CDAnet: A Physics-Informed Deep Neural Network for Downscaling Fluid Flows

Mohamad Abed El Rahman Hammoud1, Edriss S. Titi1, Ibrahim Hoteit1, and Omar Knio1

1King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, 2Department of Applied Mathematics and Theoretical Physics, University of Cambridge, Cambridge, UK, 3Department of Mathematics, Texas A&M University, College Station, TX, USA

Abstract Generating high-resolution flow fields is of paramount importance for various applications in engineering and climate sciences. This is typically achieved by solving the governing dynamical equations on high-resolution meshes, suitably nudged toward available coarse-scale data. To alleviate the computational cost of such downscaling process, we develop a physics-informed deep neural network (PI-DNN) that mimics the mapping of coarse-scale information into their fine-scale counterparts of continuous data assimilation (CDA). Specifically, the PI-DNN is trained within the theoretical framework described by Foias et al. (2014, https://doi.org/10.1070/rm2014v069n02abeh004891) to generate a surrogate of the theorized determining form map from the coarse-resolution data to the fine-resolution solution. We demonstrate the PI-DNN methodology through application to 2D Rayleigh-Bénard convection, and assess its performance by contrasting its predictions against those obtained by dynamical downscaling using CDA. The analysis suggests that the surrogate is constrained by similar conditions, in terms of spatio-temporal resolution of the input, as the ones required by the theoretical determining form map. The numerical results also suggest that the surrogate's downscaled fields are of comparable accuracy to those obtained by dynamically downscaling using CDA. Consistent with the analysis of Farhat, Jolly, and Titi (2015, https://doi.org/10.48550/arxiv.1410.176), temperature observations are not needed for the PI-DNN to predict the fine-scale velocity, pressure and temperature fields.

Plain Language Summary Detailed descriptions of the solution of dynamical systems are essential for a comprehensive understanding of their behavior. Obtaining sufficiently fine solutions is, however, computationally expensive. Mathematical tools have been developed to drive the solution of dynamical systems toward the reference solution, from which only coarse-scale observations are available. These techniques either rely on statistical correlations between the low-resolution observations and the higher-resolution reference solution, or alternatively on solving the governing equation on a high-resolution mesh with an additional forcing term that depends on the discrepancy between the observations and solution. Recently, a continuous map, called determining form, from the coarse-scale data to the high-resolution solution fields was proven to exist, subject to reasonable assumptions on the resolution of the coarse data. We propose a physics-informed deep neural network that serves as a surrogate of this theoretical map, to predict the higher-resolution solution fields from a history of coarse-scale data. The proposed network is shown to mimic the performance of dynamical downscaling, and is governed by similar constraints on the spatial and temporal resolutions of the observations.

1. Introduction

High-resolution simulations capture important fine-scale features and processes of a dynamical system, important for an enhanced understanding of the interweaving dynamics. High-resolution representations of a dynamical system are, however, limited by the numerical model’s resolution and available computational resources (Knutson et al., 2013). This problem is also challenging because observations are generally sparse and high-resolution simulations are computationally-demanding (Law et al., 2015). For instance, Chassignet and Xu (2021) report that the computational cost increases by a factor of 10 when the spatial resolution is doubled. In addition, environmental data are usually provided at a coarser resolution than required for a genuine analysis of the flow fields (Blair et al., 2019; Chassignet & Xu, 2017).

Mathematical approaches have been developed to drive the solution of a dynamical model toward the available coarse-scale information. Established approaches rely on data assimilation and/or downscaling techniques as...
means of using coarse information of the variables of a dynamical system to predict their higher-resolution counterparts (D. Chen et al., 2006). Downscaling methodologies are split into two categories: statistical and dynamical downscaling. Statistical downscaling extracts a statistical relation between the coarse information and their high-resolution counterpart (R. L. Wilby et al., 1998). These techniques were used, for instance, for downscaling regional climate (Lafalanne et al., 2016) and ocean wave fields (Camus et al., 2014). On the other hand, dynamical downscaling generates high-resolution fields by constraining a dynamical model with the coarse-scale information (R. Wilby & Wigley, 1997). Nudging techniques are popular dynamical downscaling methods, where the solution of a dynamical system is nudged toward the information, point-by-point in grid nudging and low-frequencies to low-frequencies in spectral nudging (Altaf et al., 2017). Nudging techniques are commonly utilized for predicting turbulent unsteady flows (Zauner et al., 2022), and downscaling atmospheric fields (Liu et al., 2012), to name a few.

Continuous Data Assimilation (CDA) was introduced as an alternative dynamical downscaling algorithm to construct an increasingly accurate description of the high-resolution solution of a dynamical system (Azouani et al., 2014). Specifically, CDA introduces a source term that continuously nudges the coarse-scales of the governing equations toward those of the coarse-scale information. The source term constrains the large features by comparing the coarse-data of the system state variables to the corresponding large-scales from the solution of the governing equations. CDA theorizes the existence of a map, called determining form map, from the coarse-scale data of the system state to the fine-scale counterpart (Biswas et al., 2019; Foias et al., 2012, 2014, 2017). CDA has been theoretically analyzed for a variety of dynamical systems including the 2D Rayleigh-Bénard (RB) convection (Farhat, Lunasin, & Titi, 2015; Farhat et al., 2017), surface quasi-geostrophic equations (Jolly et al., 2019), and 3D planetary geostrophic model (Farhat et al., 2016), to name a few. CDA has also been examined numerically in applications to downscale the 2D Navier-Stokes (Gesho et al., 2016), 2D RB convection (Altaf et al., 2017; Farhat et al., 2018), and recently Global Circulation Models (Desamsetti et al., 2019, 2022).

A surrogate model of the determining form map could serve as a viable solution to alleviate the computational burdens of dynamically downscaling coarse-scale data of the system state by solving the equations on high-resolution meshes. Deep neural networks are capable of learning complex relations between input and output fields (Goodfellow et al., 2016), and are also generally cheaper to evaluate, once trained (Gottwald & Reich, 2021). Training such networks has been made more efficient with the development of tools such as PyTorch (Paszke et al., 2019a, 2019b) and TensorFlow (Abadi et al., 2016; TensorFlow Developers, 2022), which harness the computational power of graphics processing units (GPUs) (Vuduc et al., 2010). This enabled deploying deep learning techniques in a variety of applications including autonomously navigating drones (Lee et al., 2021), drug discovery (K. Huang et al., 2021) and classifying crops from satellite images (Kussul et al., 2017). The predictions of a neural network can be further constrained by the underlying physical equations in the loss function (Bihlo & Popovych, 2022; Raissi et al., 2019). Specifically, the total loss function used to construct these so-called physics-informed networks can be designed to include the norm of the residual of the governing equations, initial and/or boundary conditions. An overview of various applications of physics-informed neural networks can be found in Cai et al. (2021) and Karniadakis et al. (2021). Recently, physics-informed neural networks were used in computational fluid dynamic applications (Jiang et al., 2020a; Wang et al., 2020), subgrid scale parameterization of chemically reactive flows (Bode et al., 2021), as well as weather and climate applications (Kashinath et al., 2021; Mooers et al., 2021).

