

PhySR: Physics-informed deep super-resolution for spatiotemporal data

Pu Ren^a, Chengping Rao^b, Yang Liu^c, Zihan Ma^d, Qi Wang^a, Jian-Xun Wang^e, Hao Sun^{f,g,*}

^a Department of Civil and Environmental Engineering, Northeastern University, Boston, MA 02115, USA

^b Department of Mechanical and Industrial Engineering, Northeastern University, Boston, MA 02115, USA

^c School of Engineering Science, University of Chinese Academy of Sciences, Beijing, 101408, China

^d West China School of Public Health, Sichuan University, Chengdu, Sichuan 610041, China

^e Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, USA

^f Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China

^g Beijing Key Laboratory of Big Data Management and Analysis Methods, Beijing, 100872, China

ARTICLE INFO

Article history:

Received 10 August 2022

Received in revised form 18 May 2023

Accepted 13 August 2023

Available online 21 August 2023

Dataset link: <https://github.com/paulpuren/PhySR>

Keywords:

Partial differential equations

Scientific data

Super-resolution

Physics-informed learning

Hard-encoding scheme

ABSTRACT

High-fidelity simulation of complex physical systems is exorbitantly expensive and inaccessible across spatiotemporal scales. Recently, there has been an increasing interest in leveraging deep learning to augment scientific data based on the coarse-grained simulations, which is of cheap computational expense and retains satisfactory solution accuracy. However, the major existing work focuses on data-driven approaches which rely on rich training datasets and lack sufficient physical constraints. To this end, we propose a novel and efficient spatiotemporal super-resolution framework via physics-informed learning (i.e., PhySR), inspired by the independence between temporal and spatial derivatives in partial differential equations (PDEs). The general principle is to leverage the temporal interpolation for flow estimation, and then introduce convolutional-recurrent neural networks for learning temporal refinement. Furthermore, we employ the stacked residual blocks with wide activation and sub-pixel layers with pixelshuffle for spatial reconstruction, where feature extraction is conducted in a low-resolution latent space. Moreover, we consider hard imposition of boundary conditions in the network to improve reconstruction accuracy. Results demonstrate the superior effectiveness and efficiency of the proposed method, which outperforms existing baseline algorithms, based on extensive numerical experiments.

© 2023 Elsevier Inc. All rights reserved.

1. Introduction

Modeling high-resolution (HR) response of nonlinear dynamical systems (i.e., at sufficiently small spatiotemporal scales) has emerged as one of the dominant research pursuits in the field of scientific computing, thanks to its capability of providing insights into physical quantities of interest and reliable assessments for operational planning. For instance, numerical simulation of turbulence at fine grids allows for unveiling the richness of its varieties and portraying the delicately interac-

* Corresponding author at: Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, 100872, China.
E-mail addresses: ren.pu@northeastern.edu (P. Ren), haosun@ruc.edu.cn (H. Sun).

tive behaviors of turbulent flows. Although scientists have developed valid governing equations to describe these physical systems, simulating such high-fidelity complex PDEs requires massive computational efforts, meanwhile posing significant challenges for storage space. With the limitation of computational and experimental resources, it is rather typical to store the spatiotemporally-downscaled data for post hoc analysis. This pressing issue calls for effective data augmentation approaches in scientific modeling for recovering the HR solutions from low-resolution (LR), e.g., sparse and incomplete data.

In order to upsample the LR scientific data to HR full-field dynamics, we refer to the idea of super-resolution (SR) in image and video processing. The image and video SR tasks are prohibitively ill-posed inverse problems considering the unknown blur and noise, the insufficient constraints and the non-uniqueness of reconstructed solutions. Previously, a seminal of breakthrough on single-image SR (SISR) has been made based on sparse signal representation [1], which is mathematically elegant and robust. Furthermore, due to the rapid development of deep learning (DL), there is another milestone work SRCNN [2] remarkably improving the performance of SISR by firstly introducing deep neural networks (DNNs). Later, large amounts of network architectures have been proposed and continuously achieved huge success in this intractable task, mainly covering the variants of generative adversarial networks (GAN) [3–5], residual blocks [6–9] and attention scheme [10–12], among others [13]. In the context of video SR, apart from the spatial upsampling, learning optical flow is another core and challenging issue. Many investigations leverage linear frame interpolation [14–16], quadratic video interpolation [17,18] and phase-based interpolation [19] for modeling temporal dependency and estimating multiple intermediate frames. Besides, the recurrent convolutional neural network is also proven to be an effective tool for video SR [20,21].

Thanks to the great success in image [1–3,7–9,11,22] and video super-resolution (SR) [14,16,21,23–25], considerable studies have grown up around the interplay between physics and machine learning (ML), with respect to SR of scientific data. The majority of existing investigations mainly cluster in purely data-driven ML methods which approximate the HR solutions based on potentially noisy and under-resolved simulations in turbulence [26–30], smoke [31] and climate [32] modeling. Nevertheless, different from general SR tasks for images and videos, scientific data augmentation has more explicit and well-defined objectives (e.g., obeying underlying governing equations), which could facilitate the convergence and generalization abilities with the physical principles incorporated into network architectures. There exist two primary branches of DL models informed by physics for dynamical systems: continuous learning [33–38] and discrete learning [39–49]. The continuous learning scheme typically employs fully-connected neural networks (NNs), which is excellent in capturing global solution patterns but generally limited to low-dimensional PDE systems. The discrete learning leverages convolutional/graph NNs for modeling nonlinear dynamics, projected on mesh grids, which is capable of well portraying the local solution details. In addition, there are many other significant approaches for learning nonlinear dynamics, including “white-box” models by embedding physics into the network [46,50–52] and neural operators [35,53,54].

Therefore, more recently, a flurry of work based on physics-informed DL has been recently developed for SR of scientific data [55–60]. Currently, the majority of physics-informed learning approaches for SR are limited to spatial super-resolution (S-SR) tasks (e.g., steady-state problems [57,58,60]), or only consider temporal super-resolution (T-SR) [55]. A notable latest study explored the potential of 3D convolutional operations for spatiotemporal super-resolution (ST-SR) [59], which exhibits excellent performance for 2D Rayleigh-Bénard convection (RBC) system. In this paper, we present a novel and effective ST-SR method for general spatiotemporal systems via physics-informed discrete learning. Distinct from directly using 3D convolution, we explicitly preserve the temporal evolution structure by decomposing the entire ST-SR into two modules, i.e., T-SR and S-SR. This decomposition is inspired by the observation that the temporal derivatives and spatial terms are independent in a general form of PDEs as illustrated in Section 3. This separable scheme is conducive to computational efficiency, as well as promotes flexibility and scalability to high-dimensional PDE systems (e.g., 3D systems).

Overall, our methodological contributions include: (1) a novel network architecture of physics-informed deep SR (PhySR) for scientific data, and (2) an intrinsic collaboration between high-order finite difference-based spatiotemporal filtering and hard-encoding of boundary conditions (BCs). Specifically, we apply interpolation to estimate the temporally upsampled solution, and further leverage a convolutional-recurrent network (i.e., ConvLSTM [61]) for learning temporal refinement and dynamical evolution on low-dimensional latent features during the T-SR process. In the second-stage (i.e., S-SR), we employ the stacked residual blocks with wide activation [8] to extract the spatial features and sub-pixel layers with pixelshuffle [22] for spatial upsampling. In addition, a global residual connection from the temporally-interpolated results to spatially-reconstructed variables is considered. Furthermore, the numerical differentiation determines the PDE derivative terms which are further utilized to construct PDE residual loss, and hard-imposition of BCs eliminates the risk of sacrificing solution accuracy on physical boundaries. Comprehensive numerical experiments on different types of PDE systems have been implemented for evaluating the performance of our proposed approach, compared with two baseline algorithms. Moreover, we validate the effectiveness of the significant modules of our PhySR by conducting an ablation study on ConvLSTM, the introduction of physics loss and the encoding of BCs. Finally, data degradation analysis is provided to offer more insights into the robustness of our PhySR framework accounting for modeling errors and noise effect in the training datasets.

The rest of the paper is organized as follows. Section 2 sets up the problem of ST-SR of spatiotemporal scientific data. Section 3 describes the network structure of two neural modules (i.e., T-SR and S-SR), hard-encoding strategy of BCs and physics-informed optimization. In Section 4, we show the extensive experiments between our proposed framework and baseline methods. In Section 5, we discuss the advantages, the limitations of our current work and the outlook of future directions. Section 6 concludes the entire paper.

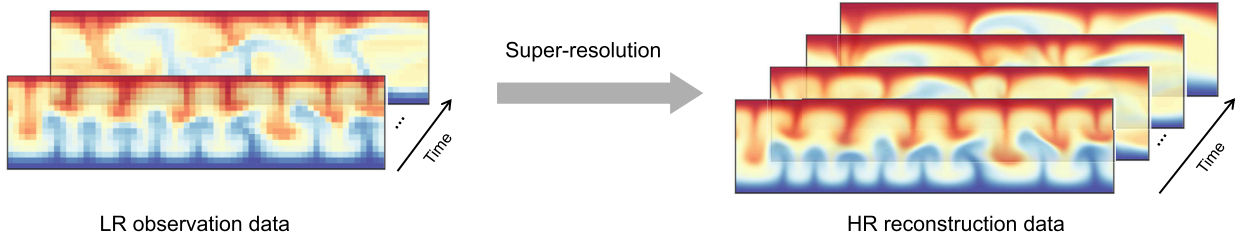


Fig. 1. Schematic of SR for 2D Rayleigh-Bénard convection system with respect to temperature. The LR observation data shown in the left part is measured at coarse-grid, and the HR reconstruction is based on fine-mesh with more time snapshots. Note that the spatial and temporal discretization considered here are both in uniform mode.

