2024 · Accepted: 27 November 2024 ·
Published Online: 17 December 2024

View Online Export Citation CrossMark

Nagahiro Ohashi (□□□寛),[1] Nam Phuong Nguyen (Nguyễn Phương Nam),[1] Leslie K. Hwang,[2] and Beomjin Kwon (권범진)[1,a)]

AFFILIATIONS

[1]School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, Arizona 85287, USA
[2]School of Electrical, Computer and Energy Engineering, Electrical Engineering, Arizona State University, Tempe, Arizona 85287, USA

[a)]Author to whom correspondence should be addressed: kwon@asu.edu

ABSTRACT

This study employs physics-informed neural networks (PINNs) to reconstruct multiple flow fields in a transient natural convection system solely based on instantaneous temperature data at an arbitrary moment. Transient convection problems present reconstruction challenges due to the temporal variability of fields across different flow phases. In general, large reconstruction errors are observed during the incipient phase, while the quasi-steady phase exhibits relatively smaller errors, reduced by a factor of 2–4. We hypothesize that reconstruction errors vary across different flow phases due to the changing solution space of a PINN, inferred from the temporal gradients of the fields. Furthermore, we find that reconstruction errors tend to accumulate in regions where the spatial gradients are smaller than the order of 10^{-6}, likely due to the vanishing gradient phenomenon. In convection phenomena, field variations often manifest across multiple scales in space. However, PINN-based reconstruction tends to preserve larger-scale variations, while smaller-scale variations become less pronounced due to the vanishing gradient problem. To mitigate the errors associated with vanishing gradients, we introduce a multi-scale approach that determines scaling constants for the PINN inputs and reformulates inputs across multiple scales. This approach improves the maximum and mean errors by 72.2% and 6.4%, respectively. Our research provides insight into the behavior of PINNs when applied to transient convection problems with large solution space and field variations across multiple scales.

I. INTRODUCTION

A. Flow field reconstruction

Flow field reconstruction is an important yet challenging task in thermofluid research, that is the process of estimating velocity, temperature or other flow variables throughout a fluid domain based on partial or indirect measurements. In many real-world applications, obtaining complete flow conditions, including velocity, temperature, and pressure distributions at all flow inlets, outlets, and other boundaries, is often impractical or impossible. For instance, laser diagnostics like particle image velocimetry, which measure continuous fluid flow velocity distributions, require light to pass through the fluid system. However, most real-world flow systems are constructed from metals with opaque surfaces, making it challenging to measure complete velocity distributions. Additionally, while point measurement sensors can be inserted into opaque flow systems, they tend to disturb the original fluid flows.

Traditionally, flow field reconstruction has been limited to scenarios such as (1) reconstructing three-dimensional (3D) volumetric fluid flow from two-dimensional (2D) particle flow images[1] and (2) mapping discrete flow parameters to 2D velocity fields.[2–4] For example, computer vision algorithms are often used to perform 3D particle image velocimetry, reconstructing 3D flow motion within the entire fluid volume when 2D particle flow images are provided.[1] Recent advances in machine learning have enabled the reconstruction of 2D velocity fields from discrete flow parameters, but these methods require large amounts of training data. For instance, Yu *et al.* used a multilayer feedforward neural network to reconstruct Mach number contours of nozzle airflows from discrete parameters such as Reynolds number (Re) and pressure ratio.[2] Similarly, Deng *et al.* employed a modified convolutional neural network to reconstruct 2D velocity fields in an air scramjet combustor from discrete pressure data.[3] However, previous methods have limitations: (1) they are challenging to generalize across different flow conditions, (2) may not satisfy physical laws in their reconstructed outputs, and (3) can only reconstruct field variables included in their training datasets. For example, if training data only contain velocity fields, traditional machine learning models cannot reconstruct other field variables like temperature or pressure.

B. Physics-informed neural networks (PINNs) for flow field reconstruction

To address these challenges, PINNs have recently attracted significant interest for their ability to reconstruct flow fields from limited measurement data while adhering to physics laws. PINNs are a class of artificial neural networks wherein the underlying physics of the problem are integrated into the neural network architecture. PINNs address the data limitation issue by inherently learning to approximate the solutions to partial differential equations (PDEs) that characterize the physics of the system. Unlike traditional machine learning methods that rely on large datasets, PINNs leverage the fundamental governing equations of the system to guide the learning process.

There have been initial investigations into the application of PINNs for reconstructing thermofluidic fields. Wang *et al.* reconstructed a 2D natural convection flow around a cylinder with the Richardson number (Ri) of 1 over a duration of 1 s.[5] The quasi-steady nature of the flow allowed for the sufficient recovery of the fields using only 10 frames at different time steps, each frame comprising 200 data points for temperature and velocity fields obtained solely from boundary conditions. In this work, an original PINN architecture with 10 hidden layers, each layer containing 100 neurons, was employed with uniform weighting to all loss functions. In another investigation, Cai *et al.* reconstructed a mixed convection flow within a 2D square domain, with heated bottom, operating at Ri = 1.[7] Only temperature data were used to reconstruct the unknown pressure and velocity fields. Training dataset spanned from initial conditions to steady state, incorporating 100 frames, with each frame containing 9950 temperature data points. They emphasized the importance of establishing an appropriate weight ratio between the data and PDE losses to ensure convergence and prevent overfitting, recommending a weight ratio of 50:1.[7] Finally, in the same work,[7] the demonstration of PINN for field reconstruction extends to experimental natural convection data. In this study, the unsteady 3D temperature field in the flow over an espresso cup was captured using Tomographic Background Oriented Schlieren. Training focused on the transient phase, using 400 frames of temperature data without incorporating other boundary conditions.

C. Limitations

Despite the successful demonstrations of original PINNs for addressing inverse problems (i.e., field reconstructions) related to transient convection phenomena, a critical gap remains in understanding how the solution space of PDEs—all possible solutions under incompletely defined problem conditions—in the time domain influences PINN accuracy. Previous studies have primarily focused on either quasi-steady or unsteady states where temporal variations in the field, or equivalently, the time-derivatives of field variables, are not significant. Therefore, future research efforts may be directed toward adapting PINNs to accommodate varying levels of time dependence, which would require adjustments in the temporal domain of training data.

Additionally, in transient convection phenomena, multiple fields can exhibit variations across several spatial-temporal scales, leading to the vanishing gradient problem in PINNs.[8,9] For instance, spatial gradients of pressure, velocity, and temperature within a flow system can vary significantly due to differences in the diffusion coefficients of momentum and heat. In such multi-scale systems, training the original PINN across all types of field variables presents a challenge. Previous research has not addressed multi-scale structures in transient flow problems, and issues related to field gradients remain largely unexplored.

D. Our scope

This work studies the impact of transience on the accuracy of PINNs in addressing unsteady convection problems. Specifically, the objective is to reconstruct the transient natural convection dynamics within a 2D square domain. Due to the increased complexity of the problem with significantly broadened solution space (i.e., possible field distributions along time domain) compared to previous research, we limit the spatial domain of the demonstration to 2D. The PINN is trained to infer completely unknown pressure (P), x- and y-velocity fields (U, V) from temperature data (T) available within an arbitrary chosen time window without the knowledge of the prior flow field development.

A crucial consideration is the varying solution space of PDEs in the time domain across different temporal phases, which may not be uniformly captured by a PINN optimized under a single setting. For example, the incipient phase, the initial stage of the flow system characterized by large time derivatives, contrasts with the quasi-steady phase, where flow field distributions are sufficiently stabilized and time derivatives approach zero. Addressing different temporal phases within transient processes may require adaptation of PINNs. The key contributions of this work are: (1) implementing PINNs for highly transient processes and investigating error sources in field reconstruction; and (2) adapting PINNs using a multi-scale approach to address the vanishing gradient problem that can arise from the presence of multiple spatial scales within the system.