Inspired by Azouani et al. (2014), Foias et al. (2014), and Jiang et al. (2020a), we propose a physics-informed deep neural network (PI-DNN), called CDAnet, that learns the determining form map that was theoretically described and established by Foias et al. (2012, 2014, 2017) for downscaling dynamical fluid flows. The PI-DNN outputs are constrained with the physics by following the approach introduced by Raissi et al. (2019). In addition, CDAnet is trained for the conditions set by Azouani et al. (2014) and Foias et al. (2014, 2017) for the existence of the determining form map, where downscaling is limited by the spatio-temporal resolution of the coarse-scale data. Some work has been carried out in the literature to address the presented issues. This is the first work that learns the determining form map as theorized and established by CDA. Specifically, purely data-driven approaches based on super-resolution were applied for enhancing the resolution of coarse information without constraining the solution by the physics (e.g., Vaughan et al., 2021). Similar work was introduced for data compression, where the training and validation of the network were conducted using the same data set, comprising a single or multiple solution trajectories (Jiang et al., 2020a). The proposed neural network was implemented to learn the CDA determining form map, from coarse-scale data of the RB convection model at a fixed Rayleigh (Ra) number.
to its high-resolution counterpart. We emphasize that our objective differs from that in Bode et al. (2021), who aimed at learning a subgrid-scale parametrization, and from that in Jiang et al. (2020a), who essentially focused on data compression. Specifically, our goal is to determine a neural network approximation of the determining form map, and assess the performance of this approximation in downscaling a history of coarse-scale observations that have not been considered in either training or in validation. CDAnet was trained to generalize different solution trajectories of the RB model by evaluating the trained model on solution trajectories that are different than that from training. In addition, being a surrogate model of this determining form map, the neural network is constrained by theoretical bounds on the extents at which downscaling could be performed, namely, the spatial and temporal resolutions of the observed data.

The performance of the proposed PI-DNN is assessed for RB flows at several Rayleigh numbers, and for different spatio-temporal resolutions of the coarse-scale information. The proposed network consists of a U-net to extract spatio-temporal features from the input (Ronneberger et al., 2015a) and a multi-layer perceptron (MLP) to predict the system states at continuous spatio-temporal coordinates. The network was trained by minimizing a compound loss function composed of a weighted sum of a regression loss and a partial differential equation (PDE) loss, which constrains the predicted solutions using the actual governing equations (Cai et al., 2021; Raissi et al., 2019).

PI-DNN models were trained to downscale the RB convection for a variety of spatio-temporal resolutions of the input. We rely on datasets corresponding to different Ra numbers to examine the impact of chaotic dynamics on downscaling and its limitations in terms of the spatial and temporal resolutions of the input. A well-trained model was also used to downscale RB flows at Ra numbers different than the one trained on to assess the robustness of the proposed model. The outputs of the proposed PI-DNN were compared to the dynamically downscaled solution by CDA for out-of-distribution (OOD) samples, which are samples not seen during training by evaluating a long-term history of coarse-scale data. The framework is also tested to downscale the RB convection without using temperature information as theoretically proven by Farhat, Jolly, and Titi (2015). Numerical experiments suggest that the trained model serves as a surrogate of the determining form map, where a well-trained model is able to mimic numerically downscaling using CDA with an additional error within 4%.

The study is organized as follows. Section 2 describes the RB convection, CDA downscaling, numerical discretization of the governing equations and the datasets used for training and validation. Section 3 presents the architecture of the PI-DNN along with the training procedure. The numerical experiments are outlined and accompanied with a thorough analysis in Section 4. Finally, Section 5 wraps up by drawing conclusions on the main results and presents possible future work.

2. Rayleigh-Bénard Convection

Consider a Newtonian fluid with kinematic viscosity $\nu$, thermal expansion coefficient $\alpha$ and thermal diffusivity $\kappa$ enclosed in a 2D periodic channel of horizontal period $L_x$ and vertical height $L_y$. Dimensional quantities are denoted using tildes. The flow is driven by buoyancy effects induced by a temperature gradient between a hot bottom plate and a cold top plate where the gravitational acceleration, $g$, points downwards (Pandey et al., 2018).

Applying the Boussinesq approximation, and the suitable scaling to the governing equations, the RB equations can be expressed as:

$$\nabla \cdot \mathbf{u} = 0,$$  \hspace{1cm} (1)

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \frac{\nu}{\Pr} \nabla^2 \mathbf{u} + \Pr \, \mathbf{e}_y,$$  \hspace{1cm} (2)

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \frac{1}{\Pr} \nabla^2 T + \mathbf{u} \cdot \mathbf{e}_z,$$  \hspace{1cm} (3)

where $Ra = (g \alpha \Delta \Theta L_x^3) (\kappa \nu)^{-1}$ is the Rayleigh number, $\Delta \Theta$ is the absolute temperature difference between the plates, $Pr = \nu \kappa / \alpha$ is the Prandtl number, $t$ is time and $\mathbf{e}_z$ is the unit vector in vertical direction. The governing equations are solved for the temperature $T$, pressure $p$, horizontal velocity $u$ and vertical velocity $v$. 

\hspace{0.5cm}
In the present work, the RB governing equations are solved in the 2D domain \( \Omega = [0, L_x] \times [0, 1] \), subject to horizontal periodic boundary conditions with period \( L_x \), no slip and fixed temperature conditions at the bottom and top boundaries, that is,

\[
\begin{align*}
\mathbf{u}(t; x, 0) &= \mathbf{u}(t; x, 1) = 0, \\
T(t; x, 0) &= T(t; x, 1) = 0.
\end{align*}
\]

The system is initialized using random velocity and temperature fields drawn from independent, spatially-uncorrelated, uniform distributions, specifically according to \( \mathbf{u}(0; x, y) \sim U(-0.1, 0.1), \ v(0; x, y) \sim U(-0.1, 0.1), \) and \( T(0; x, y) \sim U(-0.1, 0.1) \) for all \( (x, y) \in \Omega \). Here, \( U(a, b) \) refers to a uniform distribution on the interval \( (a, b) \). In order to generate a diverse data set, the system of equations was solved several times, each starting from a different realization of the initial conditions, leading to different trajectories of the system.

2.1. Continuous Data Assimilation

CDA is a dynamical downscaling algorithm that was introduced by Azouani et al. (2014). It consists in introducing a spatial feedback nudging term in the governing equations that depends on the difference between the large spatial scales of the high-resolution solution and their corresponding spatial scales from the coarse data of the system state. CDA was shown theoretically (Azouani et al., 2014; Farhat, Jolly, & Titi, 2015) and numerically (Altaf et al., 2017; Desanetti et al., 2019; Geso et al., 2016) to converge exponentially in time to the unknown reference solution, of which only sparse measurements are available. The solution converges when the nudging term is adequately scaled and sufficient observations, which are available in the form of coarse-scale information in this study, of the system states are provided. In this section, a brief outline of the CDA methodology for the RB system is presented.

Consider the spatial interpolation operator \( I_{h^o} \) of an appropriate interpolant function \( \phi \) operating on a uniform observation grid of size \( h^o \),

\[
I_{h^o}(\phi(x)) = \sum_{k=1}^{N_{h^o}} \phi(x_k) \chi_{Q_k}(x),
\]

where, \( x = (x, y) \), \( Q_k \) are disjoint subsets such that \( \text{diam}(Q_k) \leq h^o \) for \( k = 1, \ldots, N_{h^o} \), \( N_{h^o} \) is the number of observation points, \( \bigcup_{k=1}^{N_{h^o}} Q_k = \Omega, x_k \in Q_k \), and \( \chi_{Q} \) is the indicator function of the set \( S \). Let \( w = (\tilde{u}, \tilde{v}) \) denote the velocity vector of the CDA (downscaled) solution, \( \Psi \) denote the downscaled temperature, and \( p \) the pressure. For the RB system, the governing equations for the CDA-downscaled solution are expressed as:

\[
\begin{align*}
\nabla \cdot w &= 0, \\
\frac{\partial w}{\partial t} + (w \cdot \nabla) w + \nabla p &= \frac{Pr}{\sqrt{Ra}} \nabla^2 w + Pr \Theta e_y + \mu_u (I_h(u^o) - I_h(w))^o, \\
\frac{\partial \Psi}{\partial t} + (w \cdot \nabla) \Psi &= \frac{1}{\sqrt{Ra}} \nabla^2 \Psi + \mu_T (I_h(T^o) - I_h(\Psi))^o,
\end{align*}
\]

where, \( \mu_T \) and \( \mu_u \) are strictly positive constants called the nudging parameters and the superscript “\(^o\)" represents an observed quantity. In this study, the nudging factors were selected to be equal, that is, \( \mu_T = \mu_u = \mu \).