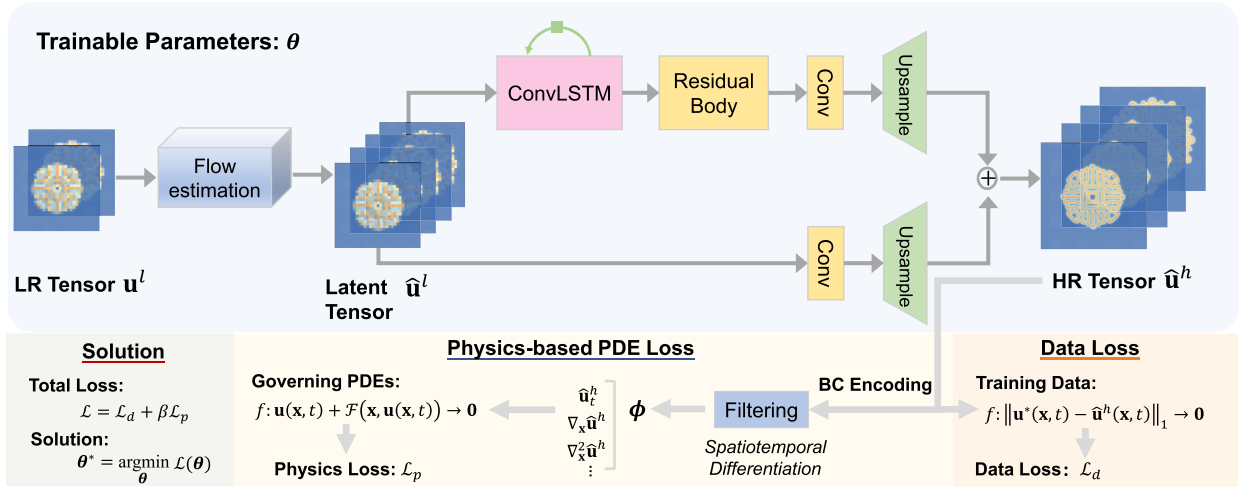


Fig. 2. Overview of the proposed PhySR framework. **Upper:** the network construction. We generate the intermediate frames by using temporal interpolation as flow estimation, and apply ConvLSTM for temporal refinement. To reconstruct the spatial resolution, we utilize the stacked residual blocks for feature extraction and sub-pixel shuffle for upsampling. Moreover, a global residual connection (i.e., in the lower arrow “Conv+Upsample”) is considered here from the estimated LR tensor $\hat{\mathbf{u}}^l$. **Bottom:** the loss components. Both the PDE residual loss \mathcal{L}_p and the data loss \mathcal{L}_d are computed based on the HR tensor $\hat{\mathbf{u}}^h$. Here the BCs are hard encoded into the network when calculating PDE derivatives through finite difference filtering.

2. Problem setup

Spatiotemporal dynamical systems are typically governed by PDEs. Let us consider a set of m -dimensional parametric PDEs in a general form:

$$\mathbf{u}_t = \mathcal{F}[\mathbf{x}, t, \mathbf{u}, \mathbf{u}_x, \mathbf{u}_y, \nabla \mathbf{u}, \dots; \lambda], \tag{1}$$

where \mathbf{x} and t represent spatial and time coordinates, respectively. $\mathbf{u}(\mathbf{x}, t)$ is the system solution comprised of n state variables and the subscripts denote the partial derivatives, e.g., $\mathbf{u}_x = \partial \mathbf{u} / \partial x$. ∇ denotes the Nabla operator (e.g., with the formulation of $[\partial / \partial x, \partial / \partial y]^T$ when $m = 2$). Moreover, $\mathcal{F}[\cdot]$ is the nonlinear functional consisting of polynomial and derivative terms and parameterized by λ . The specific initial and boundary conditions (I/BCs) are defined as $\mathcal{I}(\mathbf{u}, \mathbf{u}_t; t = 0, \mathbf{x} \in \Omega) = 0$ and $\mathcal{B}(\mathbf{u}, \nabla \mathbf{u}, \dots; \mathbf{x} \in \partial \Omega) = 0$ respectively, where $\partial \Omega$ is the boundary of the physical domain. Given LR data \mathbf{u}^l , our objective is to spatiotemporally recover the HR variables $\hat{\mathbf{u}}^h$ with consistent dynamical patterns, as shown in Fig. 1. Two constraints are considered in this paper to solve this ill-posed task: (1) reconstruction constraint, which requires the recovered HR data $\hat{\mathbf{u}}^h$ corresponding to the LR input measurement data \mathbf{u}^l with respect to the observed dynamical model; (2) physics prior, which assumes that LR and HR scientific data both obey the underlying governing equations.

3. Methodology

An overview of our proposed framework of PhySR is presented in Fig. 2. Based on the observation that time derivatives and spatial terms are mathematically separated, we design the network mainly consisting of the temporal upsampling (i.e., interpolation and refinement), as well as the spatial reconstruction. We first upscale the LR measurements to the desired frame rate in time dimension using the interpolation, and then apply ConvLSTM layers for refining the temporal approximation. Afterwards, for each snapshot, we leverage the residual blocks with wide activation and pixelshuffle layers for spatial upsampling. It is worthwhile to mention that all of the feature extraction processes are conducted in the LR

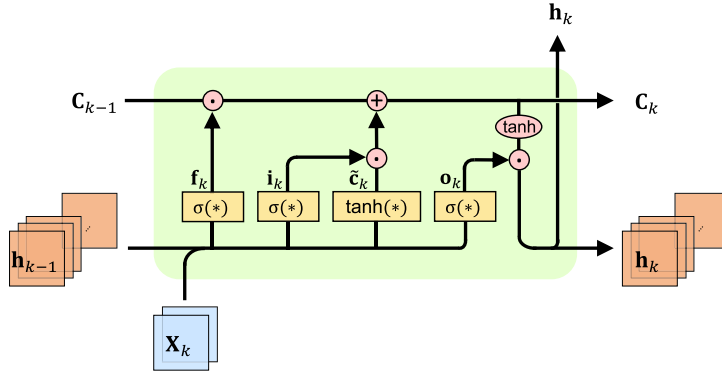


Fig. 3. Schematic of the ConvLSTM cell at time t_k . \mathbf{X}_k is the input variable. $\{\mathbf{h}_k, \mathbf{c}_k\}$ are hidden state and cell state, respectively.

latent space. The entire framework is “end-to-end”. In addition, we define the upscaling factors of time snapshots and spatial resolution as r_t and r_s respectively. In this section, we take two LR input frames $\{\mathbf{u}_{k-1}^l, \mathbf{u}_{k+1}^l\}$ at time $k - 1$ and $k + 1$ for illustration (i.e., $r_t = 2$), and attempt to recover the corresponding HR variables $\{\hat{\mathbf{u}}_{k-1}^h, \hat{\mathbf{u}}_{k+1}^h\}$ as well as synthesize the intermediate HR frame $\hat{\mathbf{u}}_k^h$ simultaneously.

3.1. ConvLSTM

Firstly, we introduce the critical neural component (i.e., ConvLSTM) in our PhysSR framework, especially for temporal upsampling. ConvLSTM is an extension from long short-term memory (LSTM) architecture [62–64] and specifically designed for learning spatiotemporal sequence-to-sequence models. It is noteworthy that LSTM is a variant of recurrent neural network (RNN) with the purpose of mitigating the issues of gradient vanishing and explosion. Moreover, a major advantage of ConvLSTM is that it works as an implicit and nonlinear numerical scheme for modeling dynamical systems [47].

The architecture of ConvLSTM cell at time t_k is presented in Fig. 3, including the gate scheme and memory cells. The general principle is to update the hidden state \mathbf{h}_k and cell state \mathbf{C}_k at time t_k by learning the transformation from the previous state variables $\{\mathbf{h}_{k-1}, \mathbf{C}_{k-1}\}$ and the current input variable \mathbf{X}_k . Four gates control the information flow for input-to-state and state-to-state transitions, including a forget gate \mathbf{f}_k , an input gate \mathbf{i}_k , an internal cell $\tilde{\mathbf{C}}_k$ and an output gate \mathbf{o}_k . In specific, the input and forget gates $\{\mathbf{i}_k, \mathbf{f}_k\}$ adaptively keep and clear the memory in \mathbf{C}_{k-1} . Moreover, a new cell candidate $\tilde{\mathbf{C}}_k$ is generated by synthesizing the information from input variable \mathbf{X}_k and previous hidden state \mathbf{h}_{k-1} . They all contribute to forming the updated cell state \mathbf{C}_k at the current time step t_k . Lastly, the hidden state \mathbf{h}_k is further updated by regulating the cell state \mathbf{C}_k based on the output gate \mathbf{o}_k . The analytical formulations for four gate operations and state variables are given by:

$$\begin{aligned} \mathbf{i}_k &= \sigma(\mathbf{W}_i * [\mathbf{X}_k, \mathbf{h}_{k-1}] + \mathbf{b}_i), & \mathbf{f}_k &= \sigma(\mathbf{W}_f * [\mathbf{X}_k, \mathbf{h}_{k-1}] + \mathbf{b}_f), \\ \tilde{\mathbf{C}}_{k-1} &= \tanh(\mathbf{W}_c * [\mathbf{X}_k, \mathbf{h}_{k-1}] + \mathbf{b}_c), & \mathbf{C}_k &= \mathbf{f}_k \odot \mathbf{C}_{k-1} + \mathbf{i}_k \odot \tilde{\mathbf{C}}_{k-1}, \\ \mathbf{o}_k &= \sigma(\mathbf{W}_o * [\mathbf{X}_k, \mathbf{h}_{k-1}] + \mathbf{b}_o), & \mathbf{h}_k &= \mathbf{o}_k \odot \tanh(\mathbf{C}_k), \end{aligned} \quad (2)$$

where $\tanh(*)$ and $\sigma(*)$ represent hyperbolic tangent activation and sigmoid functions, respectively. Here \odot and $*$ denote Hadamard product and the convolution operation. In addition, the weight and bias parameters of four gates in the ConvLSTM cell are correspondingly defined as $\{\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_c, \mathbf{W}_o\}$ and $\{\mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_c, \mathbf{b}_o\}$.

3.2. Pixel shuffle

Secondly, we leverage pixel shuffle to upscale LR latent features to HR variables, which is an efficient operation to conduct the sub-pixel convolutions [22]. The graphic illustration is shown in Fig. 4. Let us consider an example of the pixel shuffle operation in 2D setting. We define a LR feature map with the tensor shape of $(C \times r^2, H, W)$ where $\{r, C, H, W\}$ denote upscaling factor, channel size, height and width respectively. The pixel shuffle reforms the elements in the LR feature map to a HR tensor with a larger spatial resolution of $(H \times r, W \times r)$ and a shrunken feature size of C . Although it is a simple operation, the pixel shuffle still exhibits excellent SR performance in image and video tasks [22]. The second advantage is that it does not require huge computational efforts because the NN training before the spatial upsampling layer is based on the LR latent space. Furthermore, compared with other upscaling approaches (e.g., deconvolution), it leads to fewer checkerboard artifacts [65]. Overall, pixel shuffle is selected as the spatial upsampling strategy in this paper considering its contribution to computational and memory efficiency.