II. PROBLEM AND METHODOLOGY

A. Problem definition

The problem of interest is the reconstruction of the unknown field variables that are physically coupled: pressure, x- and y-velocity in a canonical transient buoyancy-driven flow within a 2D square cavity as shown in Fig. 1. The fields are reconstructed during an arbitrary chosen time window if temperature data across the domain is provided, along with boundary conditions. Both width (w) and height (h) of the cavity are 0.015 m. The left wall is at $T_1 = 37\,^\circ\mathrm{C}$, and the right wall is at $T_0 = 35\,^\circ\mathrm{C}$. The top and bottom walls are thermally insulated (i.e., $\mathrm{d}T/\mathrm{d}x$ or $\mathrm{d}T/\mathrm{d}y = 0$), and all walls exhibit the no-slip condition. Asymmetric wall temperatures create a density gradient that drives a buoyancy-driven flow circulating the cavity. The flow field within the square cavity is characterized by the Rayleigh number, [$\mathrm{Ra} = g\beta(T_{max} - T_{min})w^3/\nu\alpha$], where g is the gravitational acceleration, β is the thermal expansion coefficient ($\beta = 3.34 \times 10^{-4}\ \mathrm{K}^{-1}$), ν is the kinematic

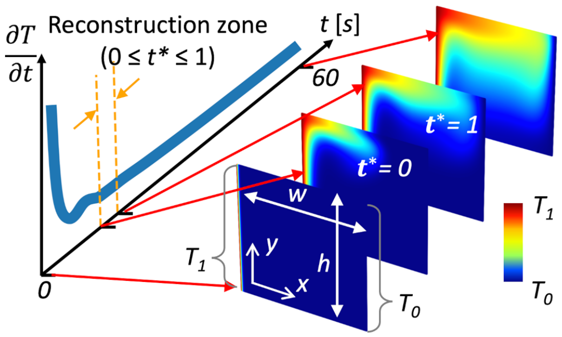

FIG 1. Schematic of a square cavity with a heated left wall, a cooled right wall, and insulated top and bottom walls. During the initial stage, the temperature field exhibits a significant temporal gradient. A reconstruction zone is arbitrary selected within the complete dataset, and a normalized timescale ($0 \le t^* \le 1$) is assigned.

viscosity ($\nu = 7.26 \times 10^{-7}\,\mathrm{m^2/s}$), and α is the thermal diffusivity ($\alpha = 1.49 \times 10^{-7}\ \mathrm{m^2/s}$). The temperature difference ($T_{max} - T_{min}$) is one way of controlling the complexity of flow fields, where T_{max} is the maximum surface temperature, and T_{min} is the minimum surface temperature. In this study, a temperature difference of 2 °C results in a Ra of 2×10^5, which is below the critical Ra of 3×10^5, indicating that the flow remains laminar.

B. Governing equations

The 2-D incompressible Navier–Stokes equation and the energy equation govern buoyancy-driven flow within a square domain. Non-dimensionalization of PDEs is often preferred to make the network adaptable to a wider array of problems by utilizing dimensionless terms to characterize the system. Furthermore, due to the nature of common activation functions used in PINNs (e.g., tanh and swish[10]) which exhibit nonlinearity and well-defined derivatives within the range of -1 to 1, normalizing the input and outputs within this interval is recommended. Consequently, the problem formulation incorporates the following normalized terms, denoted by '*' to signify nondimensional parameters with $x^* = x/w$, $y^* = y/h$, $t^* = t/t_{exp}$, $U^* = U/U_0$, $V^* = V/U_0$, $P^* = P/\rho_0 U_0^2$, $T^* = (T\text{-}T_{min})/(T_{max} - T_{min})$. The dimensional variables are defined as $x \in [0, w]$, $y \in [0, h]$, and $t \in [0, t_{\text{exp}}]$, where t is the temporal coordinate and t_{exp} is the experimental time duration, which defines the window of the reconstruction dataset. The characteristic variables are the reference velocity (U_0), maximum density (ρ_0), and reference temperature difference ($T_1 - T_0$). In order for Ri to be 1, U_0 was chosen to be $U_0 = [g\beta(T_1 - T_0)w]^{0.5}$, leading to $U_0 = 0.009\,92$ m/s. The Boussinesq approximation is used to capture density variation as a function of temperature and yields the following normalized PDEs:

$$\frac{\partial U^*}{\partial x^*} + \gamma \frac{\partial V^*}{\partial y^*} = 0, \tag{1}$$

$$St\frac{\partial U^*}{\partial t^*} + U^*\frac{\partial U^*}{\partial x^*} + \gamma V^*\frac{\partial U^*}{\partial y^*} + \frac{\partial P^*}{\partial x^*} - \frac{1}{Re}\left(\frac{\partial^2 U^*}{\partial x^{*2}} + \gamma^2\frac{\partial^2 U^*}{\partial y^{*2}}\right) = 0, \tag{2}$$

$$St\frac{\partial V^*}{\partial t^*} + U^*\frac{\partial V^*}{\partial x^*} + \gamma V^*\frac{\partial V^*}{\partial y^*} + \gamma\frac{\partial P^*}{\partial y^*} - \frac{1}{Re}\left(\frac{\partial^2 V^*}{\partial x^{*2}} + \gamma^2\frac{\partial^2 V^*}{\partial y^{*2}}\right) + RiT^* = 0, \tag{3}$$

$$St\frac{\partial T^*}{\partial t^*} + U^*\frac{\partial T^*}{\partial x^*} + \gamma V^*\frac{\partial T^*}{\partial y^*} - \frac{1}{Pe}\left(\frac{\partial^2 T^*}{\partial x^{*2}} + \gamma^2\frac{\partial^2 T^*}{\partial y^{*2}}\right) = 0. \tag{4}$$

The dimensionless parameters $\mathrm{St} = w/U_0 t_{exp}$, $\mathrm{Re} = U_0 w/\nu$, $\mathrm{Pe} = \mathrm{RePr}$, and γ represent the Strouhal number, Reynolds number, Peclet number, and geometric aspect ratio of the domain, respectively. According to the problem definition in Sec. II A, these parameters are calculated as Re = 206, Pe = 997, and $\gamma = 1$. We arbitrary set t_{exp} as 3.9 s that covers about 2.5 characteristic time periods ($t_c = w/U_0$), resulting in an St of 0.39.

C. Network architecture

The PINN architecture used for the flow field reconstruction comprises a fully connected neural network with 10 hidden layers, each housing 150 neurons (equivalently, 10 × 150 configuration). The inputs to the PINN are spatial-temporal coordinates (x^*, y^*, t^*), while the outputs are the four field quantities (T^*, U^*, V^*, P^*) computed at these coordinates. The outputs are constrained by the governing PDEs [Eqs. (1)–(4)], given temperature data, and boundary conditions, as detailed in Sec. II D. The select activation function is a locally adaptive sine function.[11,12] The governing equations demand an activation function that is second order differentiable and does not vanish to zero. The sine function has demonstrated efficacy in previous implementations when solving natural convection and mixed convection PDEs.[13,14]

To explore hyperparameter sensitivity, the architecture was varied in both size and activation function. While a larger network offers increased expressiveness, it comes with the trade-off of higher computational cost.[15] Comparative tests were conducted using networks of different size (i.e., 12 hidden layers with 200 neurons per layer), yielding negligible differences, typically less than 1% when compared to networks of size 10 × 150. Smaller network size was not considered, since a previous work[7] reported large errors. Thus, the configuration of 10 × 150 was chosen. Additionally, an alternative activation function, 'tanh', was evaluated, in which no significant improvements were observed, thus the sine activation function was kept. Comparison test results are available in the supplementary material.

D. Loss functions

The loss functions used in PINNs constrain the reconstruction results to specified boundary conditions, data, and governing equations. Thus, the composite loss function (L) includes three components as follows:

$$L = L_{BC} + L_D + L_R, \tag{5}$$

where L_{BC} is the loss computed at points along the domain boundary and L_D is the loss evaluated at points where data are prescribed, denoted as data points. L_R is the PDE loss calculated using both data points and additional points defined within the domain, referred to as collocation points. The losses are expressed as

TABLE I. Seven datasets used for field reconstructions.