The CDA algorithm, which was first introduced for the Navier-Stokes equations as a paradigm, guarantees that the downscaled solution converges to the true solution regardless of the selected initial conditions when sufficiently enough observations of the system states are provided and the nudging factors are appropriately chosen (Azouani et al., 2014). It was also proven that there exists a unique map, called the determining form map, that maps the coarse observations to the high-resolution solution. Extending this work, Farhat, Jolly, and Titi (2015) theoretically demonstrated that when CDA is applied to the RB system, the downscaled solution converges to the reference solution regardless whether temperature observations are omitted or not. The goal of this work is to construct an approximate surrogate of the determining form map by suitably training a PI-DNN.
CDA was implemented to numerically downscale the RB system of equations (Altaf et al., 2017). Numerically, the downscaled solution was shown to converge to the reference solution exponentially in time provided sufficient observations of the system state that are sufficiently spaced apart both in space and time, and the nudging factor is adequately chosen. In addition, Altaf et al. (2017) reported that the CDA algorithm is not sensitive to the choice of the interpolation function where different interpolants resulted in a similar convergence rate and asymptotic errors. To this end, a limited number of interpolation methods were tested, and the nearest neighbor interpolation was selected due to its efficient computational properties.

2.2. Numerical Solution

The RB and corresponding CDA equations are solved using a finite difference methodology on a uniformly-spaced staggered Cartesian coordinate grid. The grid has \( n_x \) and \( n_y \) points in the x and y directions, respectively, yielding cells with \( \Delta x = L_x / n_x \) and height \( \Delta y = n_y / L_y \). The governing equations are spatially discretized using a second-order central differencing scheme. The time integration is conducted using the third-order Adam-Bashforth scheme with a fixed time step, \( \Delta t \). A pressure projection scheme (Chorin, 1968) is used to satisfy the continuity equation, and a fast Fourier transform algorithm is employed to solve the corresponding elliptic equation for pressure.

The RB equations are solved over a 2D periodic channel with a 3:1 aspect ratio, that is, \( L_x = 3 \) and \( L_y = 1 \). The fundamental domain \( \Omega \) is discretized using a computational grid with \( n_x = 768 \) and \( n_y = 256 \). Time marching was conducted using a time step \( \Delta t = 5 \times 10^{-4} \), and the simulations are performed up to a final time \( t_f = 60 \). We consider a fixed Prandtl number of 0.7, and three different Rayleigh numbers, \( Ra = 10^5, 10^6, \) and \( 10^7 \), allowing us to observe transitional and chaotic flow regimes (Lohse & Xia, 2010; Plumley et al., 2016). For all the selected parameters, the maximum grid Reynolds number is approximately \( \mathcal{O}(10) \), and the peak Courant-Frechet-Levy (CFL) number remains below 0.15. The numerical simulations are thus considered to be sufficiently resolved in space and time, and thus suitable for the purpose of training and validation.

2.3. Training and Validation Data Sets

The governing Equations 1–3 were solved 25 times for each of the aforementioned Ra numbers, each initialized by a unique random field. Obtaining multiple solution trajectories enriches the datasets with a variety of unique features, which helps the deep neural network to have better generalizability (Jan et al., 2019; Paschali et al., 2018). From the solution trajectory described in the previous section, the solution corresponding to \( t \in [25, 45] \) at time increments of \( \delta t \) were considered for training and validation. Specifically, \( \delta t \) was selected to be 0.1 for the flow with \( Ra = 10^5 \), and 0.05 for flows at \( Ra \) numbers of \( 10^6 \) and \( 10^7 \). This selection would first remove the initial transient corresponding to the noisy field and second leaves instances outside the training to examine the performance of the trained neural network at downscaling OOD samples. The training data set corresponding to each Ra number consists of 20 model runs while the validation set is constructed using the remaining five runs.

A supervised training approach is adopted for training the PI-DNN, which requires input-output pairs to do so. The training pairs consist of clips defined by a sequence of snapshots of the solution fields, where the input clip is composed of a sequence of coarse spatio-temporal snapshots of the solution fields, while the output clip corresponds to the high-resolution counterpart. To generate these training pairs, randomly sampled clips are first extracted from the available datasets by selecting crops of size \([n_{x,c}, n_{y,c}, n_t]\), where \( n_t \) is the clip length, \( n_{x,c} \), and \( n_{y,c} \) are the number of points in the crops along the horizontal and vertical dimensions, respectively. The sampled clips are then coarsened in both space and time to generate the input clips to the network. Denote by \( S \) the spatial downscaling factor, where the coarse input clip is constructed by choosing every \( S \) grid point in space. Similarly, denote by \( T \) the temporal downscaling factor where the coarse input clip is constructed by selecting one frame every \( T \) snapshots in time. Therefore, the sampled clips would consist of a low-resolution input that is \( S \) times smaller in space and \( T \) times smaller in time than the high-resolution truth, which is used to supervise the PI-DNN during training. The coarse input clip is interpolated spatially using the \( I_{S,T} \) function propagating it through the network, which is the first link to the CDA theory.

Remarks. As outlined above, our approach to training the neural network consists in drawing training data from different trajectories than those used to provide the validation data. This differs from the framework of Jiang et al. (2020a), who focused on data compression and relied on a single data set, comprising a single or several solution trajectories, for training and validation.
Finally, we emphasize that the low-resolution data considered as input to the neural network is synthesized so as to mimic the spatially-sparse (local) measurements typically encountered in environmental flow applications. This differs from the setting of some LES applications, where coarse (spatially-filtered) LES data is used as input (e.g., Bode et al. (2021)). As noted earlier, the use of local measurements simultaneously places us in the framework of observations assimilation and downscaling, and within the scope of a well-established theory (Azouani et al., 2014; Cao et al., 2022; Foias et al., 2012, 2014).

3. CDAnet

The PI-DNN, CDAnet, is composed of two-stages, a features extractor that precedes an MLP. The low-resolution input is first input to a 3D U-net that extracts spatio-temporal features yielding features vectors at each coarse-spatio-temporal grid point. The features vector is then combined with the spatio-temporal coordinates by concatenation, and input to the MLP to predict the high-resolution output and to compute the necessary partial derivatives needed to approximate the residual of the PDE. The features extractor is first described in Section 3.1 where the adopted architecture is explained. The details of the architecture of the MLP are then presented in Section 3.2. Figure 1 outlines the training pipeline accompanied by a descriptive illustration of the PI-DNN detailing the architecture of each of the two stages involved. The total loss, consisting of a linear combination of the regression and PDE loss terms, is discussed in Section 3.3.

3.1. Feature Extraction

The input to the PI-DNN is a low-resolution clip consisting of sparse snapshots of the evolution of the solution fields. First, the clips are propagated through a modified U-net, which is a convolutional neural network that is used to extract the spatial and temporal features of the input. Originally, the U-net was first employed for semantic segmentation of medical images (Ronneberger et al., 2015a, 2015b). The utility of this convolution network was later extended on to include computer vision tasks such as classification (Alom et al., 2019), human pose estimation (Newell et al., 2016) and image denoising (Komatsu & Gonsalves, 2020). For the application at hand, the input is 3-dimensional in \((x, y, t)\) with pressure, temperature and two velocity components treated as different channels. This requires modifying the original U-net that takes 2-dimensional, RGB or gray-scale, images as input by replacing the 2D convolution with their 3D counterparts, thus, extracting features in both space and time. In addition to the 3D convolution operations, we use Inception-ResNet blocks (Szegedy et al., 2017) to first improve the gradient flow for enhanced training properties (He et al., 2016) and second to have a larger receptive field (Szegedy et al., 2015). We tested Residual (He et al., 2016) and Dense connections (G. Huang et al., 2017), however, the Inception-ResNet blocks were found to provide the best compromise between memory requirement and model accuracy.