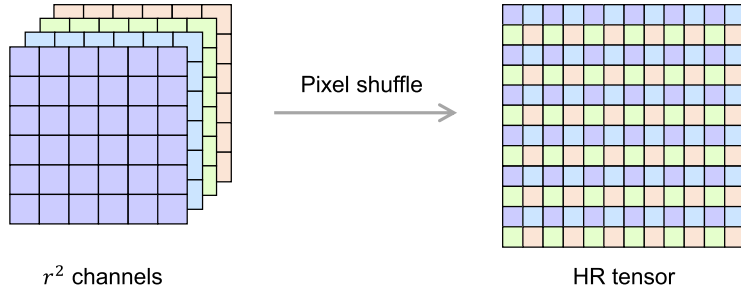


Fig. 4. Schematic of a pixel shuffle layer in 2D setting.

3.3. Temporal upsampling

In the temporal upsampling part, given the LR frames $\{\mathbf{u}_{k-1}^l, \mathbf{u}_{k+1}^l\}$, we first estimate the flow evolution to generate the intermediate LR frame $\widehat{\mathbf{u}}_k^l$ and then map them into the LR latent space as $\{\mathbf{h}_{k-1}, \mathbf{h}_k, \mathbf{h}_{k+1}\}$. Herein, we hybrid the traditional interpolation approaches and the deep convolutional-recurrent neural networks for initial flow estimation and temporal correction, respectively. The reason behind using interpolation is that we find it can effectively reduce the artifacts and oscillations in the time dimension, which is similarly reported in image SR [65].

3.3.1. Temporal interpolation

Different from the general video frame estimation, scientific data SR is equation-specific which promisingly inspires the direction of temporal interpolation model. For instance, for those PDEs with the first-order time derivative (i.e., \mathbf{u}_t), applying linear interpolation is capable of roughly approximating the slow-evolutionary dynamics. Besides, previous methods [14,21] also implement the interpolation on the learned features to synthesize the intermediate feature maps considering the complex motions in the real world. Nevertheless, here we can directly interpolate on the input LR frames considering the explicit and simple time derivative terms, which remarkably improves the computational efficiency. Furthermore, the following ConvLSTM layers can capture more straightforward and essential information from the input spatiotemporal dynamics compared with the learned features. Additionally, it helps to build a reliable global residual connection from the interpolated results and the final super-resolved variables.

Let us define a generic form of the interpolation function $f[\cdot]$, which only works for temporal upscaling. Therefore, the initial flow estimation $\widehat{\mathbf{u}}_k^l$ is formulated as:

$$\widehat{\mathbf{u}}_k^l = f[\mathbf{u}_{k-1}^l, \mathbf{u}_{k+1}^l]. \tag{3}$$

3.3.2. Temporal refinement

Based on the direct estimation from the temporal interpolation, we further improve the capability of synthesizing intermediate frames by introducing DNNs for learning correction. Here we consider the ConvLSTM architecture, which is a spatiotemporal sequence-to-sequence learning tool for modeling long-term temporal dependencies. Given the consecutive LR interpolated frames $\{\mathbf{u}_{k-1}^l, \widehat{\mathbf{u}}_k^l, \mathbf{u}_{k+1}^l\}$, we aim to apply ConvLSTM to obtain the corresponding LR latent features $\{\mathbf{h}_{k-1}, \mathbf{h}_k, \mathbf{h}_{k+1}\}$. Hence, the temporal updating at time t_k is given by

$$\mathbf{h}_k, \mathbf{c}_k = \mathcal{N}\mathcal{N}^t[\mathbf{h}_{k-1}, \mathbf{c}_{k-1}, \widehat{\mathbf{u}}_k^l; \theta^t], \tag{4}$$

where \mathbf{h}_k and \mathbf{c}_k are hidden state and cell state at time t_k ; $\mathcal{N}\mathcal{N}^t$ represents ConvLSTM networks and θ^t is the corresponding learnable parameters. Herein, the temporal refinement is implicitly captured through two previous states (i.e., \mathbf{h}_{k-1} and \mathbf{c}_{k-1}) and the current interpolated frame $\widehat{\mathbf{u}}_k^l$. Essentially, the learning process of Eq. (4) can be regarded as a forward-propagated state-space model with nonlinear activation function. Therefore, due to the similarity to the numerical scheme, ConvLSTM can effectively extract the dynamics from input LR measurements. Additionally, setting ConvLSTM on LR space can relieve the memory burden and facilitate the training efficiency.

3.4. Spatial reconstruction

After acquiring the sequence of LR feature maps $\{\mathbf{h}_{k-1}, \mathbf{h}_k, \mathbf{h}_{k+1}\}$, we propose to reconstruct the corresponding HR snapshots $\{\widehat{\mathbf{u}}_{k-1}^h, \widehat{\mathbf{u}}_k^h, \widehat{\mathbf{u}}_{k+1}^h\}$ by applying a temporally-shared fully-convolutional network. Generally, there are two parts for spatial upsampling. The first neural component is the stacked residual blocks with wide activation to extract the deep spatial features and one sub-pixel layer with pixelshuffle for upsampling, which contributes to increasing the spatial resolution fast and efficiently. Here we define the HR output at time t_k as $\mathcal{N}\mathcal{N}^s[\mathbf{h}_k; \theta^s]$.

Moreover, we consider adding a global residual connection directly from each interpolated LR snapshot $\widehat{\mathbf{u}}_k^l$ to the corresponding HR output $\mathcal{N}\mathcal{N}^r[\widehat{\mathbf{u}}_k^l; \theta^r]$, where single convolution layer and one sub-pixel layer are utilized for increasing the

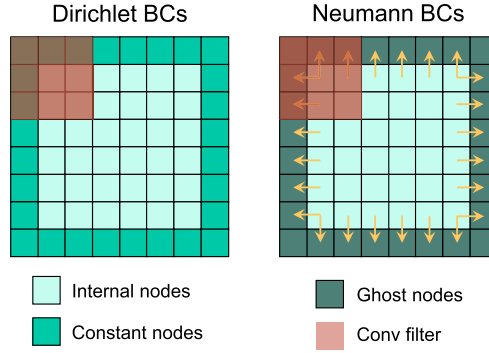


Fig. 5. A graphic illustration of hard encoding of BCs into the framework. We perform the hard enforcement of BCs both for NN learning and numerical differentiation.

number of channels to nr_s^m and further generating the HR counterparts respectively. Herein, $\mathcal{NN}^s[\cdot]$ and $\mathcal{NN}^r[\cdot]$ represent the first network component and the global residual network separately. θ^s and θ^r are the corresponding trainable parameters. Therefore, the final synthesized HR output variable $\hat{\mathbf{u}}_k^h$ is formulated as

$$\hat{\mathbf{u}}_k^h = \mathcal{NN}^s[\mathbf{h}_k; \theta^s] + \mathcal{NN}^r[\hat{\mathbf{u}}_k^h; \theta^r]. \quad (5)$$

In addition, weight normalization [66] is employed in this work for alleviating the training difficulty.

3.5. Hard-encoding of BCs

Basically, SR is a typical ill-posed task due to the harsh requirement of extrapolation from limited measurement data. However, prior knowledge is always accessible for SR of scientific data, including the specific governing equations and the I/BCs. The physical information can provide accurate description of the underlying dynamics and effective guidance for extrapolation. Therefore, here we propose to hard impose the BCs into the network to facilitate network training.

Considering the uniform discretization of the spatial domains, it is easy to encode the BCs by using padding pixel-wisely. The graphical depiction is shown in Fig. 5. Firstly, for Dirichlet BCs, we rigorously incorporate the known boundary values into the HR tensor $\hat{\mathbf{u}}^h$. Secondly, for Neumann BCs, we apply ghost nodes [67] for inferring the external node values via finite difference (FD) methods, where the physical correlation between ghost nodes and internal nodes is time-invariant.

3.6. Physics-informed loss function

Since SR of scientific data is equation-specific, it is beneficial and necessary to also consider the PDE residual loss apart from the traditional data loss between reconstructed HR variables and the ground-truth HR simulations. Here we apply the numerical differentiation on HR variables $\hat{\mathbf{u}}^h$ to approximate the derivatives of interest. The discussion on the selection of FD stencils and the introduction of numerical errors is presented in Appendix A. Furthermore, the specific PDE residual is formulated as $\mathcal{R}(\mathbf{x}, t; \theta) := \hat{\mathbf{u}}_t^h + \mathcal{F}[\hat{\mathbf{u}}^h; \lambda]$ where θ is the NN parameters. For a two-dimensional PDE, the physics loss is finally given by

$$\mathcal{L}_p = \frac{1}{N_x N_y N_t N_b} \|\mathcal{R}(\mathbf{x}, t; \theta)\|_2, \quad (6)$$

where N_x, N_y are the width and height in the spatial grid; N_t is the number of time steps; N_b is the batch size; $\|\cdot\|_2$ denotes Frobenius ℓ_2 norm. Moreover, the data loss is measured by the averaged ℓ_1 norm between HR ground truth \mathbf{u}^* and reconstructed variables $\hat{\mathbf{u}}^h$, which is formulated as

$$\mathcal{L}_d = \frac{1}{N_x N_y N_t N_b} \|\mathbf{u}^* - \hat{\mathbf{u}}^h\|_1. \quad (7)$$

Next, the total loss is a weighted combination of data loss \mathcal{L}_d and physics loss \mathcal{L}_p for optimizing the network by both considering the guidance of HR labeled data and the constraint of physical laws. It is given by

$$\mathcal{L} = \mathcal{L}_d + \beta \mathcal{L}_p, \quad (8)$$

where β is the weighting coefficient.

4. Experiments

In this section, we conduct extensive numerical experiments to evaluate the performance of our proposed approach on different types of spatiotemporal systems, ranging from convection phenomena to reaction-diffusion (RD) dynamics. The specific investigations include: (1) a comparative study on three PDE systems against the baseline models; (2) an ablation study on the effectiveness of ConvLSTM, the introduction of physics loss and the encoding of BCs; and (3) a data degradation study on a different downsampling method, large downscaling factors and noisy scenarios. The results indicate the superiority of our framework in reconstruction accuracy, computational efficiency and scalability.