	Set 1	Set 2	Set 3	Set 4	Set 5	Set 6	Set 7
Start frame	10	60	100	150	200	300	500
Start time t_0 [s]	1	6	10	15	20	30	50
End frame	49	99	139	189	239	339	539
End time $t_0 + t_{exp}$ [s]	4.9	9.9	13.9	18.9	23.9	33.9	53.9

$$L_{BC} = \frac{1}{N_{BC}}\sum_{i=1}^{N_{BC}} \left|F(x_i, y_i, t_i) - F_i\right|^2, \tag{6}$$

$$L_D = \frac{1}{N_D}\sum_{i=1}^{N_D} \left|T^*(x_i, y_i, t_i) - T_i^*\right|^2, \tag{7}$$

$$L_R = \frac{1}{N_R + N_D}\sum_{i=1}^{N_R+N_D} \sum_{k=1}^{4} \left|e_k(x_i, y_i, t_i)\right|^2, \tag{8}$$

where N_{BC}, N_D, and N_R represent the number of boundary points, data points, and collocation points, respectively. For L_{BC}, the field variables, F, reconstructed at the boundary are compared with the boundary conditions F_i. For L_D, reconstructed temperature values are compared against those obtained from computational fluid dynamics (CFD) simulations, denoted as T_i^*. For L_R, the residuals between the reconstruction results and Eqs. (1)–(4), denoted as e_1 through e_4, are computed. The mean square error is used as the optimization criterion for PINNs.

In the composite loss, each loss term can be assigned a weight to bias the reconstruction toward a specific constraint. Previous studies have favored data over PDEs in PINNs by applying an L_D/L_R weight ratio of 50–100.[7] However, in our experiments, using an L_D/L_R weight ratio from 1 to 100 did not yield substantial improvements. While a large L_D/L_R weight ratio can accelerate convergence by aligning network parameters more quickly with data constraints, it may also compromise the generalization capacity of the PINNs. With a large L_D/L_R weight ratio, the network parameters may deviate from accurately satisfying the PDEs, potentially leading to overfitting. Thus, equal weights were assigned to the loss terms for further analysis.

E. Network optimization

PINN was built using Pytorch libraries and was executed on a single Nvidia A100 GPU. The network optimization utilized floating-point precision with a machine epsilon of 1.19×10^{-7}. The optimization of the loss function employed a coupled strategy,[16] initially utilizing the Adam optimizer with 100 000 epochs and a learning rate of 0.0001 followed by refinement with the L-BFGS optimizer until convergence.[6,17] The network parameters were initialized using the Xavier normal distribution.

To evaluate the reconstruction results and provide temperature data to the PINNs, reference data were generated using a finite volume model (FVM), implemented in ANSYS Fluent. Independent mesh and time step convergence analyses were performed to ensure accuracy within 1% of a finer step size. The transient natural convection was simulated over a 60-second duration with a time step of 0.001 seconds, while data were extracted at intervals of 0.1 seconds (denoted as

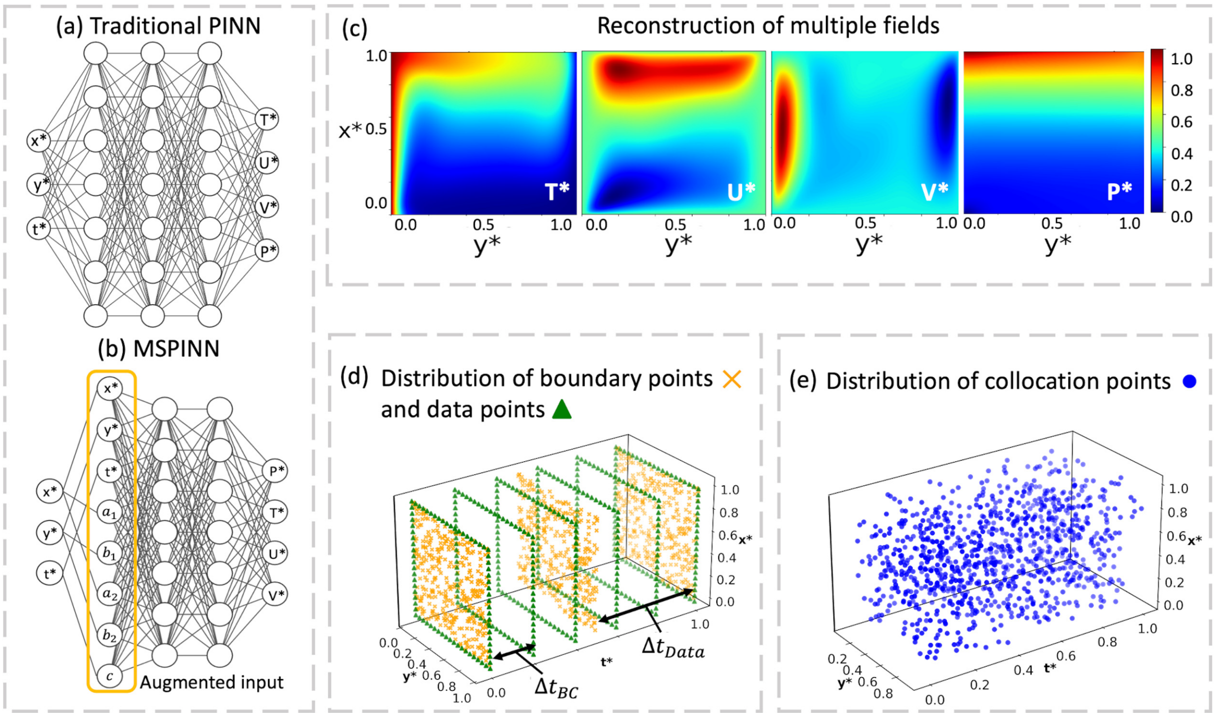

FIG. 2. Overview of field reconstruction process. Two PINN architectures are tested: (a) a traditional PINN and (b) a PINN modified based on multiple scale method (MSPINN). (c) PINNs reconstruct unknown fields (U^*, V^*, P^*) based on the provided temperature field (T^*). Neural network losses are evaluated at (d) boundary and data points, as well as at (e) collocation points.

Δt_{Data}), resulting in a total dataset of 600 frames. From the total dataset, seven different subsets were sampled at distinct temporal phases of the flow, each spanning 3.9 seconds (equivalent to t_{exp} and 40 frames). Table I lists the start and end times of the subsets. Although the start times t_0 differ among subsets, when the subset data are provided to a PINN, t_0 is normalized to $t^* = 0$, as illustrated in Fig. 1. The normalization of t_0 ensures that no information regarding the prior flow conditions or the absolute time (t) is provided to the PINN. This approach allows the PINN to operate independently of the absolute time of dataset, enabling flow field reconstruction at any point in time without relying on prior states.

Figure 2 illustrates a schematic overview of the field reconstruction process, displaying the points where the losses are computed. The number of boundary points in a single edge along the x, y, and t axes, denoted as N_x, N_y, and N_t, is set to 100 each, resulting in a total of 40 000 boundary points, estimated as $N_b = [2(N_x + N_y)N_t]$. Consequently, the time step between boundary points along the t axis ($\Delta t_{BC} = t_{exp}/N_t$) is 0.039 seconds. Collocation points, ranging from $N_r = 162\,880$ to $262\,880$, are randomly sampled throughout the domain using the Latin-Hypercube (LHS) method.[18] Data points are also randomly sampled using LHS from the nodal information of the FVM data, representing 2.5% to 10% of the dataset (equivalently, between 893 and 3572 points per frame). The coordinates of all points are provided to the PINN, where the loss is computed, and the network is updated through iterations.