In similar fashion to the original U-net, the proposed feature extractor consists of an encoder-decoder setup, where the encoder is composed of several stages of convolutional Inception-ResNet blocks, succeeded by a max-pooling layer with a stride equal to 2. The Inception-ResNet blocks were designed to comprise of three convolution branches with kernel sizes of \([1 \times 1 \times 1],[5 \times 5 \times 3]\), and \([9 \times 9 \times 5]\), respectively. The outputs of the Inception-ResNet are first concatenated and then linked to the input of the convolution block via a Res-connection, as proposed by Szegedy et al. (2017). A batch normalization (BatchNorm) layer follows each convolution layer, which in turn is succeeded by a rectified linear activation unit activation function.

Zero-padding was adopted so that the input and output of each convolution layer maintain the same size spatially and temporally. The decoder is constructed to mirror the encoder with nearest neighbor upsampling replacing the max-pooling operations. Finally, the U-net skip connections link the features from each level of the encoder to its counterpart from the decoder. These skip connections were shown to improve training by promoting a stronger gradient flow and maintaining information from previous layers (He et al., 2016; Ronneberger et al., 2015a).

3.2. Multi-Layer Perceptron Network

The next stage of the proposed PI-DNN consists of an MLP, based on the architecture originally proposed by Z. Chen and Zhang (2019a, 2019b) for representation learning and shape generation. In this study, however, the features extracted by the 3D U-net are input along with the spatio-temporal coordinates to the network. Moreover,
Figure 1. Training pipeline: (a) Illustration of the training pipeline. CDAnet takes a low-resolution clip of the flow field and predicts the high-resolution counterpart. The low-resolution clip is first fed to a 3D U-net that outputs a latent grid of spatio-temporal features. The features are concatenated and passed to an MLP that predicts the high-resolution system states. The network is supervised by high-resolution ground truth clip, which is compared with the predicted clip via a regression loss $\mathcal{L}_{\text{reg}}$. A partial differential equation (PDE) loss $\mathcal{L}_{\text{PDE}}$ is used to enforce the physics described by the governing equations. (b) Illustration of the architecture of CDAnet, showing the 3D U-net, the Inception-ResNet blocks and the MLP.
this input vector is concatenated with each of the hidden layers of the network, maintaining the information throughout the network. The resulting MLP outputs the temperature, pressure and two velocity components corresponding to the input coordinates, thus yielding a “continuous” prediction of the high-resolution fields. Specifically, the MLP predicts the system state for the input spatio-temporal coordinates within the domain. The MLP predictions are performed at the spatio-temporal coordinates described by the reference solution grid used to supervise the training, where the predicted and reference solutions should lie on the same grid to compute the error metrics suitably.

The MLP facilitates constraining the predicted solution to a physically-consistent one by including of a soft constraint in the loss function. Considering the spatio-temporal coordinates as input to the MLP enables computing the partial derivatives of the four output variables with regards to the spatial and temporal coordinates. This methodology was first developed by Raissi et al. (2019) who proposed using backward propagation to approximate the partial derivatives. The physics of the system can then be used to constrain the network’s prediction by including the norm of the residual of the PDEs as an additional loss term in the total loss function. Several activation functions were examined, however the infinitely differentiable Softplus nonlinearity was chosen for the MLP because of its reasonable memory requirement and high prediction accuracy.

### 3.3. Loss Functions

To meet the objectives of downscaling while enforcing the dynamics, the PI-DNN is trained by minimizing a total loss function $\mathcal{L}$ consisting of a linear combination of the regression loss ($\mathcal{L}_{reg}$) and the PDE loss ($\mathcal{L}_{PDE}$). $\mathcal{L}_{reg}$ measures how far the prediction is from the truth, while $\mathcal{L}_{PDE}$ measures how well the prediction satisfies the physical constraints. The total loss function is expressed as:

$$\mathcal{L} = \mathcal{L}_{reg} + \lambda \mathcal{L}_{PDE}$$

where $\mathcal{L}_{reg} = \|y_{true} - \hat{y}\|_p + \lambda\|\Gamma \hat{y} - f\|_p$,

Here, $y_{true} = \{T, p, u, v\}$ is the true solution of the system, $\hat{y}$ is the network’s predicted solution, $\|\cdot\|_p$ denotes the $p$-norm, $\lambda$ is a positive constant scaling the PDE loss, and $\Gamma \hat{y} - f$ is the residual of the system of governing equations (Equations 1–3) with $\Gamma$ the discrete differential operator and $f$ the discretized forcing term.

Computing $\mathcal{L}_{PDE}$ at all spatio-temporal grid points of the high-resolution grid is computationally prohibitive because a computational graph needs to be constructed for each point. To alleviate these burdens, graphs are constructed for a limited number of points where $\mathcal{L}_{PDE}$ is evaluated. Specifically, a total of 3,000 points are selected from the predicted clip having a spatial distance of $h'$. We note that the value of $\lambda$, which reflects a trade-off between exact reconstruction of the training data and dynamical consistency, was observed to have a significant impact on network training and performance. In particular, the impact of $\lambda$ becomes more pronounced at larger Ra and for coarser resolution inputs. Specifically, a large value of $\lambda$ would lead to the model overfitting to noise, whereas a small value leads to poor overall reconstruction accuracy in comparison to a model trained with the tuned value of $\lambda$. It is noteworthy to mention that this experience is different than that of Jiang et al. (2020a), where the trade-off parameter did not noticeably impact the accuracy of the trained models. This may be due to the fact that in the present study the validation and evaluation rely on a large number of OOD samples, whereas Jiang et al. (2020a) rely on a single data set of finite-horizon trajectories.

### 3.4. Evaluation Metrics

The trained models are assessed based on their performance on evaluation metrics quantifying the disagreement between the true and predicted output fields. Specifically, these metrics include the $\ell_1$ error, $\ell_2$ error, and normalized $\ell_2$ error given by relative root-mean-squared error (RRMSE). In this manuscript, only results based on the RRMSE are presented since it was used to select the best model. The RRMSE is expressed as:

$$RSM SE_\varepsilon = \frac{\|g(x, y; t) - \hat{g}(x, y; t)\|_2^2}{\|g(x, y; t)\|_2^2}$$

(12)
where \( g \) is a generic true field, and \( \hat{g} \) is its predicted counterpart. The error metric is computed on the high-resolution grid, for each solution variable independently of the other variables. These metrics are averaged over \( \Omega \), the time span of the output and the evaluation batch.

### 3.5. Training Setup

CDAnet is trained for 100 epoch, each consisting of 3,000 randomly-sampled clips totaling to 300,000 instances for training. Models are trained for the three datasets described in Section 2.3 and for different combinations of \( S \) and \( T \). The input clips are made up of eight snapshots with a spatial dimension of \([256S^{-1}, 256S^{-1}]\). The predicted clips have a spatial dimension of \([256, 256]\) and a temporal dimension of \(8 \times T\) for memory-efficient training. After training, the model is evaluated on the entire domain, that is, output arrays of spatial size \([768, 256]\). Results in the following section are presented for evaluating the model on an entire history of coarse-scale data to yield forecasts of the high-resolution counterpart both in space and time.

A stochastic gradient descent (SGD) algorithm was selected for minimizing \( \mathcal{L} \). The SGD algorithm was tuned to have a momentum of 0.9 and weight decay of \(10^{-4}\). The \( \ell_1 \)-norm was used for both terms of \( \mathcal{L} \), however, the best model was selected based on the average RRMSE across all solution variables. The reconstruction loss was computed over the entire high-resolution grid, in contrast to the random sampling approach presented in meshfreeFlow-net (Jiang et al., 2020a, 2020b), designed to test recovery of a particular solution trajectory at arbitrary space-time coordinates. The PDE loss term was computed for 3,000 randomly selected spatio-temporal grid points along the high-resolution prediction, where the number of points is limited by the available GPU memory.