4.1. Setup

4.1.1. Datasets

In the numerical experiments, we first consider 2D RBC system [59], which is a paradigm for nonlinear and chaotic dynamics. The RBC model describes the buoyancy-driven flow of a fluid heated from below and cooled from above and contains a non-trivial coupling between velocity, pressure and temperature. It is one of the most extensively investigated physical phenomena in geophysics, astrophysics, meteorology, oceanography and engineering. The governing equations for RBC system are given by

$$\begin{aligned} \nabla \cdot \mathbf{u} &= \mathbf{0}, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} &= -\nabla p + T \mathbf{e}_x + R^* \nabla^2 \mathbf{u}, \\ \frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T &= P^* \nabla^2 T, \end{aligned} \quad (9)$$

where $\mathbf{u} = [u, v]$ denotes the velocity components in x and y directions and p, T refer to pressure and temperature terms. Here R^* and P^* are $(P_r/R_a)^{0.5}$ and $(R_a P_r)^{-0.5}$ respectively, where R_a and P_r represent dimensionless Rayleigh and Prandtl numbers with expression of $g\alpha(\Delta T)h^3/(\nu\kappa)$ and ν/κ . $\{g, \alpha, \Delta T, h, \nu, \kappa\}$ are gravity acceleration, thermal expansion coefficient, the temperature difference between the top and bottom walls, the length between the plates, the kinematic viscosity and the heat conductivity coefficient. In addition, \mathbf{e}_x denotes the unit vector in x direction. Herein, we generate the HR ground truth dataset by using the open-source code in [59] with R_a and P_r as 1×10^6 and 1 respectively. The spatial domain of $[0, 4] \times [0, 1]$ is discretized as 512×128 , and the time period is $[0, 16]$ for 1600 steps. Next, we divide the HR dataset into 100 samples with 16 snapshots and a spatial resolution of 512×128 .

Apart from the convection phenomena, we also evaluate the performance of our proposed framework on RD equations. In particular, Gray-Scott (GS) RD equations are considered due to their depictions of complex dynamical patterns and local details, which work as an ideal system to test the predictive limit of the proposed approach [68]. In general, the GS RD system, which typically portrays complex chemical and biological process, is governed by

$$\mathbf{u}_t = \mathbf{D} \Delta \mathbf{u} + \mathbf{R}(\mathbf{u}) \quad (10)$$

where $\mathbf{u} = [u, v]^T$ denotes the concentration vector, $\mathbf{D} = [\gamma_u, 0; 0, \gamma_v]$ is the diagonal diffusion coefficient matrix and $\mathbf{R} = [-uv^2 + f(1-u), uv^2 - (f+k)v]^T$ represents the reaction vector. Moreover, we conduct the comparative study between our PhysSR against baseline models on both 2D and 3D GS RD dynamics. In specific, we set $\{\gamma_u, \gamma_v, f, k\}$ to be $\{0.16, 0.08, 0.06, 0.062\}$ for 2D case and $\{0.2, 0.1, 0.025, 0.055\}$ for 3D case of the GS RD systems. In order to train the network for 2D GS RD dynamics, we generate 400 reference solutions/samples, where the spatial grid size is 256×256 with $\delta x = 1$ and the simulated time period is $[0, 200]$ with $\delta t = 10$. In light of the intensive computation for 3D GS RD case, we generate 400 reference HR samples with the Cartesian grid of $48 \times 48 \times 48$ for $\mathbf{x} \in [0, 100]^3$ and 20 time steps for $t \in [0, 20]$ (i.e., $\delta t = 1$). All of the GS RD datasets are obtained via the high-order FD method with periodic BCs.

Furthermore, the HR labeled datasets are downsampled uniformly in the spatiotemporal domains, and blurred simultaneously for spatial resolutions. Herein, we design two ST-SR scenarios with different downsampling ratios $r_{(c)}$ for evaluation: (1) $r_t = 4, r_s = 8$ for 2D RBC and 2D GS dynamics; (2) $r_t = 2, r_s = 4$ for 3D GS system. In addition, we set the splitting ratio for training and testing datasets as 70% and 30%. All of the results reported in the experiments (i.e., comparative study, ablation study, and data degradation study) are obtained from the testing dataset.

4.1.2. Baselines

Although there are various baseline models for the traditional video ST-SR task, they are not specifically designed for ST-SR of scientific data, which may cause the unfairness in the experimental evaluation. To this end, here we consider the state-of-the-art baseline model MeshfreeFlowNet [59] in the field of scientific ST-SR for the performance comparison. In addition, we consider the classical interpolation approach (i.e., trilinear interpolation for 2D PDEs and quadlinear interpolation for 3D PDE case) as a benchmark method for comparison.

Regarding the MeshfreeFlowNet, we adopt the same network architecture¹ as in [59], i.e., a convolutional encoder named the Context Generation Network to map the LR inputs to a latent context grid, followed by a fully connected neural network (FCNN) called Continuous Decoding Network to generate continuous spatiotemporal solutions. The Context Generation Network is implemented with 3D variant of the U-Net [69] with residual blocks. The loss function used for training the MeshfreeFlowNet also follows the form of Eq. (8), i.e., a linear combination of data (or regression) and physics loss. The weighting coefficient β for 2D RBC system is set as 0.05 which is the optimal value reported in [59], and we select $\beta = 0.025$ for 2D GS case after empirical tests with $\{0.01, 0.025, 0.05, 0.1, 0.5, 1, 5, 10\}$. It should be noted that the physics loss in MeshfreeFlowNet is computed via automatic differentiation since the fully connected Continuous Decoding Network is used for generating the HR solution. The training is conducted on four NVIDIA Tesla V100 GPU cards (32G) due to the limit of our computational resources.

4.1.3. Evaluation metrics

In this paper, we consider using relative full-field ℓ_2 error ϵ to measure the discrepancy between the ground truth HR dynamics \mathbf{u}^* and the reconstructed HR variables $\hat{\mathbf{u}}^h$. Specifically, it has the formulation of

$$\epsilon = \sqrt{\frac{\|\mathbf{u}^* - \hat{\mathbf{u}}^h\|_2}{\|\mathbf{u}^*\|_2}} \times 100\%. \quad (11)$$

Moreover, we also compare our PhySR with two baseline models on other three metrics, including (1) the number of trainable parameters, (2) the training time T_{train} and (3) the inference time T_{infer} . These metrics are employed for evaluating computational and memory efficiency.

4.1.4. Implementation

Specifically, in this paper, we consider periodic BCs for the numerical experiments of 2D and 3D GS RD equations where periodic padding (i.e., circular padding in Pytorch [70]) is adopted in our network both for feature mapping and HR outputs. For the RBC system, the boundary information is unknown but we also apply periodic padding for feature mapping. The similar padding operation appears in another forecasting work of RBC system [42].

We train our PhySR using the Adam optimizer [71] with 1000 epochs for 2D RBC system and 4000 epochs for both 2D and 3D GS RD cases. The learning rate is set as 0.001 and the weight decay is 1×10^{-6} . The batch size is defined as 16. We use 2 residual blocks with the feature channel of 32 for both 2D and 3D cases. The size of the convolution kernel is set to be 3. Note that the weighting coefficient β in Eq. (8) is set as 0.025 for all three PDE systems after empirical trials of $\{0.01, 0.025, 0.05, 0.1, 0.5, 1, 5, 10\}$. While training the PhySR models, we normalize the input LR measurements by the mean and standard deviation of the HR ground truth. All of the PhySR models are implemented with PyTorch and trained on a NVIDIA Tesla V100 GPU card (32G) in a standard workstation.

4.2. Comparative study

We compare the ST-SR performance between PhySR and the baselines by using three evaluation metrics as shown in Section 4.1.3 and the reconstructed HR snapshots. The main finding is that our PhySR has much fewer parameters compared with MeshfreeFlowNet but holds the best reconstruction accuracy among all applicable methods. Moreover, PhySR is computationally efficient, which is consistently faster than MeshfreeFlowNet in both training and inference. Regarding the classical interpolation methods (i.e., trilinear and quadlinear), the expansion of dimensions induces an significant increase in inference time, which makes it undesirable for recovering high-dimensional complex systems.

4.2.1. 2D Rayleigh-Bénard convection system

The first case is the 2D RBC system. From Table 1, we can see that both PhySR and MeshfreeFlowNet predict the HR dynamics well and achieve excellent reconstruction accuracy (ϵ under 5%), while the traditional interpolation method (i.e. trilinear interpolation) presents a non-negligible error ($\epsilon > 20\%$) and exhibits obvious weakness in high-dimensional upsampling. In addition, the slight superiority of PhySR is rooted in the explicit modeling of temporal evolution by introducing the ConvLSTM architecture to PhySR. Nevertheless, MeshfreeFlowNet directly applies 3D convolution operations to realize the temporal upscaling.

Moreover, we compare the snapshots of these ST-SR results to further understand the reconstructed spatiotemporal dynamics. As shown in Fig. 6, we observe the consistent results that PhySR presents excellent ST-SR performance and the reconstruction of MeshfreeFlowNet is also very close to the HR ground truth. The interpolation method can generally capture the dynamical behaviors but produces more vague reconstruction compared with the other two methods. It is also noteworthy that MeshfreeFlowNet has a larger number of network parameters, which causes longer training and inference time compared with PhySR. Overall, the comparative study on 2D RBC case validates the effectiveness and efficiency of PhySR.

¹ Except the number of the input and output channels are set to be 2 due to the state variables of u and v in our 2D and 3D GS RD datasets.