F. Multiple scale method

Equations (1)–(4) are rescaled using the multiple scale method in perturbation theory, a technique that analyzes variable scales to identify those that contribute most significantly to the overall solution. The PINN based on multiple scale method (MSPINN) closely aligns with previous works.[19–22] Huang *et al.* used *matched asymptotic expansions* of multiple networks to create a uniformly valid solution of governing equations with different scales.[23] Unlike Huang's approach, which solves outer and inner solutions separately using *matched asymptotic expansions*, we introduce augmented inputs via scaling factors directly integrated into the input layer, allowing the solution to be calculated with a single network.

For Eqs. (1)–(4), we define the lengths ($\delta_{\mathrm{l,i}}$) and time (δ_{t}) scaling factors and corresponding rescaled variables, where $a_i = x^*/\delta_{\mathrm{l,i}}$, $b_i = y^*/\delta_{\mathrm{l,i}}$, and $c = t^*/\delta_{\mathrm{t}}$, where the subscript i denotes the i^{th} scale. Thus, in MSPINN, the original PINN inputs, $\boldsymbol{x} = [x^*, y^*, t^*]$, are expanded into a modified input structure, $\bar{x} = [x^*, y^*, t^*, a_1, b_1, a_2, b_2, c]$. The modified input occupies a distinct layer following the initial network inputs $\boldsymbol{x}$, as shown in Fig. 2(b).

To determine the scaling factors, rescaled forms of Eqs. (1)–(4) are needed which are expressed with rescaled derivatives of field variables (U^*, V^*, T^*, P^*). The derivatives of U^* with respect to scaling factors can be obtained using the chain rule as follows:

$$\frac{\partial U^*}{\partial x^*} = \frac{\partial U^*}{\partial a_i}\frac{\partial a_i}{\partial x^*} + \frac{\partial U^*}{\partial x^*}\frac{\partial x^*}{\partial x^*} = \frac{U^*_{a,i}}{\delta_{l,i}} + U^*_x, \qquad (9)$$

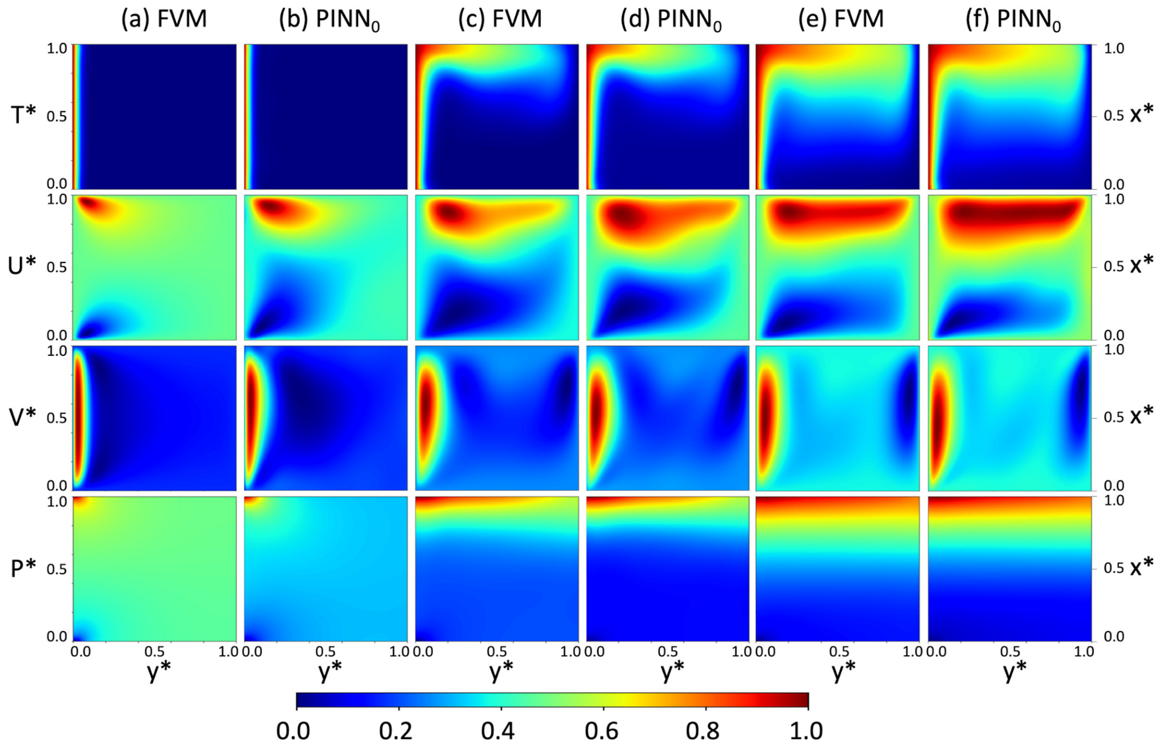

FIG. 3. Comparison of FVM (a, c, e) and PINN$_0$ predictions (b, d, f), evaluated at three different flow phases: (a, b) Incipient phase at $t = 1$s (start frame of set 1), (c, d) development phase at $t = 20$s (start frame of set 5), and (e, f) quasi-steady state at $t = 50$s (start frame of set 7).

$$\frac{\partial U^*}{\partial y^*} = \frac{\partial U^*}{\partial b_i}\frac{\partial b_i}{\partial y^*} + \frac{\partial U^*}{\partial y^*}\frac{\partial y^*}{\partial y^*} = \frac{U^*_{b,i}}{\delta_{l,i}} + U^*_y, \tag{10}$$

$$\frac{\partial U^*}{\partial t^*} = \frac{\partial U^*}{\partial c}\frac{\partial c}{\partial t^*} + \frac{\partial U^*}{\partial t^*}\frac{\partial t^*}{\partial t^*} = \frac{U^*_c}{\delta_t} + U^*_t. \tag{11}$$

Similarly, rescaled derivatives of T^*, V^*, and P^* can be obtained.

Using the rescaled derivatives, Eqs. (1)–(4) can be rewritten in their rescaled forms. The rescaled forms of Eqs. (2) and (4) are presented as follows:

$$\begin{aligned} &St\left(U^*_c\frac{1}{\delta_{t,1}} + U^*_t\right) + U^*\left(U^*_{a_1}\frac{1}{\delta_{l,1}} + U^*_x\right) + \gamma V^*\left(U^*_{b_1}\frac{1}{\delta_{l,1}} + U^*_y\right) \\ &\quad + \left(P^*_{a_1}\frac{1}{\delta_{l,1}} + P^*_x\right) - \frac{1}{Re}\left[\left(U^*_{a_1a_1}\frac{1}{\delta_{l,1}{}^2} + \frac{2}{\delta_{l,1}}U^*_{a_1x} + U^*_{xx}\right)\right. \\ &\quad \left. + \gamma^2\left(U^*_{b_1b_1}\frac{1}{\delta_{l,1}{}^2} + \frac{2}{\delta_{l,1}}U^*_{b_1y} + U^*_{yy}\right)\right] = 0, \end{aligned} \tag{12}$$

$$\begin{aligned} &St\left(T^*_c\frac{1}{\delta_{t,2}} + T^*_t\right) + U^*\left(T^*_{a_2}\frac{1}{\delta_{l,2}} + T^*_x\right) \\ &\quad + \gamma V^*\left(T^*_{b_2}\frac{1}{\delta_{l,2}} + T^*_y\right) - \frac{1}{Pe}\left[\left(T^*_{a_2a_2}\frac{1}{\delta_{l,2}{}^2} + \frac{2}{\delta_{l,2}}T^*_{a_2x} + T^*_{xx}\right)\right. \\ &\quad \left. + \gamma^2\left(T^*_{b_2b_2}\frac{1}{\delta_{l,2}{}^2} + \frac{2}{\delta_{l,2}}T^*_{b_2y} + T^*_{yy}\right)\right] = 0. \end{aligned} \tag{13}$$

The magnitudes of each term in the rescaled equations should be compared to identify the dominant terms. In the rescaled equations, most terms scale with unity, except for the terms multiplied by dimensionless numbers such as St, 1/Re, and 1/Pe. In rescaled Eq. (1), all terms are of the same scale, thus do not require further scale analysis. The scale analysis for rescaled Eqs. (2) and (3) is identical due to their structural similarity and the fact that Ri = 1. Thus, we use only Eqs. (12) and (13) for the scale analysis to determine the scaling factors.