For each combination of \((S, T)\) and each data set, a parameter search over the learning rate (LR) and \(\lambda\) was conducted. Specifically, \(\lambda\) was varied logarithmically within \([0.001, 0.01, 0.1]\) and LR within \([0.01, 0.05, 0.1, 0.15, 0.2, 0.25]\) when constructing the surrogate of the determining form map. The values for \(\lambda\) were selected in this range to make sure that the two loss terms have a similar magnitude. A model was trained for each \(\lambda\) and LR combination mentioned and the RRMSEs corresponding to each of these trained models are analyzed. The value of \(\lambda\) could then be fine-tuned to obtain a trained model with optimal RRMSE values. The best model for which the results are shown corresponds to that with the lowest averaged RRMSE, where the averaging is performed over all the solution variables and 3,000 randomly selected clips drawn from the validation data set. During training, the LR was scaled by a factor of 0.1 if no new best model was achieved within 10 consecutive training epochs to enable taking finer optimization steps.

**Remark.** The results presented in the following section rely on an evaluation strategy having the following features. For each Rayleigh number considered, an evaluation data set consisting of one solution trajectory is used, and the entire history of coarse-scale information spanning the domain is lifted using the PI-DNN to obtain high-resolution fields. Note that the evaluation trajectory is independent of the data sets used in training and validation. This enables us to test the predictive capabilities of CDAnet, and provides a fair setting for contrasting its performance with that of CDA.

### 4. Results

CDAnet is systematically analyzed by conducting a number of numerical experiments to examine its performance in downscaling the RB flow at various Ra numbers, and for a number of \((S, T)\) combinations. First, the RRMSE of the best trained models are presented for different Ra numbers and combinations of \((S, T)\). The results of the trained PI-DNN are then compared to those dynamically downscaled using CDA. One of the trained model is then evaluated on datasets corresponding to flows with Ra numbers different than the one trained on. Finally, the framework was tested for downscaling coarse-scale velocity and pressure observations only, which required slightly modifying the network.

#### 4.1. Learning the Determining Form

The primary objective of this study is to construct a surrogate of the determining form map theorized in Cao et al. (2022) and Foias et al. (2012, 2014, 2017), which maps the coarse-scale observations of the system state to the higher-resolution counterpart. To this end, the proposed PI-DNN in Section 3 is trained using the datasets described in Section 2.3. Several networks are trained for different combinations of downscaling parameters...
(S, T). In particular, a grid search for the optimal (λ, LR) pair is conducted for each data set and for each (S, T). This enables a thorough analysis of the proposed methodology across the downscaling resolutions and different Ra numbers.

Table 1 outlines the best (λ, LR) combinations resulting from the parameter search, the average RRMSE corresponding to each solution variable, and the training and validation losses. The RRMSEs of the solution variables are below 5% for S = 16 and below 2.5% for values of S equal to 2, 4, 8, for all T. These RRMSE values scale linearly with T for a fixed value of S. For a fixed T, comparable RRMSE values are achieved for S values of 2 and 4. As S increases, the RRMSE corresponding to the temperature field roughly doubles as the value of S doubles to 8, and for S = 16 the trained model achieves an RRMSE of approximately 3% when T = 4.

Figure 2 illustrates the evolution of the RRMSE of CDAnet for RB flow with Ra = 10^3. Predictions of the high-resolution solution variables are performed using (S, T) = (4, 4) and for observations of the system state at δt = 0.1 starting from the initial random field and till t = 60. The figure shows that CDAnet is not able to predict the fine-scale solution during the initial transient of the flow field, which is not surprising because the simulations are initialized from a random state that does not lie on the attractor. However, once a fully-developed flow regime is reached, CDAnet is able to accurately predict the fine-scale solution. This illustrates the fact that, similar to CDA, CDA.net can be initialized from an arbitrary state and low prediction errors are achieved when the flow is developed. Note that the RRMSE of all the variables oscillates about an error level that corresponds to that listed in Table 1. In addition, the red dots resemble the RRMSE at the last frame of the predicted clip, which appear to be at the minimum RRMSE. This indicates that CDA.net is a surrogate of the determining form map because it was able to downscale coarse-scale observation that were not accessible during training.

Table 2 presents the error metrics along with the training and validation losses corresponding to the best trained networks for different combinations of (S, T) for the Ra = 10^6 data set. The trained models achieve RRMSE values below 3% for all the considered (S, T) pairs. Similarly to Ra = 10^3 case, the RRMSE increases linearly with T for a fixed value of S. For a fixed T, however, the RRMSE increases with S with no particular relation being observed. We note that beyond S = 8, the CDA downscaled fields diverge away from the reference solution because the spatial resolution is not within the conditions set by the theory. This was numerically validated by solving the CDA-based RB equations with coarse-scale observations with S = 8, where the RRMSE of the CDA solution fails to drop below 50%. Similarly, the lowest RRMSEs achieved by CDA.net for the same downscaling parameters were slightly above 15% for all solution variables.

Finally, Table 3 lists the optimal (λ, LR) pair corresponding to each (S, T) combination for downscaling the flow at Ra = 10^7 accompanied with the RRMSEs of the solution variables and the training and validation losses.
Figure 2. Evolution of the RRMSE of the predicted (a) temperature, (b) u-velocity, and (c) v-velocity fields for $(S, T) = (4, 4)$. The RRMSE of the predicted fields decreases as time progresses, reaching a mean error level that corresponds to the average RRMSE obtained during training. The insets provide enlarged views of RRMSE at large time. The blue curve represents the RRMSE at every predicted high-resolution frame, whereas the red circles represent the RRMSE of the last predicted frame in the evaluation clip.
Similarly to the Ra = 10^6 case, it is only feasible to downscale the coarse-scale observations using CDAnet and CDA for the spatial resolution presented, that is, S = 2. For larger S values, the spatial resolution is too coarse for the downscaled solution to converge to the reference solution. For this spatial resolution, the RRMSE of the temperature field roughly doubles as T is doubled from 2 to 4, and approximately triples when T doubles from 4 to 8. This indicates that for T ≥ 8, the temporal resolution is coarse, leading to a rapid increase in RRMSEs. Lower prediction RRMSEs might be achieved if additional data were employed during training, however, this is limited by the available memory (Soekhoe et al., 2016). A lower RRMSE may also be obtained by using a deeper and/or wider network with more features, however, a larger network is prone to overfitting (Srivastava et al., 2014) and is limited by the available computational resources (Thompson et al., 2020).

### 4.2. Comparison With CDA

The proposed network is trained as means to build a surrogate model of the determining form map inspired by Azouani et al. (2014) and theoretically outlined by Foias et al. (2014), Biswas et al. (2019), and Cao et al. (2022). Essentially, this surrogate model is expected to follow similar theoretical error bounds as the determining form map. First, the network was shown to converge with high-accuracy only when the dynamically downsampling via CDA does so. Specifically, only under the necessary conditions that are outlined in the theory on the spatial and temporal resolutions of the observation would the network be able to downscaled observations of the RB flow at similar Ra numbers. In addition, there is a particular difference between CDA and the surrogate model in the sense that CDA constructs an increasingly accurate high-resolution solution of the system state. The proposed network, however, is a surrogate of the determining form map, and is a map from the coarse observations to the fine solution (i.e., no time dependence to improve the estimate).