Table 1

The comparison between PhySR and baseline models. “# Parameters” refers to the number of trainable parameters. T_{train} and T_{infer} denote the training time per epoch and the inference time per time step, respectively. For MeshfreeFlowNet and PhySR, means and standard deviations are presented by performing 10 training times with different random seeds. Besides, “N/A” represents being inapplicable to the specific index, and “-” means failure for being out of memory due to 4D convolution. Note that “M” denotes million and “ms” is millisecond.

| PDEs | Methods | # Parameters [M] | T_{train} [s] | T_{infer} [ms] | Relative full-field error [%] |
|--------|-----------------|------------------|------------------------|-------------------------|-----------------------------------|
| 2D RBC | Interpolation | N/A | N/A | 0.095 | 20.78 |
| | MeshfreeFlowNet | 6.95 | 15.86 | 56.10 | 4.39 ± 0.32 |
| | PhySR | 0.29 | 4.70 | 11.15 | 2.17 ± 0.26 |
| 2D GS | Interpolation | N/A | N/A | 0.08 | 31.58 |
| | MeshfreeFlowNet | 4.28 | 27.71 | 56.06 | 18.31 ± 1.87 |
| | PhySR | 0.23 | 6.91 | 9.64 | 3.60 ± 0.27 |
| 3D GS | Interpolation | N/A | N/A | 364.91 | 31.42 |
| | MeshfreeFlowNet | - | - | - | - |
| | PhySR | 0.68 | 12.13 | 12.68 | 2.77 ± 0.19 |

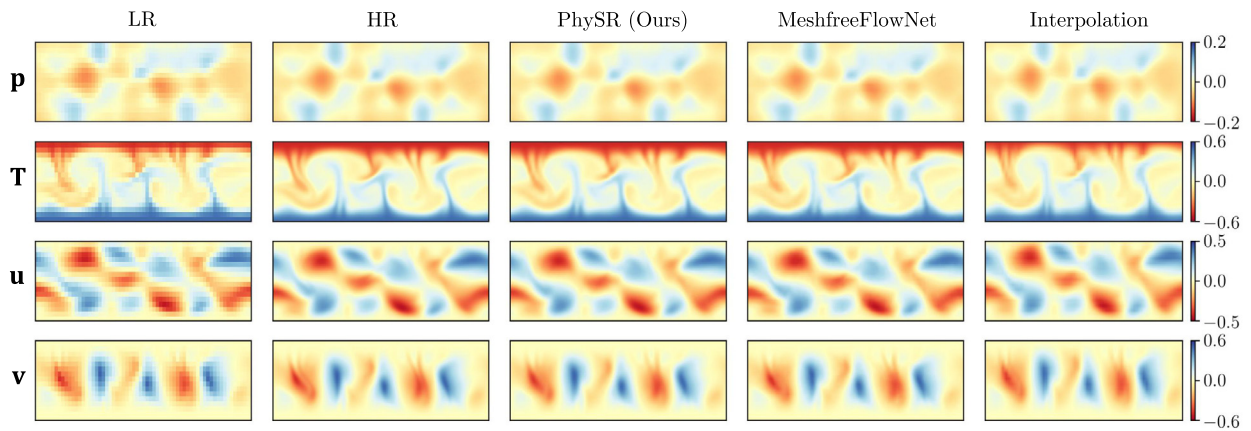


Fig. 6. The comparison of SR performance for 2D RBC system at representative snapshots, including four physical quantities (i.e., p , T , u , v). The corresponding Rayleigh and Prandtl numbers are 1×10^6 and 1. The spatial domain is $[0, 4] \times [0, 1]$, and the specific snapshot is selected at time $t = 14.4$.

4.2.2. 2D GS equation

RD dynamics are also tested for measuring the ST-SR performances of PhySR and baseline models in addition to the convection system. The second PDE system, 2D GS RD equation, also holds complex and rich dynamical patterns. The characteristic of 2D GS system lies in the delicate local features, which poses challenges to the network training. As presented in Table 1, it is evident that the relative full-field errors of MeshfreeFlowNet and trilinear interpolation models increase largely. It empirically indicates their limitations to such complicated spatiotemporal systems with various local details. Nevertheless, PhySR still holds the satisfactory ST-SR performance for 2D GS RD system though the error ϵ rises up to $3.60 \pm 0.27\%$ due to the complexity of the dynamical behaviors.

From Fig. 7(a), we find that the discrepancy between LR measurement data and HR ground truth is significant, which also implies the difficulty of reconstructing GS RD systems. Correspondingly, PhySR is still capable of portraying general patterns and evolutionary details of 2D GS dynamics, whereas MeshfreeFlowNet and trilinear barely discover the clear patterns. We ascribe the unsatisfactory results of MeshfreeFlowNet to two-folds: (1) error propagation due to lack of explicit temporal modeling; (2) the global learning scheme of FCNN module which tends to fail in capturing delicate local dynamics [45,47]. The experiment on 2D GS RD equation further validates the superiority of PhySR over baseline models thanks to the specified time marching strategy.

4.2.3. 3D GS equation

Thirdly, we consider a higher dimensional RD system (i.e., 3D GS equation) to evaluate the ST-SR performance of PhySR against two baseline methods. Note that MeshfreeFlowNet requires demanding computational memory for the 3D GS case by using 4D convolutional operations, which exceeds the limit of our server. Therefore, we mark it a failed test for MeshfreeFlowNet. In addition, we observe that the high-dimensional linear interpolation method (i.e., quadlinear) shows much slower inference performance compared with those for 2D cases, and the reconstruction results are still unsatisfactory. However, PhySR showcases outstanding performance regarding all these three challenging requirements: reconstruction accuracy, computational efficiency and scalability. As shown in Fig. 7(b), the representative predicted snapshots of PhySR match the corresponding HR ground truth well, which proves its remarkable capability for ST-SR of scientific data even in higher di-

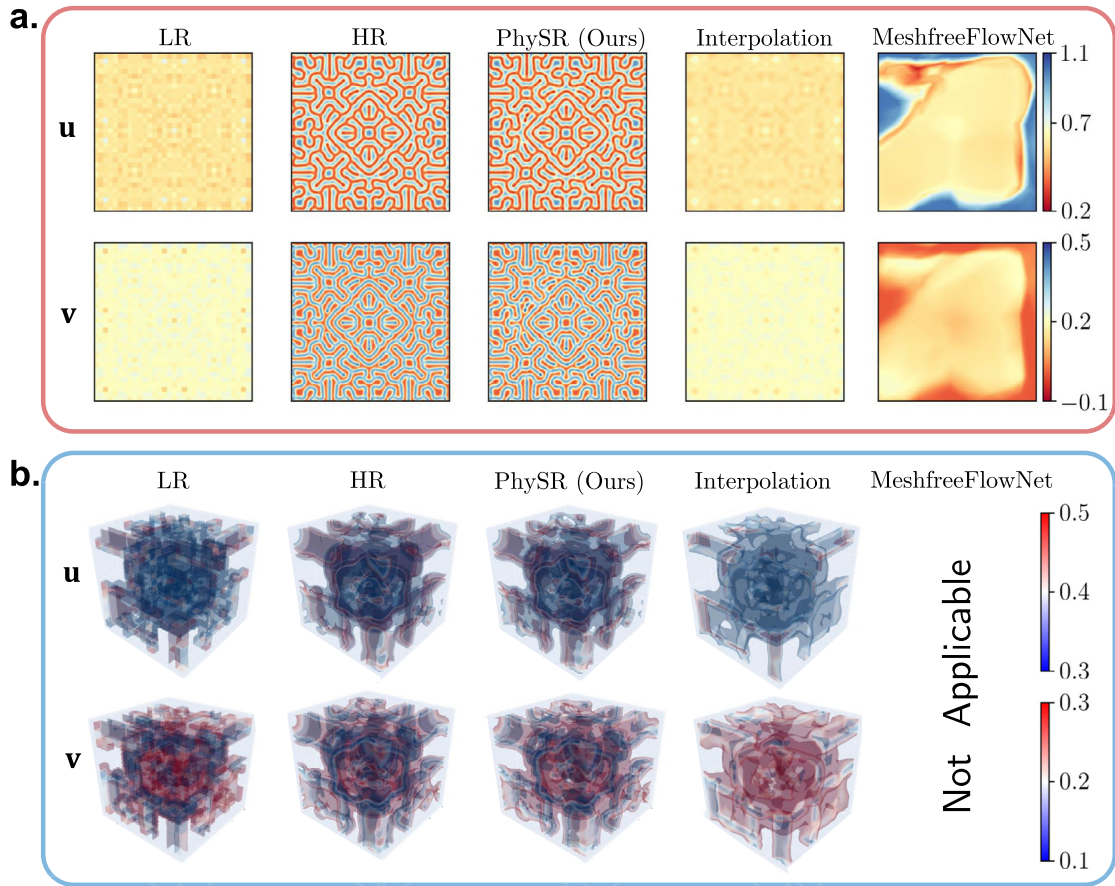


Fig. 7. The comparison of representative snapshots for GS RD systems. Here (a) and (b) represent the results for 2D and 3D GS equations respectively. The spatial domains are $[0, 256]^2$ for (a) and $[-50, 50]^2$ for (b). All of the snapshots are selected at time $t = 0.5$. Note that MeshfreeFlowNet for (b) is unavailable due to out-of-memory.

mensions. We can also see the positive evidences of PhySR in Table 1 that the reconstruction accuracy is appealing (ϵ under 5%) and the training and inference time increase mildly when extended to the 3D case.

4.3. Ablation study

To evaluate the effectiveness of each component of our PhySR framework, we perform the ablation study on 2D GS equation under ST-SR scenarios S_1 (shown in 4.1.1). Here we mainly investigate the contributions of ConvLSTM for temporal propagation of dynamics, the introduction of physics loss \mathcal{L}_p and the hard-encoding of BCs. These three components are specialized for ST-SR of scientific data, which makes them worthwhile to be further analyzed. Note that the hyper-parameters (i.e., $\beta = 0.025$) and the training effort are kept same for all numerical experiments in this part.

4.3.1. Effectiveness of ConvLSTM

Firstly, we examine the importance of ConvLSTM for temporal refinement. We consider two models: (A) the full PhySR with ConvLSTM, and (B) PhySR without ConvLSTM where the corresponding part is substituted by a standard convolution operation with the same channel size. From Fig. 8, we observe that the reconstruction snapshot of Model (B) clearly presents much blurrier results than that of Model (A). It indicates the relatively weaker capability of Model (B) in capturing the temporal evolution compared with Model (A) due to the lack of the explicit time marching module (i.e., ConvLSTM). Besides, the comparison of relative full-field errors between Model (A) and (B) in Table 2 further validates that the incorporation of ConvLSTM effectively contributes to temporal refinement and enhances the performance of temporal upsampling.