In Eq. (12), the order of magnitudes for each term is represented by the coefficients of T^*, U^*, V^*, P^* terms, such as St/$\delta_{t,1}$, St, 1/$\delta_{l,1}$, 1, 1/$\delta_{l,1}{}^2$Re, 1/$\delta_{l,1}$Re, and 1/Re. For simplicity in analysis, we assume that the velocities U^* and V^* are of the same order of magnitude as 1/$\delta_{l,1}$. Consequently, the terms 1/$\delta_{l,1}{}^2$Re and St/$\delta_{t,1}$ are considered dominant. By balancing the dominant terms with the constant term (i.e., 1), we obtain $\delta_{l,1} = \mathrm{Re}^{-0.5}$ and $\delta_{t,1} = \mathrm{St}$. Similarly, the scale analysis of Eq. (13) leads to a dominant balance, yielding $\delta_{l,2} = \mathrm{Pe}^{-0.5}$ and $\delta_{t,2} = \mathrm{St}$. Holmes provides in-depth explanations on dominant balance and multiple-scale analysis.[24] Note that these scaling factors should be updated for different problems, as each problem will have its own unique scaling relations. However, problems governed by the same PDEs may share the same scaling constants, thus eliminating the need for repeated derivation.

In MSPINN, only the input structure—represented by the first layer of the neural network—is modified, while the remaining layers, forming a fully connected network, remain identical to those in the original PINNs. The risk of overfitting is minimal, as the additional trainable parameters account for less than 0.7% of the total parameters. Furthermore, the computational cost of MSPINN is nearly identical to that of traditional PINNs, since the optimization time is primarily influenced by the number of trainable parameters, assuming other factors (e.g., loss function, training points) remain constant. Both PINNs and MSPINN are expected to exhibit similar performance impacts when applied to complex geometric domains, as MSPINN focuses on addressing the multiscale aspects within the PDEs, rather than the computational challenges associated with the domain geometry.

III. RESULTS AND DISCUSSION

A. Optimization of traditional PINNs

To assess the efficacy of traditional PINNs, baseline PINNs without rescaled input variables (PINN$_0$) were first optimized using the

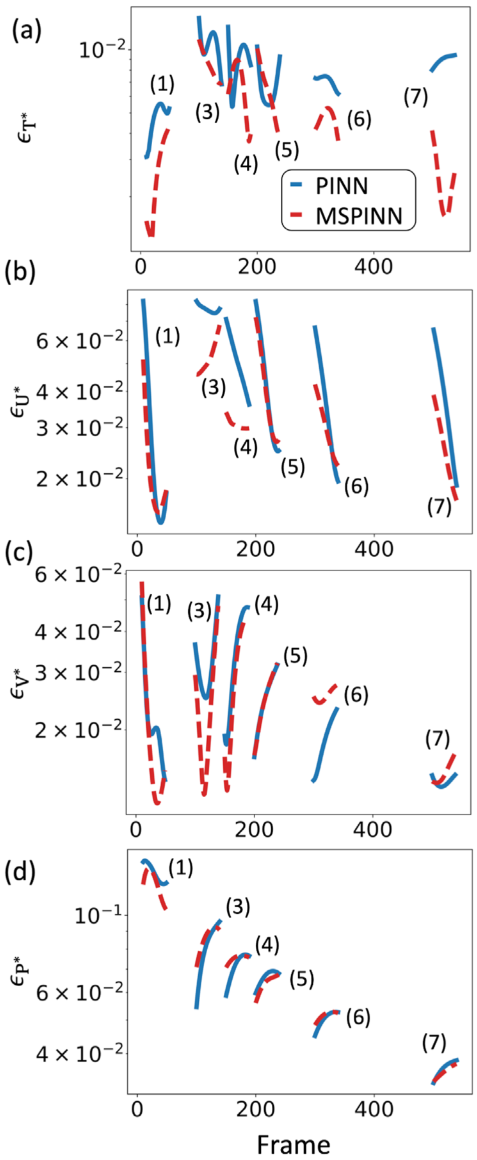

FIG. 4. ε of PINN and MSPINN for the six different temporal sets in (a) T^*, (b) U^*, (c) V^* and, (d) P^* fields. Numbers in each plot represent the dataset number.

most resource-intensive settings. These settings utilized 10% of the available FVM data points (i.e., $N_d = 142\,880$ points), with $N_b = 40\,000$, and $N_r = 162\,880$. The network optimization time was influenced by factors such as the number of epochs and the volume of collocation points. On average, completing 100 000 epochs while processing N_r of 162 880 took about 6 h and used around 40 GB of memory.

Figure 3 compares the reconstruction results of PINN_0 at three different flow phases (i.e., sets 1, 5, 7) with the FVM reference data. The reconstructed fields show good qualitative agreement with the reference data. Initially, the warm left wall leads to an upward buoyancy-driven flow, represented by a large V^*, resulting in pressurization at the top left corner characterized by an elevated P^*. During the development phase, the pressurized, high-temperature fluid at the top left corner moves along the top wall toward the cold right wall, inducing a circulating flow pattern. Along the cold right wall, the fluid descends, facilitating the formation of stable convection cells within the cavity. For a visualization purpose, FVM maps were linearly interpolated to the same grid resolution of PINN_0 (i.e., 500 × 500).

Next, to identify an efficient network optimization setting, the traditional PINN was optimized using two datasets—set 2 or set 3—under various configurations. These datasets represent the most transient flow regimes, characterized by the largest temperature gradients over time. The most transient regimes are typically recognized as more challenging for optimizing PINN. In this experiment, the mean absolute error, ε, was used to quantitatively evaluate the performance of the PINN given as $\varepsilon = |F\text{-}\widehat{F}|_1/N$, where F, $\widehat{F}$, and N are the PINN prediction, FVM ground truth, and number of points, respectively. Errors were calculated for each time step across all grid nodes. Due to the stochastic nature of network optimization, errors varied in each optimization. Thus, the PINN underwent five optimization iterations.

Experiments revealed that increasing N_r from 162 880 to 262 880 did not lead to significant improvements in errors, thus N_r of 162 880 was chosen. The variation in N_d from 2.5% to 10% of total data resulted in minor but noticeable differences. It was found that at least 5% of data should be used for the network optimization, and 10% was chosen to ensure the PINN accurately captures the detailed features of the flow. Tuning the learning rate showed that 1×10^{-4} provided the optimal balance between error reduction and optimization time. For example, about 3% decrease in error was traded for a doubled optimization time. Finally, it was determined that 100 000 epochs were optimal. Longer epochs are usually required for tackling more complex problems. The hyperparameters derived in this section are used for subsequent reconstructions. For detailed descriptions of these experiments, refer to the supplementary material.

B. Reconstructions by PINN and MSPINN

The transient evolution of the flow can be characterized into three distinct phases: the incipient phase (R_{in}), development phase (R_{de}), and quasi steady-state phase (R_{ss}), each exhibiting unique features. In the incipient phase [Figs. 3(a) and 3(b)], the temperature and velocity fields display relatively uniform distributions, with a significant solution space of the next phase and large field gradients with respect to time. In this context, the "solution space" refers to the range of possible field distributions that the PINN can converge to. Next, the development phase [Figs. 3(c) and 3(d)] covers the majority of transient behaviors, with the fields having the largest time derivatives. Finally, the quasi steady-state phase [Figs. 3(e) and 3(f)] is the transition from development to steady-state, in which the temporal effects become less apparent. The differences among three phases can impact the accuracy of PINN reconstruction, particularly considering the varying solution space.