The performance of CDAnet is compared against that of dynamically downsampling using CDA in a number of ways. First, as indicated in Figure 2, the lowest RRMSE is generally achieved at the last frame of the predicted clip, which is similar to CDA in constructing an increasingly accurate representation of the reference solution. Moreover, as eluded to in Section 4.1, the constraints on the spatio-temporal resolution of the observations, for

### Table 2

| S   | T   | λ  | LR  | RRMSE_T | RRMSE_p | RRMSE_ϕ | RRMSE_V | Training loss | Validation loss |
|-----|-----|----|-----|---------|----------|----------|----------|---------------|-----------------|
| 2   | 2   | 0.01 | 0.2  | 0.0029  | 0.0029   | 0.0024   | 0.0021   | 0.0064        | 0.0017          |
| 2   | 4   | 0.001 | 0.2  | 0.0048  | 0.0032   | 0.0031   | 0.0039   | 0.0031        | 0.0024          |
| 2   | 8   | 0.001 | 0.15 | 0.0193  | 0.0124   | 0.0097   | 0.0137   | 0.0082        | 0.0085          |
| 4   | 2   | 0.01  | 0.25 | 0.0050  | 0.0024   | 0.0049   | 0.0030   | 0.0070        | 0.0022          |
| 4   | 4   | 0.01  | 0.2  | 0.0091  | 0.0049   | 0.0059   | 0.0067   | 0.0091        | 0.0041          |
| 4   | 8   | 0.001 | 0.2  | 0.0213  | 0.0131   | 0.0131   | 0.0156   | 0.0105        | 0.0098          |
| 8   | 2   | 0.001 | 0.1  | 0.0160  | 0.0043   | 0.0150   | 0.0063   | 0.0062        | 0.0052          |
| 8   | 4   | 0.01  | 0.1  | 0.0181  | 0.0057   | 0.0159   | 0.0083   | 0.0118        | 0.0063          |
| 8   | 8   | 0.001 | 0.1  | 0.0282  | 0.0128   | 0.0187   | 0.0151   | 0.0114        | 0.0108          |

**Note.** The PI-DNN is evaluated on a data set of RB convection at Ra = 10^6 and δt = 0.05. The RRMSE of the solution variables are reported along with the training and validation losses for different combinations of S and T.

### Table 3

| S   | T   | λ  | LR  | RRMSE_T | RRMSE_p | RRMSE_ϕ | RRMSE_V | Training loss | Validation loss |
|-----|-----|----|-----|---------|----------|----------|----------|---------------|-----------------|
| 2   | 2   | 0.001 | 0.25 | 0.0069  | 0.00323  | 0.0055   | 0.0024   | 0.0031        | 0.0027          |
| 2   | 4   | 0.001 | 0.15 | 0.0109  | 0.0040   | 0.0065   | 0.0038   | 0.0050        | 0.0038          |
| 2   | 8   | 0.001 | 0.25 | 0.0373  | 0.0140   | 0.0161   | 0.0127   | 0.0166        | 0.0132          |

**Note.** The PI-DNN is evaluated on a data set of RB convection at Ra = 10^6 and δt = 0.05. The RRMSE of the solution variables are reported along with the training and validation losses for different combinations of S and T.
a fixed Ra number, of the system state for the downscaled solution to accurately represent the reference solution are similar for both CDA and the surrogate CDAnet. Specifically, for observational data that is too coarse (i.e., beyond the conditions of the theory), the CDA-downscaled solution diverges from the reference solution. Similarly, for the same conditions, RRMSEs above 20% are obtained with a trained CDAnet.

Figure 3 illustrates plots of the evolution of the RRMSE of the temperature solution in time for different combinations of \( S, T \) selected snapshots of the temperature fields downscaled using CDA and CDAnet are contrasted in Figure 4. RRMSE curves obtained by dynamically downscaling coarse observations using CDA are compared to those obtained by evaluating the trained deep neural network. In particular, we notice that both CDA and CDAnet are oscillatory in time, however, the RRMSEs corresponding to CDA decrease with time while those corresponding to CDAnet oscillate with an almost uniform amplitude. In addition, the CDAnet RRMSE curves are almost always higher in value than the CDA curves, which is an expected behavior being a surrogate of the determining form map. The CDA RRMSE curves decrease exponentially in time asymptoting to zero error as opposed to the CDAnet RRMSEs, which remain about the RRMSEs indicated in Table 2.

**4.3. Generalizability to Flows at Different Ra**

In this section, we investigate whether a network trained at a specific Ra number can perform well at downscaling observations of the RB flow at a different Ra number. Such generalizability would be particularly beneficial as it would help avoid computational overhead required to perform independent training for different Ra values.

Another consideration is that in realistic settings, the “effective” Ra and Pr numbers might not be precisely known. We investigate these questions by analyzing the performance of CDAnet trained using the Ra = 10^6 data set and downscaling parameters \( S, T = (4, 4) \), and evaluating the trained model on datasets with the following Ra numbers \{10^5, 7 \times 10^5, 8 \times 10^5, 9 \times 10^5, 1.1 \times 10^6, 1.2 \times 10^6, 1.5 \times 10^6, 2 \times 10^6, 3 \times 10^6, 5 \times 10^6, 1 \times 10^7\}. Although this is a wide range of Ra numbers, we are particularly interested in the generalizability of a network trained at a given Ra number to neighboring Ra values. Note that the present approach differs from the experiment conducted by Jiang et al. (2020a) in which a single data set, comprising 10 solution trajectories having different
Figure 4. Snapshots of the temperature fields at (a) $t = 15$, (b) $t = 18.2$ and (c) $t = 21.5$. The input, ground truth, CDAnet prediction and CDA prediction are plotted for comparison. The results shown correspond to $Ra = 10^6$ and $(S, T) = (8, 8)$. 
Ra numbers, was used for training and validation. Specifically, in the present case, the trained model is evaluated on unseen flow regime, whereas no OOD samples were considered in the analysis of Jiang et al. (2020a).

Figure 5 shows curves of the evolution of the temperature RRMSE, for the aforementioned Ra numbers, in time. The figure portrays RRMSE curves that drop rapidly after the initial transient to an error plateau. The height of the RRMSE plateau is comparable to that obtained with training for all considered Ra numbers. The RRMSE plateau's height is slightly lower when the Ra number of the flow is smaller than the one used for training, but is noticeably higher when CDAnet is evaluated for flows at higher Ra numbers. This is not surprising as one would anticipate higher energy in the small scales as the Ra number increases. For all Ra numbers up to 2 × 10^6, the RRMSE curves are within 1% of the RRMSE corresponding to downscaling a flow with the same Ra number as the one trained on. As Ra increases beyond 2 × 10^6, the RRMSE of the downscaled fields increases, and the oscillations of the RRMSE curves increase as well. Nevertheless, for all cases, the RRMSE plateau's height remains below 4%, suggesting that networks trained at a fixed Ra value continue to perform well for neighboring Ra values, with only a slight increase in RRMSE values.

4.4. Downscaling Without Temperature Observations

CDA was proven to be able to downscale the RB convection even if temperature observations were not available (Farhat, Jolly, & Titi, 2015; Farhat, Lunasin, & Titi, 2015). To ensure convergence in this setting, the theory requires stricter conditions on the spatial resolution of the observations (i.e., finer observations) and a smaller nudging factor. This leads us to consider a more complex network with a larger learning capacity to determine the more sophisticated determining form map, namely, from coarse-partial state data to high-resolution full-state prediction. Specifically, we double the number of features extracted by the U-net and those input to the MLP to construct a surrogate model that achieves adequate error levels. This also amounts to doubling the number of neurons in the U-net, and adding a hidden layer to the MLP. The input to the U-net thus consists of a low-resolution 3D clip with three channels corresponding to the pressure and both velocity components.

Table 4 presents the RRMSE of the solution variables along with the training and validation losses for the best trained model with their respective optimal (c, LR) combinations. The table shows that for all (S, T) pairs, both c and LR are smaller than the case when all solution variables are observed at a coarse-scale. In addition, the results suggest that the RRMSE of the temperature field is approximately 2% larger than the case when the full state is observed.

Figure 6 illustrates snapshots of the input, true and predicted solution fields for the case of (S, T) = (16, 8). The snapshots indicate that the predicted solutions are qualitatively similar to the true solutions for all time steps. Figure 7 presents the evolution of the RRMSE of the temperature and velocity components in time. All three curves decrease as time evolves till reaching a plateau around which the error oscillates. The plots also indicate that although the training data set contained clips of the fields for t ∈ [20, 40], the trained PI-DNN was capable of generalizing to OOD clips during evaluation.

4.5. Downscaling Using Temperature Only: A Counter Example

In the literature, researchers have tried to examine the use of temperature information only to recover the full system state (Agasthya et al., 2022; Charney et al., 1969). This approach, however, is not supported by theory (Farhat, Jolly, & Titi, 2015), and computational evidence (Altaf et al., 2017) provides a concrete example showing divergence of the downscaled fields from the true system state.