4.3.2. Effectiveness of physics loss \mathcal{L}_p

Secondly, to test the effectiveness of introducing physics loss \mathcal{L}_p , we design two comparative experiments for ST-SR of scientific data. We consider two models: (A) the full PhySR constrained by both data loss \mathcal{L}_d and physics loss \mathcal{L}_p , and (C) PhySR only trained with data loss \mathcal{L}_d . As shown in Fig. 8, the reconstruction of Model (C) presents relatively discontinuous contours (marked in black boxes) compared with the prediction snapshot of Model (A). This vague phenomenon is induced

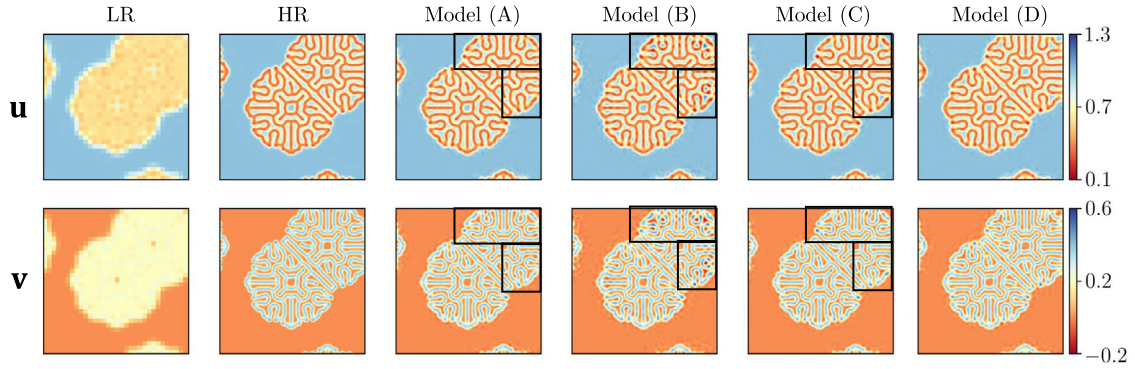


Fig. 8. The comparison of representative snapshots on ablation study. The evaluation is performed on 2D GS RD system. The spatial domain is $[0, 256]^2$.

Table 2

The comparative results of ablation study on 2D GS RD equations. Means and standard deviations are computed via 10 training times with different random seeds.

| Metric | Model (A) | Model (B) | Model (C) | Model (D) |
|-------------------------------|-----------------|-----------------|-----------------|-----------------|
| Relative full-field error [%] | 3.60 ± 0.27 | 4.49 ± 0.36 | 3.71 ± 0.32 | 4.32 ± 0.42 |

by the lack of the constraint of physical laws. Namely, embedding physical principles into the learning framework can facilitate the interpolation/extrapolation ability in ST-SR. In addition, we also show that the full PhySR model outperforms Model (C) in reconstruction accuracy (see Table 2), which further validates the significant of physics loss for ST-SR of scientific data.

4.3.3. Effectiveness of hard-encoding of BCs

Thirdly, we evaluate the effectiveness of hard-encoding BCs in the network. There are also two models designed for comparison: (A) the full PhySR with BCs encoded, and (D) PhySR without BCs incorporated. In this part, Model (D) is trained by replacing periodic padding to zero padding in feature mapping and does not consider the known BCs enforced into the output variables. It is apparent that the reconstructed results on the boundaries of Model (D) are not competitive as those of Model (A), especially on the top side (see Fig. 8). Moreover, as shown in Table 2, we observe the consistent performances that the ST-SR accuracy of Model (D) is inferior than that of Model (A). The effectiveness of encoding BCs is also experimentally validated. Hence, it is beneficial to incorporate BCs into learning machines when the boundary information is available.

4.4. Data degradation study

Furthermore, we further the performance of our proposed PhySR model using different training datasets accounting for modeling errors and noise effect. In specific, three data degradation scenarios are considered: (1) making LR simulation data as input, (2) exploring large downsampling factors, and (3) adding noise to LR input data.

4.4.1. LR simulation data

In this part, we investigate the effectiveness of our proposed PhySR framework with LR simulations as input data instead of using uniformly-downsampled data, which is described in Section 4.1.1. We test on 2D RBC system and follow all of the training details in Section 4.1.4. Note that the LR simulation data is directly generated with the spatial resolution of 64×16 and the time interval of 0.04. Other physical parameters are kept the same. We find that the relative full-field error is 4.46% for the testing datasets, which also indicates the good performance of PhySR to different a “downsampling” method. The representative snapshots are presented in Fig. 9. It is observed that there are slightly more discrepancies compared with the results based on uniformly-downsampled LR data since using LR simulation as inputs is a more complex downsampling scenario.

4.4.2. Large downsampling factors

In addition, it is worth exploring the influence of larger sub-sampling ratios on the reconstruction accuracy of HR predictions. We also consider evaluating the SR performance on 2D RBC system. We further decrease the downsampling ratios for LR input data and keep the spatial resolution of HR predictions the same (i.e., 512×128). In this numerical experiment, three large downscaling scenarios are defined: (1) $r_t = 8, r_s = 16$, (2) $r_t = 16, r_s = 16$, and (3) $r_t = 16, r_s = 32$. As shown in Table 3, the relative full-field errors increase gradually as the downsampling ratios become larger. It is also surprising to see that PhySR can still present great reconstruction performance ($\epsilon < 5\%$) for large spatiotemporal upscaling (e.g.,

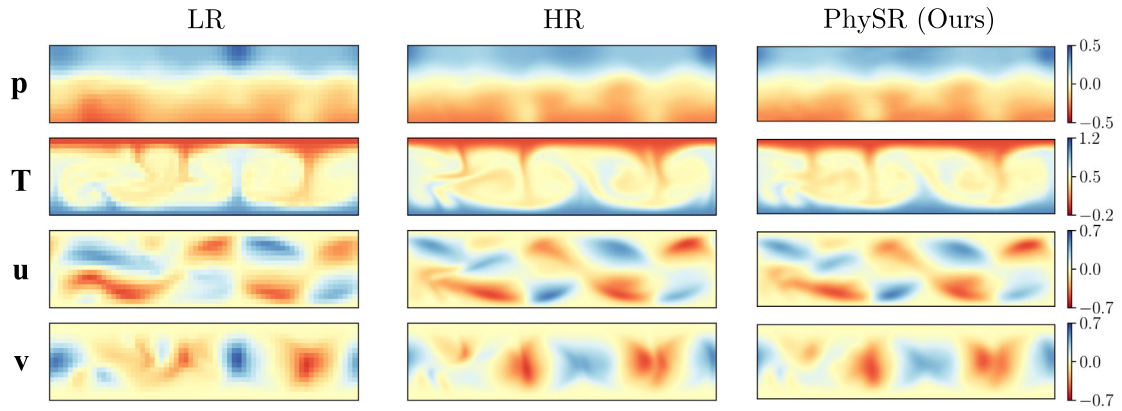


Fig. 9. Super-resolution of the 2D RBC dataset at representative snapshots by PhySR. The input data is LR simulation with the spatial resolution of 64×16 . The spatial resolution of the corresponding HR predictions is 512×128 . The spatial domain is $[0, 4] \times [0, 1]$.

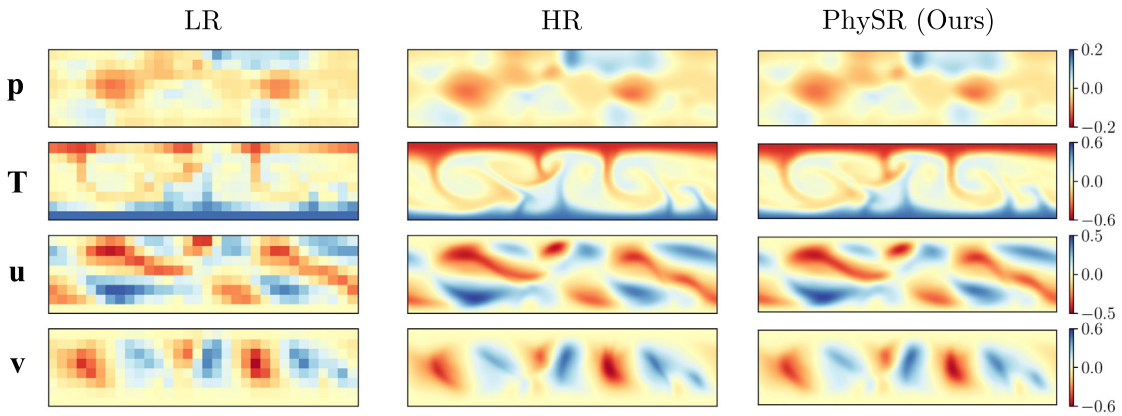


Fig. 10. Super-resolution of the 2D RBC dataset at representative snapshots by PhySR with a large upscaling factor ($r_t = 16, r_s = 16$). The spatial domain is $[0, 4] \times [0, 1]$ with the spatial resolution of 512×128 .

Table 3
The effect of downsampling factors on SR performance for 2D RBC equations.

| Metric | $r_t = 8, r_s = 16$ | $r_t = 16, r_s = 16$ | $r_t = 16, r_s = 32$ |
|-------------------------------|---------------------|----------------------|----------------------|
| Relative full-field error [%] | 4.02 | 4.36 | 7.95 |

$r_t = 16, r_s = 16$), which suggests the benefits of the incorporation of physical laws and the implicit time marching method (ConvLSTM). Fig. 10 exhibits the representative reconstruction snapshots with a large upscaling factor (i.e., $r_t = 16, r_s = 16$). Nevertheless, when further reducing the resolution of LR inputs with $r_t = 16$ and $r_s = 32$, the SR performance appears to be inferior. Note that the largest sub-sampling ratio may vary, which depends on the characteristics of the spatiotemporal system being studied.

4.4.3. Noisy scenarios

Noise issues are ubiquitous in real-world experiments and applications. Therefore, adding noise to LR synthetic data is also important to test the robustness of the proposed method for SR. In this part, we consider two noisy scenarios (i.e., 5% and 10% Gaussian noise) for evaluating the performance of PhySR on 2D RBC system. The noise level is based on the channel-wise standard deviation of the HR simulation dataset. The summarized results are presented in Table 4. As the noise level increases from 5% to 10%, the SR performance of our proposed PhySR exhibits a gradual decline but maintains an acceptable reconstruction accuracy. We also show the snapshots inferred from the learned model in Fig. 11, which is trained with low-fidelity data with 10% noise. It is apparent that PhySR can still capture detailed dynamic patterns and present excellent robustness to noisy scenarios.