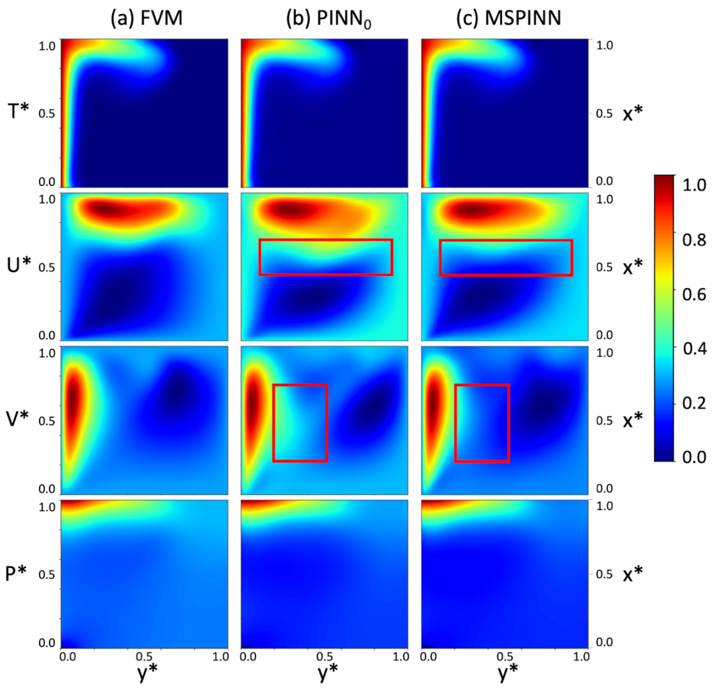

FIG. 5. Comparison between (a) FVM, (b) PINN, and (c) MSPINN at frame 140 of dataset. The improvements in reconstructions are most prominent at the upper surface of boundary layers and are indicated by the boxed regions.

To understand the impact of flow transience on PINN reconstruction, both the traditional PINN and MSPINN were used to reconstruct the fields at six flow phases (i.e., sets 1, 3, 4, 5, 6, and 7). Each reconstruction underwent three to five iterations of network optimization, with the subsequent analysis based on the average results of these iterations. Figure 4 shows ε of both original PINNs and MSPINNs, offering several interesting observations. First, errors are relatively large at the incipient phase (R_{in}) and diminish at the quasi-steady state (R_{ss}). Second, within each flow phase, there is wide variation in error. For a statistical analysis of ε, we estimate the mean (denoted as $\bar{\epsilon}$) and range (denoted as $\Delta\varepsilon$), defined as the difference between maximum and minimum values of ε. Despite a few outliers, both $\bar{\epsilon}$ and $\Delta\varepsilon$ decrease over time, indicating that PINNs are more accurate during R_{ss} than during R_{in}.

The comparison between the original PINNs and MSPINNs shows that improvements can be made by rescaling, in which the impact is more prominent in earlier velocity reconstructions. The maximum improvement in ε was around 3.78×10^{-2} in U^* (roughly 72.2%), and on average the improvement in ε across all field variables was around 1.96×10^{-3} (around 6.4%). This demonstrates that MSPINN effectively accounts for the various scales present in the natural convection process. Additionally, Fig. 5 compares the fields reconstructed by PINN and MSPINN during a development phase (i.e., frame 140). While the PINN captures large-scale temperature patterns, the MSPINN provides more accurate reconstructions of sharp temperature gradients in localized regions.

C. Error sources

We introduce two metrics to understand the cause of relatively large inaccuracies observed during the incipient phase. The first metric, namely, the temporal gradient ($\nabla_t \hat{F} = |\hat{F}_{t+1}$ - $\hat{F}_t|/\Delta t$), is used to investigate the cause of relatively large inaccuracies in the incipient phase, where $\hat{F}$ is the field variable obtained from FVM groundtruth and Δt is the time step. We hypothesize that the size of the solution space influences the network performance. This hypothesis stems from the observation that PINNs tend to exhibit higher errors in systems with a larger solution space, especially evident in ill-posed reconstruction problems. A greater solution space implies more possible independent ways in which the field variables can vary over time. Therefore, a higher temporal gradient (denoted as $\nabla_t \hat{F}$) is expected in systems with greater solution space. Accordingly, we indirectly characterize the size of solution space in the system by using $\nabla_t \hat{F}$.

Different solution spaces could exist under various timescales. For example, the incipient phase will include large temporal derivatives (equivalently larger solution space), while the quasi-steady phase will have near zero temporal derivatives (equivalently smaller solution space). These phases, each with significantly different solution space, may necessitate PINNs optimized under different settings.

Figure 6 presents $\nabla_t \hat{F}$, where $\nabla_t \hat{F}$ is an order of magnitude greater during the incipient phase than during the quasi-steady phase. Thus, the incipient phase is shown to possess a larger solution space. It is also clear that the relative magnitude of $\nabla_t \hat{F}$ loosely correlates with the magnitude of $\Delta\varepsilon$ and $\bar{\epsilon}$, in which a larger $\nabla_t \hat{F}$ roughly corresponds to a higher $\Delta\varepsilon$ and $\bar{\epsilon}$.

The second metric is the absolute spatial gradients ($\nabla_s \hat{F} = |\hat{F}_{s+1} - \hat{F}_s|/\Delta s$) of each field, where Δs is the spatial step along either the x (Δs_x) or y (Δs_y) axis that in general can be different. Figure 7 depicts $|\nabla_s \hat{F}|$ on a logarithm base 10 scale of original PINNs. Three frames were considered, i.e., 10th, 200th, and 500th frames, representing different phases of field development. To compute the gradients, FVM data were interpolated on a 500×500 grid with a Δs of 3×10^{-5} m. Figure 7 shows that the $|\nabla_s \hat{F}|$ spans over four orders of magnitude, from 10^{-6} to 10^{-2}. When $|\nabla_s \hat{F}|$ becomes very small, it results in small gradient terms in the loss function. This, in return, hinders effective parameter updates during backpropagation through deep neural networks, a phenomenon referred to as the vanishing gradient issue.[8,9] While the neural network updates its parameters, repeated

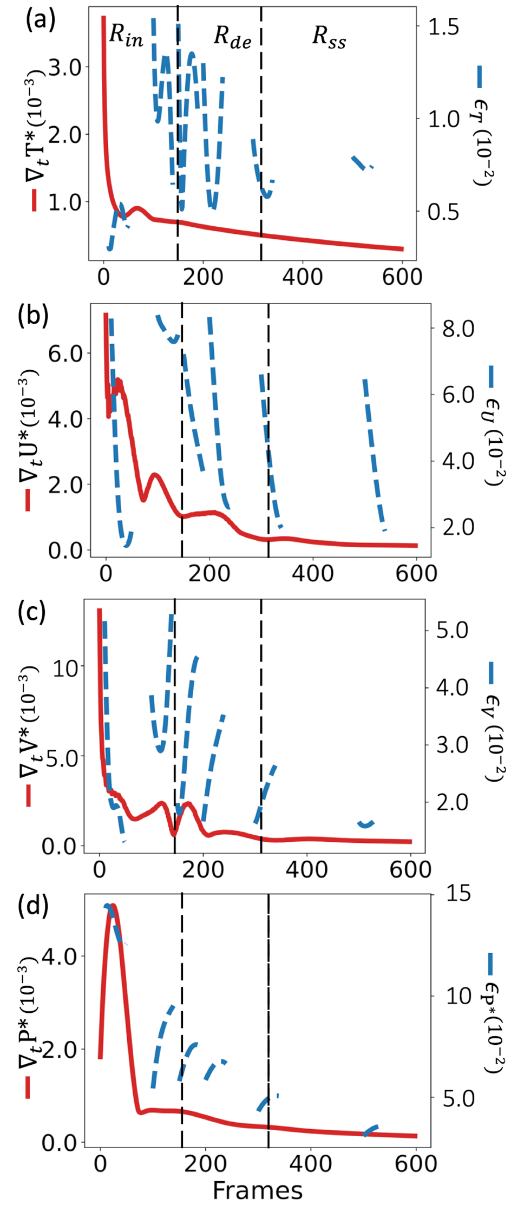

FIG. 6. $\nabla_t \hat{F}$for each field (solid line) overlaid with ε of PINN (dashed line) in (a) T^*, (b) U^*, (c) V^* and, (d) P^* fields.