In this section, we examine the behavior of CDA and CDAnet in this context, namely when only temperature data is available. Specifically, in the case of CDA, the source term is present in Equation (9) only, whereas for CDAnet,
the input to the PI-DNN consists of coarse temperature data only. In both cases, we are of course interested in evaluating the full system state on a fine resolution mesh.

To this end, we performed experiments at $Ra = 10^5$, with downscaling parameters $(S, T) = (4, 4)$. To analyze the results, we contrast downscaled results obtained with PI-DNN trained with (a) all state variables used as input, (b) velocity and pressure are used as input, and (c) only temperature is used as input. For CDA, we also consider three analogous scenarios, assimilating (a) both velocity and temperature observations, (b) velocity observations only, and (c) temperature observations alone.

Figure 8 shows the evolution of RRMSE for all six cases outlined above. As expected (Altaf et al., 2017; Farhat, Jolly, & Titi, 2015), the results indicate that CDA recovers the true system state exponentially in time, when coarse velocity and temperature observations are downscaled or when only velocity observations are available. On the other hand, when only temperature observations are available, at late stages the downscaled CDA solution diverges from the true system state. Similar experiences are observed for CDAnet. Specifically, in scenarios (i) and (ii), the RRMSE errors in CDA tend to 0, whereas in CDAnet the RRMSE oscillates around a small amplitude plateau. In scenario (iii), both predictions diverge at later stages, with comparable error values.

These experiences underscore the importance of performing downscaling within a well-defined framework, which guarantees convergence of the downscaling algorithm or alternatively guarantees the existence of a learnable mapping.

### 5. Discussion and Conclusions

We proposed a PI-DNN, named CDAnet, that represents a surrogate model for downscaling fluid flows based on the CDA methodology, which theorizes a map, called determining form map, from the coarse observations of the system states to the fine solution. CDAnet was trained to predict the high-resolution solution, in both space and time, of a fluid flow provided coarse observations of the system state. The proposed framework harnesses elements from super-resolution networks along with physics-informed techniques to learn the determining form map using the data and governing equations. We assess the performance of the proposed PI-DNN using a number of numerical experiments that compare the predictions of CDAnet to dynamically downscaled solutions using CDA. In particular, the network was first used to learn the determining form map for flows at different Rayleigh (Ra) numbers and for different spatial and temporal downscaling parameters $(S, T)$. The outputs from CDAnet are then compared to those obtained by numerically downscaling the CDA equations. The model was tested for generalizability for downscaling flows at a Ra number different that the one originally trained on. Finally, the
Figure 6. Qualitative results of CDAnet without temperature fields. Snapshots of the input, true and predicted solutions to CDAnet for (a) $t = 15$, (b) $t = 20$, and (c) $t = 25$. The low-resolution pressure and velocity fields are input to the network that predicts the temperature, pressure and velocities at a higher spatial and temporal resolutions. The results shown correspond to $Ra = 10^5$ and $(S, T) = (16, 8)$. 
Figure 7. Evolution of the RRMSE of the (a) temperature, (b) u-velocity and (c) v-velocity fields for CDAnet. The network was trained to downscale the Rayleigh-Bénard convection at $(S, T) = (4, 4)$ without temperature measurements in the input. The RRMSE curves decrease as time progresses, reaching an oscillatory mean state. The insets provide enlarged views of RRMSE at large time.
The study was concluded by testing the proposed methodology at downscaling the RB convection without using temperature observations as shown theoretically by Farhat, Jolly, and Titi (2015).

The RRMSE was adopted as an error metric to evaluate the performance of the PI-DNN during training based on which the best model was selected. For $Ra = 10^5$, the model was able to downscale coarse observations up to $S = 16$ with RRMSEs below 5%. Similarly, for $Ra = 10^6$ the RRMSE corresponding to each of the solution variables were below 3% and below 4% for the case of $Ra = 10^7$. Numerical experiments demonstrated that the trained CDAnet is able to downscale observations of the flow at time instances not included in the training data set. The evolution curves of the RRMSE all show that beyond the initial transient, the errors oscillate about a mean error level.

Figure 8. Evolution of the RRMSE of the downscaled (a) temperature, (b) $u$-velocity, and (c) $v$-velocity fields for both CDA and CDAnet. Downscaling was performed for RB flow at $Ra = 10^5$ with $(S, T) = (4, 4)$. Plotted are curves corresponding to three scenarios: (i) the full state is observed, (ii) temperature information is missing, and (iii) only temperature is observed. The insets provide enlarged views of RRMSE at large time.
The errors of the downscaled fields were larger in the case of CDAnet in comparison to numerically downscaling via CDA. CDAnet was shown to generalize to flows at Ra numbers different than the one trained on. Specifically, numerical experiments show that a network trained to downscale flows at a specific Ra number provides robust predictions for flows at neighboring Ra numbers, with only a marginal increase in RRMSE. Moreover, the framework was slightly modified to accommodate downsampling without coarse temperature data. In particular, the network was made deeper and wider to be able to learn this more complex mapping. The trained models were able to achieve RRMSE values below 5% for all solution variables and all combinations of \((S, T)\) that were considered. On the other hand, when only temperature data is assimilated or used as input, both CDA and CDAnet failed to recover the fine-scale truth. This highlights the need to perform assimilation or learning in a setting that can guarantee convergence of downscaling algorithms or the existence of mappings that can be approximated by neural networks.

The current PI-DNN was trained to generate a deterministic mapping of perfect low-resolution information onto high-resolution fields. In environmental flow applications, low-resolution fields are nowadays generally provided as ensembles of possible scenarios raising the question of handling these uncertain inputs. In such settings, one could pursue training of networks with random parameters (Gagne et al., 2020; Ravuri et al., 2021) or alternatively the possibility of using deterministic mappings for the purpose of lifting solution statistics. We plan to explore both avenues in future work.

In summary, the proposed framework offers a PI-DNN architecture and training pipeline to learn the determining form map of a dissipative dynamic system. Although currently tested for a 2D time-varying flow, this framework can be extended to more complex settings, such as 3D time-dependent flows or environmental flow-predictions based on general circulation models. Such extensions will be the focus of follow-on work.

**Data Availability Statement**

The codes for constructing the neural network alongside the data used for training, validating and evaluating the neural network are available at https://datadryad.org/stash/share/lgIRCRVUGetrJYTwdD3G0zdLpjU5dgO079DPwzewm4U.

**References**

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). TensorFlow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)* (pp. 265–283). Retrieved from https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

Agasthya, L., Leon, P. C. D., & Biferale, L. (2022). Reconstructing Rayleigh–Bénard flows out of temperature-only measurements using nudging. *Physics of Fluids, 34*(1), 015128. https://doi.org/10.1063/5.0079625

Alom, M. Z., Yakopcic, C., Hasan, M., Taha, T. M., & Asari, V. K. (2019). Recurrent residual U-Net for medical image segmentation. *Journal of Medical Imaging, 6*(1), 1–16. https://doi.org/10.1117/1.JMI.6.1.014006

Altuf, M. U., Titi, E. S., Gebrael, T., Knio, O. M., Zhao, L., McCabe, M. F., & Hotet, I. (2017). Downscaling the 2d bénard convection equations using continuous data assimilation. *Computational Geosciences, 21*(3), 393–410. https://doi.org/10.1007/s10596-017-9619-2

Arouani, A., Olson, E., & Titi, E. S. (2014). Continuous data assimilation using general interpolant observables. *Journal of Nonlinear Science, 24*(2), 277–304. https://doi.org/10.1007/s00332-013-9189-y

Bihlo, A., & Popovych, R. O. (2022). Physics-informed neural networks for the shallow-water equations on the sphere. *Journal of Computational Physics, 456*(C), 111024. https://doi.org/10.1016/j.jcp.2022.111024

Biswas, A., Foisas, C., Mondaini, C. F., & Titi, E. S. (2019). Downscaling data assimilation algorithm with applications to statistical solutions of the Navier–Stokes equations. *Annales de l’Institut Henri Poincare (C) Analyse Non Lineaire, 36*(2), 295–326. https://doi.org/10.1016/j.anihpc.2018.05.004