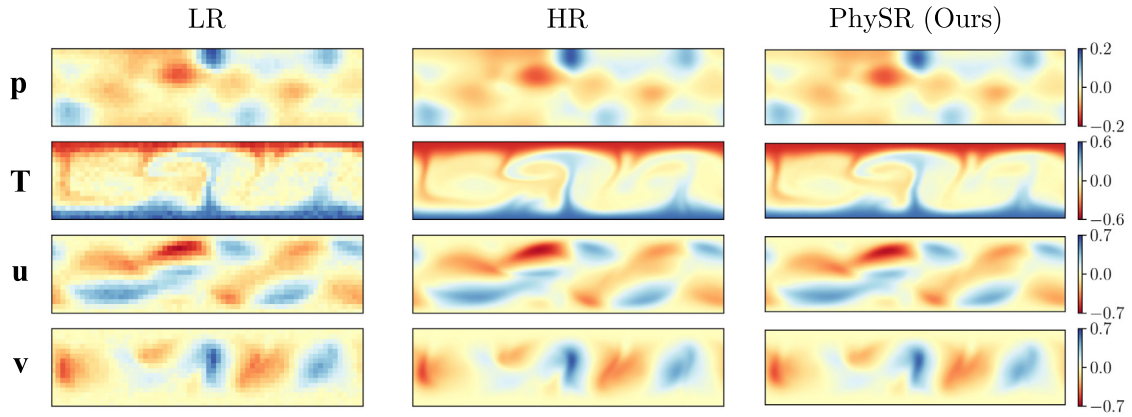


Fig. 11. Super-resolution of the 2D RBC dataset at representative snapshots by PhySR considering input noise. There is 10% noise added in input LR data. The spatial domain is $[0, 4] \times [0, 1]$ with the spatial resolution of 512×128 .

Table 4
The effect of noise on SR performance for 2D RBC equations.

| Metric | 5% noise | 10% noise |
|-------------------------------|----------|-----------|
| Relative full-field error [%] | 2.95 | 4.01 |

5. Discussion

In this section, we discuss the advantages and current limitations of the proposed discrete learning framework for ST-SR, as well as the possible solutions for future research directions. Firstly, there are three main benefits of our physics-informed discrete learning architecture, i.e., excellent reconstruction accuracy, efficiency and scalability. Regarding the ST-SR accuracy, PhySR outperforms baseline models due to the specialized time marching neural module (e.g., ConvLSTM) for learning temporal evolution and the incorporation of physical principles (i.e., governing equations and boundary information). Although there is a concern about the numerical errors caused by FD approximation of derivatives compared with the automatic differentiation through fully-connected NNs [33,59], we show that high-order FD kernels can be utilized to mitigate this issue in discrete learning frameworks, as reported in Appendix A. The reconstruction accuracy is improved when a higher-order FD kernel is chosen. In addition, we decompose the ST-SR task into temporal upsampling and spatial reconstruction. This setting helps to make the model scalable to higher-dimensional PDE systems (e.g., 3D GS RD equation) compared with the network directly using 3D/4D convolution. Moreover, in the part of spatial reconstruction, the spatial features are learned in LR latent space. Thus, a small convolutional filter (e.g., filter size of 3) can be applied for feature mapping, which facilitates the efficiency and reduces the computational memory. Although the performance of PhySR is tested on convection and RD systems, it is also applicable to other spatiotemporal systems where the PDE formulation is given.

However, there still remain several issues that are worthy to explore in the future. For example, the present network is rooted in convolution operation and limited to structured grids of computational domains. Leveraging graph neural networks (GNNs) is an alternative to resolve this problem, but it is challenging to determine the derivative terms based on irregular meshes and further assemble the PDE residuals. Secondly, the requirement for time stepping is demanding when dealing with very chaotic or fast-evolving systems because of the nature of discretized methods. We need to define a relatively smaller time interval (δt) for training compared with continuous learning architectures. Thirdly, the physics-informed learning essentially considers the physical laws as a soft regularizer, which makes the optimization and hyper-parameters tuning intractable. To handle this, encoding the PDE structures into the network can be a promising direction for learning a model that strictly obeys the underlying governing equations [46,52].

6. Conclusion

In this paper, we propose a novel and efficient physics-informed deep super-resolution framework (PhySR) for spatiotemporal scientific data. The network is comprised of the temporal interpolation, temporal refinement and spatial reconstruction modules, which is inspired by the independence between spatial and temporal derivative terms in a general PDE form. Specifically, we utilize the interpolation to directly increase the frame rate and ConvLSTM layers to correct the errors of temporal upsampling. Furthermore, we adopt residual blocks to extract LR features, sub-pixel layers for spatial upscaling and a global residual connection for stable propagation of spatial information. Besides, the BCs are hard-encoded into our network to facilitate convergence and improve the approximation on boundaries. The performance of our proposed framework has been assessed on three nonlinear PDE systems (i.e., 2D RBC system, 2D GS and 3D GS RD equations) and compared with baseline algorithms. The numerical results show the superiority of our method in reconstruction accuracy, efficiency

and scalability. Moreover, the ablation study validates the effectiveness of the incorporation of ConvLSTM and physics principles, and the data degradation analysis presents the robustness of PhySR.

Declaration of competing interest

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

Data availability

All the datasets and source codes to reproduce the results in this study are available on GitHub at <https://github.com/paulpuren/PhySR>.

Acknowledgement

The work is supported by the Beijing Natural Science Foundation (No. 1232009) and the National Natural Science Foundation of China (No. 62276269). H. Sun would like to thank the support from the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 202230265). Y. Liu would like to thank the support from the Fundamental Research Funds for the Central Universities (No. E2EG2202X2). P. Ren would like to gratefully acknowledge the Vilas Mujumdar Fellowship at Northeastern University.

Appendix A. Selection of FD kernels

FD kernels are used to approximate derivative terms in PDEs, which directly affect the computational efficiency and the reconstruction accuracy. Therefore, it is crucial to choose appropriate FD kernels for discretized-based learning frameworks. For spatiotemporal systems, we need to both consider temporal and spatial derivatives. In specific, second-order central difference is utilized for calculating temporal derivatives, i.e.,

$$\frac{\partial u}{\partial t} = \frac{-u(t - \delta t, \xi, \eta) + u(t + \delta t, \xi, \eta)}{2\delta t} + \mathcal{O}((\delta t)^2), \quad (\text{A.1})$$

where $\{\xi, \eta\}$ represent the spatial locations and δt is time spacing. In the network implementation, it can be organized as a convolutional kernel K_t ,

$$K_t = [-1, 0, 1] \times \frac{1}{2\delta t}. \quad (\text{A.2})$$

However, the derivatives at first and last steps are inaccessible due to the nature of central difference scheme. Herein, we substitute it with forward and backward Euler methods to compute the corresponding time derivatives (at first and last steps).

Likewise, we also apply central difference to calculate the spatial derivatives for internal nodes and use forward/backward difference for boundary nodes. For instance, in this paper, the fourth-order central difference is utilized to approximate the first and second spatial derivatives. The FD kernels for 2D cases with the shape of 5×5 are given by

$$K_{s,1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & -8 & 0 & 8 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \times \frac{1}{12(\delta x)}, \quad K_{s,2} = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 16 & 0 & 0 \\ -1 & 16 & -60 & 16 & -1 \\ 0 & 0 & 16 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{bmatrix} \times \frac{1}{12(\delta x)^2}, \quad (\text{A.3})$$

where δx denotes the grid size of HR variables; $K_{s,1}$ and $K_{s,2}$ are FD kernels for the first and second derivatives respectively. In addition, we conduct a parametric study on the selection of FD kernels, including the second-order (3×3), the fourth-order (5×5) and the sixth-order (7×7) central difference strategies. The numerical experiments are based on 2D GS RD

Table A.5

The comparative results on the selection of FD kernels. T_{train} denotes the training time per epoch. The experiments are conducted on 2D GS RD equations. For each scenario, we train the network 10 times with different random seeds to obtain the corresponding mean and standard deviation.

| FD kernels | T_{train} [s] | Relative full-field error [%] |
|--------------|------------------------|-------------------------------|
| 3×3 | 6.85 | 3.68 ± 0.27 |
| 5×5 | 6.91 | 3.60 ± 0.27 |
| 7×7 | 7.16 | 3.60 ± 0.23 |

system. As shown in Table A.5, the training time rises up and the reconstruction accuracy grows as the FD order increases. Nevertheless, we also notice that the PhySR with 5×5 FD kernel can already produce competitive results compared with the model using 7×7 FD kernel. It reflects that the reconstruction performance can be improved very slowly with a higher-order FD scheme (e.g., sixth-order) after the approximation accuracy of derivatives are satisfied. Therefore, we select the 5×5 FD kernel in this paper for the numerical experiments considering a trade-off between accuracy and efficiency. Generally, for those systems with complex patterns or fast-evolving features (e.g., the chaotic RBC system), it is more suitable to consider high-order FD kernels.