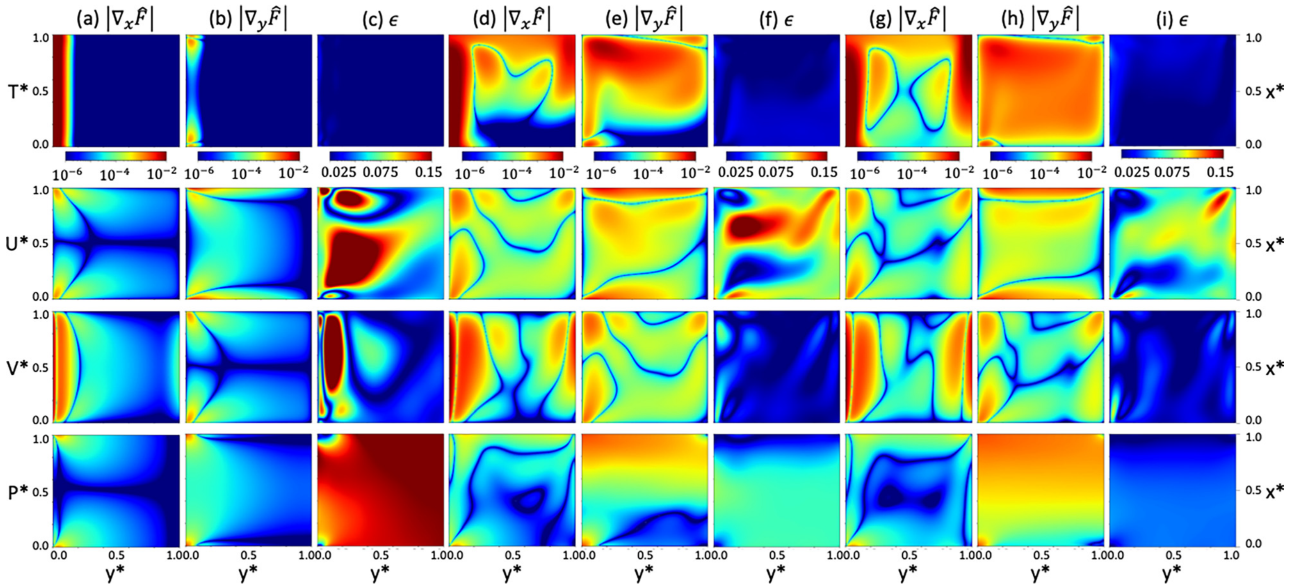

FIG. 7. $|\nabla_s\widehat{F}|$ for x- and y-axis, and ε of field variables obtained at (a–c) 10th frame, (d–f) 200th frame, and (g–i) 500th frame. The fields are plotted on a clipped log scale between 10^{-2} and 10^{-6} to enhance contrast. Error fields of original PINNs are shown with color scales clipped between 5×10^{-3} and 1.5×10^{-1} to show important contrast.

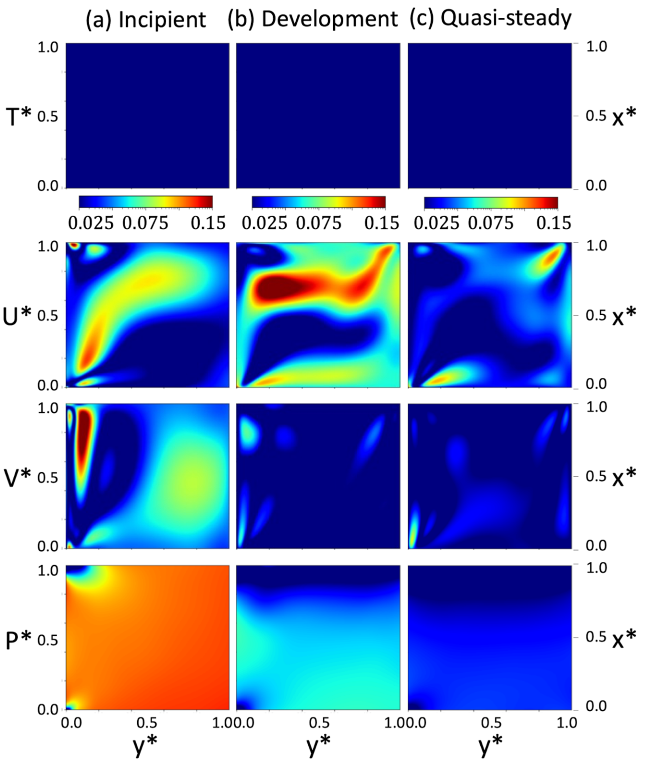

FIG. 8. Error field of MSPINNs obtained at (a) incipient (10th frame), (b) development (200th frame), and (c) quasi-steady phase (500th frame) with color scales clipped between 5×10^{-3} and 1.5×10^{-1} to show important contrast. Lower errors near small gradient regions are observed.

multiplication of small gradient values can result in extremely small (i.e., vanishing) gradient terms, causing the update process to stagnate. Figures 7(c), 7(f), and 7(i) also show the error distribution of original PINNs in each field at 10th, 200th, and 500th frames. The pressure errors exhibit relatively uniform distribution, while exhibiting large errors appear near the cold wall at the incipient phase where the $|\nabla_x P|$ is the smallest. For the temperature field, errors are less than 2×10^{-2}, since the PINN was constrained by the temperature data. For the velocity fields, errors are highest during the incipient phase and around the areas with near-zero gradients. The alignment of regions characterized by extremely small $|\nabla_s\widehat{F}|$ and large ε supports the presence of a vanishing gradient issue, particularly noticeable during the incipient phase.

Additionally, Fig. 7 reveals multiple length scales within the field. First, the spatial scale of the gradient field variation is relatively small. Gradient field values exhibit an order of magnitude variation even across Δx^* or Δy^* of 0.1. However, the spatial scale of original fields shown in Fig. 3 is relatively large. An order of magnitude variation of original field variables occurs across the length of the entire domain. If there are multiple length scales between the original and gradient fields, it necessitates rescaling the PDEs in Eqs. (1)–(4). Otherwise, PINNs tend to favor solutions preserving the larger-scale variations, while the small-scale variations become recessive due to the vanishing gradient issue.

Figure 8 shows the error maps of MSPINN at three frames (i.e., 10th, 200th, and 500th frames) corresponding to those in Fig. 7. Compared to Fig. 7, the errors of MSPINN are apparently reduced, although some inaccuracies still appear in regions with small gradients. The most improvement occurs during the incipient phase, which aligns with the trends observed in Fig. 4.

In addition to ε, we compare the Reynolds number (Re) to evaluate the reconstruction results of MSPINN. Since MSPINN reconstruct

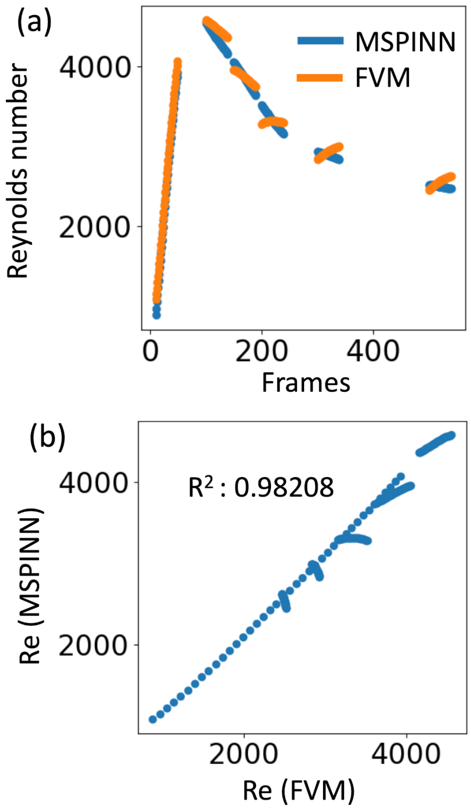

FIG. 9. (a) Reynolds number based on the using mean of top 5% U^*, V^* for FVM and MSPINN for the six tested datasets. (b) Representative Re for FVM and MSPINN plotted against each with an R^2 of 0.982.

unseen velocity fields, the comparison of Re yields insight into the ability of MSPINN to reconstruct accurate velocity magnitudes. Here, we use a characteristic velocity different from U_0 in Sec. II B. Specifically, the characteristic velocity is defined as the average of the top 5% of velocity magnitudes at points within the cavity. Figure 9 compares the Re values between MSPINN and FVM, showing good alignment with a coefficient of determination (R^2) of 0.982. Interestingly, the incipient phase (i.e., frames < 50) demonstrates higher accuracy than subsequent flow phases, consistent with the trends observed in Figs. 4 and 7. Mean values for the top 5% of U^* and V^* are available in the supplementary material.

IV. CONCLUSIONS

This paper explored the ability of PINNs to reconstruct pressure and velocity fields from an instantaneous temperature data for a transient natural convection process. A fully connected network with a structure of 10 layers and 150 neurons per layer was employed to construct the PINN, and its performance was enhanced through hyperparameter tuning. Study into the effects of transience on PINNs led to a few key results. PINN exhibited relatively higher accuracies during the quasi-steady state, while less accuracies during the incipient state. To explain the inaccuracies during the incipient phase, we examined two metrics: spatial and temporal gradients of the field variables ($\nabla_s\widehat{F}$ and $\nabla_t\widehat{F}$). The temporal gradients of field variables serve as indicators of the solution space, complexity and variability within a field. As expected, the magnitude of $\nabla_t\widehat{F}$ was an order of magnitude greater during the incipient phase than during the quasi-steady phase, indicating the presence of significantly larger solution space during the incipient phase. The spatial gradients of field variables revealed that $\nabla_s\widehat{F}$ spanned more than four orders of magnitude in the natural convection studied, even reaching down to 10^{-6}. The correlation between regions exhibiting extremely small $|\nabla_s\widehat{F}|$ and large ε indicated the presence of a vanishing gradient problem, particularly evident during the incipient phase.

Furthermore, we noticed various length scales present within the field. To reconstruct the fields at different scales, we employed multiple scale method, in which the inputs were recast into three distinct scales. The MSPINN resulted in lower ε with maximum and average improvements of around 72.2% and 6.4%, respectively. This result demonstrates the potential efficacy of the multiple scale method in solving thermofluidic reconstruction problems where solutions span across different scales. Next, the top 5% of U^* and V^* of MSPINN yielded Reynolds numbers with R^2 of 0.982 when compared to FVM data, indicating good agreement in velocity magnitudes that are representative of the flow state. Though scaling is currently performed manually, research into perturbation theory could automate this process. Moreover, generalization of MSPINN to other systems and experimental data will further solidify the applicability of this method, including the turbulent systems which necessitate solving multiple scale equations.

SUPPLEMENTARY MATERIAL

See the supplementary material for neural network optimization, detailing the effects of varying collocation points, data points, learning rates, epochs, network sizes, and activation functions.

ACKNOWLEDGMENTS

This work was supported by two National Science Foundation grants under Grant No. 2053413 and 2337973. Additionally, this study was partially supported by Arizona State University startup funds.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Nagahiro Ohashi: Investigation (lead); Writing – original draft (lead). **Nam Phuong Nguyen:** Formal analysis (equal). **Leslie K. Hwang:** Conceptualization (supporting); Resources (lead); Writing – review & editing (equal). **Beomjin Kwon:** Conceptualization (equal); Funding acquisition (lead); Supervision (lead); Writing – review & editing (lead).

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon reasonable request.

REFERENCES

[1] Y. Ding, Z. Li, Z. Chen, Y. Ji, J. Yu, and J. Ye, "Full-volume 3D fluid flow reconstruction with light field PIV," IEEE Trans. Pattern Anal. Mach. Intell. **45**(7), 8405–8418 (2023).

[2] J. Yu and J. S. Hesthaven, "Flowfield reconstruction method using artificial neural network," AIAA J. **57**(2), 482–498 (2019).

[3] X. Deng, M. Guo, H. Chen, Y. Tian, J. Le, and H. Zhang, "Dual-path flow field reconstruction for a scramjet combustor based on deep learning," Phys. Fluids **34**(9), 095118 (2022).

[4] M. Kang, N. Phuong Nguyen, and B. Kwon, "Deep learning model for rapid temperature map prediction in transient convection process using conditional generative adversarial networks," Therm. Sci. Eng. Prog. **49**, 102477 (2024).

[5] T. Wang, Z. Huang, Z. Sun, and G. Xi, "Reconstruction of natural convection within an enclosure using deep neural network," Int. J. Heat Mass Transfer **164**, 120626 (2021).

[6] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," Math. Program. **45**(3), 503–528 (1989).

[7] Cai, S., Wang, Z., Fuest, F., Jeon, Y.-J., Gray, C., and Karniadakis, G. E., "Flow over an espresso cup: Inferring 3D velocity and pressure fields from tomographic background oriented schlieren videos via physics-informed neural networks," J. Fluid Mech. **915**, A102 (2021).

[8] D. Lucor, A. Agrawal, and A. Sergent, "Physics-aware deep neural networks for surrogate modeling of turbulent natural convection," arXiv:2103.03565 (2021).

[9] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," Int. J. Uncertainty Fuzziness Knowledge-Based Syst. **6**(2), 107–116 (1998).

[10] S. Cuomo, V. S. di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, "Scientific machine learning through physics-informed neural networks: Where we are and what's next," J. Sci. Comput. **92**(3), 88 (2022).

[11] S. Cai, Z. Wang, S. Wang, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks for heat transfer problems," J. Heat Transfer **143**(6), 060801 (2021).

[12] A. D. Jagtap, K. Kawaguchi, and G. Em Karniadakis, "Locally adaptive activation functions with slope recovery for deep and physics-informed neural networks," Proc. R. Soc. A: Math. Phys. Eng. Sci. **476**(2239), 20200334 (2020).

[13] M. Raissi, Z. Wang, M. S. Triantafyllou, and G. E. Karniadakis, "Deep learning of vortex induced vibrations," J. Fluid Mech. **861**, 119–137 (2019).

[14] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations," Science **367**(6481), 1026–1030 (2020).

[15] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, "On the expressive power of deep neural networks," in International Conference On Machine Learning (2017).

[16] H. Eivazi, M. Tahani, P. Schlatter, and R. Vinuesa, "Physics-informed neural networks for solving Reynolds-averaged Navier–Stokes equations," Phys. Fluids **34**(7), 075117 (2022).

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv:1412.6980 (2017).

[18] "Randomized Designs—pyDOE 0.3.6 Documentation."

[19] J. L. Callaham, J. V. Koch, B. W. Brunton, J. N. Kutz, and S. L. Brunton, "Learning dominant physical processes with data-driven balance models," Nat. Commun. **12**(1), 1016 (2021).

[20] Y. Weng and D. Zhou, "Multiscale physics-informed neural networks for stiff chemical kinetics," J. Phys. Chem. A **126**(45), 8534–8543 (2022).

[21] S. Wang, H. Wang, and P. Perdikaris, "On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks," Comput. Methods Appl. Mech. Eng. **384**, 113938 (2021).

[22] L. Zhang and G. He, "Multi-scale-matching neural networks for thin plate bending problem," Theor. Appl. Mech. Lett. **14**(1), 100494 (2024).

[23] J. Huang, R. Qiu, J. Wang, and Y. Wang, "Multi-scale physics-informed neural networks for solving high Reynolds number boundary layer flows based on matched asymptotic expansions," Theor. Appl. Mech. Lett. **14**(2), 100496 (2024).

[24] M. H. Holmes, *Introduction to Perturbation Methods* (Springer Science & Business Media, 2012).