References

- [1] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, *IEEE Trans. Image Process.* 19 (11) (2010) 2861–2873.
- [2] C. Dong, C.C. Loy, K. He, X. Tang, Image super-resolution using deep convolutional networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 38 (2) (2015) 295–307.
- [3] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4681–4690.
- [4] W. Zhang, Y. Liu, C. Dong, Y. Qiao Ranksrgan, Generative adversarial networks with ranker for image super-resolution, in: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3096–3105.
- [5] K.C. Chan, X. Wang, X. Xu, J. Gu, C.C. Loy Glean, Generative latent bank for large-factor image super-resolution, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14245–14254.
- [6] J. Kim, J.K. Lee, K.M. Lee, Accurate image super-resolution using very deep convolutional networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1646–1654.
- [7] B. Lim, S. Son, H. Kim, S. Nah, K. Mu Lee, Enhanced deep residual networks for single image super-resolution, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 136–144.
- [8] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, T. Huang, Wide activation for efficient and accurate image super-resolution, *arXiv preprint, arXiv: 1808.08718*, 2018.
- [9] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, Image super-resolution using very deep residual channel attention networks, in: *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 286–301.
- [10] F. Yang, H. Yang, J. Fu, H. Lu, B. Guo, Learning texture transformer network for image super-resolution, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5791–5800.
- [11] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, W. Gao, Pre-trained image processing transformer, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 12299–12310.
- [12] Q. Zhou, C. Du, H. He, Exploring the brain-like properties of deep neural networks: a neural encoding perspective, *Mach. Intell. Res.* 19 (2022) 439–455.
- [13] D.-P. Fan, Z. Huang, P. Zheng, H. Liu, X. Qin, L.V. Gool, Facial-sketch synthesis: a new challenge, *Mach. Intell. Res.* 19 (2022) 257–287.
- [14] S. Niklaus, L. Mai, F. Liu, Video frame interpolation via adaptive separable convolution, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 261–270.
- [15] Z. Liu, R.A. Yeh, X. Tang, Y. Liu, A. Agarwala, Video frame synthesis using deep voxel flow, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4463–4471.
- [16] H. Jiang, D. Sun, V. Jampani, M.-H. Yang, E. Learned-Miller, J. Kautz, Super slo-mo: high quality estimation of multiple intermediate frames for video interpolation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9000–9008.
- [17] X. Xu, L. Si-Yao, W. Sun, Q. Yin, M. Yang, Quadratic video interpolation, in: *NeurIPS*, 2019, pp. 1645–1654.
- [18] Y. Liu, L. Xie, L. Siyao, W. Sun, Y. Qiao, C. Dong, Enhanced quadratic video interpolation, in: *European Conference on Computer Vision*, Springer, 2020, pp. 41–56.
- [19] S. Meyer, A. Djelouah, B. McWilliams, A. Sorkine-Hornung, M. Gross, C. Schroers, Phasenet for video frame interpolation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 498–507.
- [20] M.S. Sajjadi, R. Vemulapalli, M. Brown, Frame-recurrent video super-resolution, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6626–6634.
- [21] X. Xiang, Y. Tian, Y. Zhang, Y. Fu, J.P. Allebach, C. Xu, Zooming slow-mo: fast and accurate one-stage space-time video super-resolution, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3370–3379.
- [22] W. Shi, J. Caballero, F. Huszár, J. Totz, A.P. Aitken, R. Bishop, D. Rueckert, Z. Wang, Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.
- [23] D. Sun, X. Yang, M.-Y. Liu, J. Kautz, Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8934–8943.
- [24] S. Niklaus, F. Liu, Context-aware synthesis for video frame interpolation, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1701–1710.
- [25] X. Wang, K.C. Chan, K. Yu, C. Dong, C. Change Loy, Edvr: video restoration with enhanced deformable convolutional networks, in: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0 2019.
- [26] Y. Xie, E. Franz, M. Chu, N. Thurey, tempogan: a temporally coherent, volumetric gan for super-resolution fluid flow, *ACM Trans. Graph. (TOG)* 37 (4) (2018) 1–15.
- [27] Y. Bar-Sinai, S. Hoyer, J. Hickey, M.P. Brenner, Learning data-driven discretizations for partial differential equations, *Proc. Natl. Acad. Sci.* 116 (31) (2019) 15344–15349.
- [28] B. Liu, J. Tang, H. Huang, X.-Y. Lu, Deep learning methods for super-resolution reconstruction of turbulent flows, *Phys. Fluids* 32 (2) (2020) 025105.
- [29] K. Fukami, K. Fukagata, K. Taira, Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows, *J. Fluid Mech.* 909 (2021).
- [30] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning-accelerated computational fluid dynamics, *Proc. Natl. Acad. Sci.* 118 (21) (2021).
- [31] K. Bai, W. Li, M. Desbrun, X. Liu, Dynamic upsampling of smoke through dictionary-based learning, *ACM Trans. Graph. (TOG)* 40 (1) (2020) 1–19.
- [32] K. Stengel, A. Glaws, D. Hettinger, R.N. King, Adversarial super-resolution of climatological wind and solar data, *Proc. Natl. Acad. Sci.* 117 (29) (2020) 16805–16815.
- [33] M. Raissi, P. Perdikaris, 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.* 378 (2019) 686–707.
- [34] M. Raissi, A. Yazdani, G.E. Karniadakis, Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations, *Science* 367 (6481) (2020) 1026–1030.

- [35] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via deepnet based on the universal approximation theorem of operators, *Nat. Mach. Intell.* 3 (3) (2021) 218–229.
- [36] C. Rao, H. Sun, Y. Liu, Physics-informed deep learning for computational elastodynamics without labeled data, *J. Eng. Mech.* 147 (8) (2021) 04021043.
- [37] G.E. Karniadakis, I.G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, Physics-informed machine learning, *Nat. Rev. Phys.* 3 (6) (2021) 422–440.
- [38] Z. Chen, Y. Liu, H. Sun, Physics-informed learning of governing equations from scarce data, *Nat. Commun.* 12 (2021) 6136.
- [39] Y. Long, X. She, S. Mukhopadhyay, Hybridnet: integrating model-based and data-driven learning to predict evolution of dynamical systems, in: *Conference on Robot Learning*, PMLR, 2018, pp. 551–560.
- [40] Y. Zhu, N. Zabarar, P.-S. Koutsourelakis, P. Perdikaris, Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, *J. Comput. Phys.* 394 (2019) 56–81.
- [41] N. Geneva, N. Zabarar, Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks, *J. Comput. Phys.* 403 (2020) 109056.
- [42] R. Wang, K. Kashinath, M. Mustafa, A. Albert, R. Yu, Towards physics-informed deep learning for turbulent flow prediction, in: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1457–1466.
- [43] R. Zhang, Y. Liu, H. Sun, Physics-informed multi-lstm networks for metamodeling of nonlinear structures, *Comput. Methods Appl. Mech. Eng.* 369 (2020) 113226.
- [44] R. Zhang, Y. Liu, H. Sun, Physics-guided convolutional neural network (phycnn) for data-driven seismic response modeling, *Eng. Struct.* 215 (2020) 110704.
- [45] H. Gao, L. Sun, J.-X. Wang, Phygeonet: physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain, *J. Comput. Phys.* 428 (2021) 110079.
- [46] C. Rao, P. Ren, Q. Wang, O. Buyukozturk, H. Sun, Y. Liu, Encoding physics to learn reaction-diffusion processes, *Nat. Mach. Intell.* 428 (2023) 765–779.
- [47] P. Ren, C. Rao, Y. Liu, J.-X. Wang, H. Sun, PhyCRNet: physics-informed convolutional-recurrent network for solving spatiotemporal PDEs, *Comput. Methods Appl. Mech. Eng.* 389 (2022) 114399.
- [48] H. Gao, M.J. Zahr, J.-X. Wang, Physics-informed graph neural Galerkin networks: a unified framework for solving PDE-governed forward and inverse problems, *Comput. Methods Appl. Mech. Eng.* 390 (2022) 114502.
- [49] X.-Y. Liu, H. Sun, M. Zhu, L. Lu, J.-X. Wang, Predicting parametric spatiotemporal dynamics by multi-resolution PDE structure-preserved deep learning, *arXiv preprint*, arXiv:2205.03990, 2022.
- [50] Z. Long, Y. Lu, X. Ma, B. Dong, PDE-Net: learning PDEs from data, in: *International Conference on Machine Learning*, PMLR, 2018, pp. 3208–3216.
- [51] Z. Long, Y. Lu, B. Dong, PDE-Net 2.0: learning PDEs from data with a numeric-symbolic hybrid deep network, *J. Comput. Phys.* 399 (2019) 108925.
- [52] C. Rao, P. Ren, Y. Liu, H. Sun, Discovering nonlinear PDEs from scarce data with physics-encoded learning, in: *International Conference on Learning Representations*, 2022, pp. 1–19.
- [53] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: graph kernel network for partial differential equations, *arXiv preprint*, arXiv:2003.03485, 2020.
- [54] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, *arXiv preprint*, arXiv:2010.08895, 2020.
- [55] J. Han, C. Wang, TSR-TVD: Temporal super-resolution for time-varying data analysis and visualization, *IEEE Trans. Vis. Comput. Graph.* 26 (1) (2019) 205–215.
- [56] M. Bode, M. Gauding, Z. Lian, D. Denker, M. Davidovic, K. Kleinheinz, J. Jitsev, H. Pitsch, Using physics-informed super-resolution generative adversarial networks for subgrid modeling in turbulent reactive flows, *arXiv preprint*, arXiv:1911.11380, 2019.
- [57] C. Wang, E. Bentivegna, W. Zhou, L. Klein, B. Elmegeen, Physics-informed neural network super resolution for advection-diffusion models, *arXiv preprint*, arXiv:2011.02519, 2020.
- [58] A. Subramaniam, M.L. Wong, R.D. Borker, S. Nimmagadda, S.K. Lele, Turbulence enrichment using physics-informed generative adversarial networks, *arXiv e-prints*, 2020.
- [59] S. Esmailzadeh, K. Azizzadenesheli, K. Kashinath, M. Mustafa, H.A. Tchelepi, P. Marcus, M. Prabhat, A. Anandkumar, et al., Meshfreeflownet: a physics-constrained deep continuous space-time super-resolution framework, in: *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, IEEE, 2020, pp. 1–15.
- [60] H. Gao, L. Sun, J.-X. Wang, Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels, *Phys. Fluids* 33 (7) (2021) 073603.
- [61] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional LSTM network: a machine learning approach for precipitation nowcasting, *Adv. Neural Inf. Process. Syst.* 28 (2015) 802–810.
- [62] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.
- [63] A. Graves, Generating sequences with recurrent neural networks, *arXiv preprint*, arXiv:1308.0850, 2013.
- [64] I. Sutskever, O. Vinyals, Q.V. Le, Sequence to sequence learning with neural networks, *arXiv preprint*, arXiv:1409.3215, 2014.
- [65] A. Odena, V. Dumoulin, C. Olah, Deconvolution and checkerboard artifacts, *Distill* 1 (10) (2016) e3.
- [66] T. Salimans, D.P. Kingma, Weight normalization: a simple reparameterization to accelerate training of deep neural networks, *Adv. Neural Inf. Process. Syst.* 29 (2016) 901–909.
- [67] T.J. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Courier Corporation, 2012.
- [68] N. Geneva, N. Zabarar, Transformers for modeling physical systems, *Neural Netw.* 146 (2022) 272–289.
- [69] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.
- [70] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an imperative style, high-performance deep learning library, *Adv. Neural Inf. Process. Syst.* 32 (2019) 8026–8037.
- [71] D.P. Kingma, J. Ba Adam, A method for stochastic optimization, *arXiv preprint*, arXiv:1412.6980, 2014.