Blair, G. S., Henrys, P., Leeson, A., Watkins, J., Eastoe, E., Jarvis, S., & Young, P. J. (2019). Data science of the natural environment: A research roadmap. *Frontiers in Environmental Science, 7*. https://doi.org/10.3389/fenvs.2019.00121

Bode, M., Gauding, M., Lian, Z., Denker, D., Davidovic, M., Kleinheinz, K., et al. (2021). Using physics-informed enhanced super-resolution generative adversarial networks for subfilter modeling in turbulent reactive flows. *Proceedings of the Combustion Institute, 38*(2), 2617–2625. https://doi.org/10.1016/j.proci.2020.06.022

Cai, S., Miao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. (2021). Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica, 37*(12), 1727–1738. https://doi.org/10.1007/s10409-021-01148-1

Camus, P., Menéndez, M., Méndez, F. J., Izaguirre, C., Espejo, A., Cánovas, V., et al. (2014). A weather-type statistical downscaling framework for ocean wave climate. *Journal of Geophysical Research: Oceans, 119*(11), 7389–7405. https://doi.org/10.1002/2014JC010141

Cao, Y., Jolly, M., & Titi, E. (2022). Determining form for the 2d Rayleigh–Bénard problem. *Pure and Applied Functional Analysis, 7*(1), 99–132.

Charney, J., Halem, M., & Jastrow, R. (1969). Use of incomplete historical data to infer the present state of the atmosphere. *Journal of the Atmospheric Sciences, 26*(5), 1160–1163. https://doi.org/10.1175/1520-0469(1969)026<1160:uoiohd>2.0.co;2

Chassagne, E. P., & Xu, X. (2017). Impact of horizontal resolution (1/12° to 1/50°) on Gulf Stream separation, penetration, and variability. *Journal of Physical Oceanography, 47*(8), 1999–2021. https://doi.org/10.1175/jpo-d-17-0031.1
Lee, T., Mckeever, S., & Courtney, J. (2021). Flying free: A research overview of deep learning in drone navigation autonomy. *Drones, 5*(2), 52. https://doi.org/10.3390/drones5020052

Liu, P., Tsipmidi, A. P., Hu, Y., Stone, B., Russell, A. G., & Nenes, A. (2012). Differences between downscaling with spectral and grid nudging using WRF. *Atmospheric Chemistry and Physics, 12*(8), 3601–3610. https://doi.org/10.5194/acp-12-3601-2012

Lobe, D., & Xia, K.-Q. (2010). Small-scale properties of turbulent Rayleigh–Bénard convection. *Annual Review of Fluid Mechanics, 42*(1), 335–364. https://doi.org/10.1146/annurev.fluid.010908.165152

Mooers, K., Pritchard, M., Beucker, T., Ott, J., Yacalis, G., Baldi, P., & Gentine, P. (2021). Assessing the potential of deep learning for emulating cloud superparameterization in climate models with real-geography boundary conditions. *Journal of Advances in Modeling Earth Systems, 13*(5). https://doi.org/10.1029/2020MS002385

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer vision – ECCV 2016* (pp. 483–499). Springer International Publishing. https://doi.org/10.1007/978-3-319-46484-8_29

Pandey, A., Scheel, J. D., & Schumacher, J. (2018). Turbulent superstructures in Rayleigh–Bénard convection. *Nature Communications, 9*(1), 2118. https://doi.org/10.1038/s41467-018-04478-0

Paschali, M., Conjeti, S., Navarro, F., & Navab, N. (2018). Generalizability vs. robustness: Investigating medical imaging networks using adversarial examples. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, & G. Fichtinger (Eds.), *Medical image computing and computer assisted intervention – MICCAI 2018* (pp. 493–501). Springer International Publishing. https://doi.org/10.1007/978-3-030-09928-1_56

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019a). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 32). Curran Associates, Inc. Retrieved from https://proceedings.neurips.cc/paper/2019/file/2a07a37596567fa9f701277740-Paper.pdf

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., et al. (2019b). PyTorch: An imperative style, high-performance deep learning library [Software]. GitHub. Retrieved from https://github.com/pytorch/pytorch/

Plumley, M., Julien, K., Marti, P., & Stellmach, S. (2016). The effects of Ekman pumping on quasi-geostrophic Rayleigh–Bénard convection. *Journal of Fluid Mechanics, 803*, 51–71. https://doi.org/10.1017/jfm.2016.452

Raissi, M., Perdikaris, P., & Karniadakis, G. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics, 378*, 686–707. https://doi.org/10.1016/j.jcp.2018.10.045

Ravuri, S., Lenc, K., Willson, M., Kangin, D., Lam, R., Mirowski, P., et al. (2021). Skilful precipitation nowcasting using deep generative models of radar. *Nature, 597*(7878), 672–677. https://doi.org/10.1038/s41586-021-03854-z

Ronneberger, O., Fischer, P., & Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), *Medical image computing and computer-assisted intervention – MICCAI 2015* (pp. 234–241). Springer International Publishing. https://doi.org/10.1007/978-3-319-24574-4_28

Ronneberger, O., Fischer, P., & Brox, T. (2015b). U-net: Convolutional networks for biomedical image segmentation [Software]. Vision. Retrieved from https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

Soekhoo, D., van der Putten, P., & Pfaaf, A. (2016). On the impact of data set size in transfer learning using deep neural networks. In H. Boström, A. Knobbe, C. Soares, & P. Papapetrou (Eds.), *Advances in intelligent data analysis xv* (pp. 50–60). Springer International Publishing. https://doi.org/10.1007/978-3-319-46349-0_5

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research, 15*(56), 1929–1958. Retrieved from http://jmlr.org/papers/v15/srivastava14a.html

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In Proceedings of the thirty-first AAAI conference on artificial intelligence (pp. 4278–4284). AAAI Press. https://doi.org/10.1609/aaai.v31i1.11231

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2015). Going deeper with convolutions. In *2015 IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–9). https://doi.org/10.1109/CVPR.2015.729594

TensorFlow Developers. (2022). TensorFlow [Software]. Zenodo. https://doi.org/10.5281/zenodo.6574269

Thompson, N. C., Greenewald, K. H., Lee, K., & Manso, G. F. (2020). The computational limits of deep learning. *CoRR, abs/2007.05558*. Retrieved from https://arxiv.org/abs/2007.05558

Vaughan, A., Tebbit, W., Hosking, J. S., & Turner, R. E. (2021). Convolutional conditional neural processes for local climate downscaling. *CoRR, abs/2101.07950*. Retrieved from https://arxiv.org/abs/2101.07950

Vuluc, R., Chandramowlishwaran, A., Choi, J., Guney, M., & Shringarpure, A. (2010). On the limits of GPU acceleration. In Proceedings of the 2nd USENIX conference on hot topics in parallelism (p. 13). USENIX Association. Retrieved from https://dl.acm.org/doi/10.5555/1863086.1863099

Wang, R., Kashinath, K., Mustafa, M., Albert, A., & Yu, R. (2020). Towards physics-informed deep learning for turbulent flow prediction. In Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining (pp. 1457–1466). Association for Computing Machinery. https://doi.org/10.1145/3394486.3403198

Wilby, R., & Wigley, T. (1997). Downscaling general circulation model output: A review of methods and limitations. *Progress in Physical Geography: Earth and Environment, 21*(4), 530–548. https://doi.org/10.1177/030913339702100403

Wilby, R. L., Wigley, T. M. L., Conway, D., Jones, P. D., Hewitson, B. C., Main, J., & Wilks, D. S. (1998). Statistical downscaling of general circulation model output: A comparison of methods. *Water Resources Research, 34*(11), 2995–3008. https://doi.org/10.1029/98WR02577

Zauner, M., Mons, V., Marquet, O., & Leclaire, B. (2022). Nudging-based data assimilation of the turbulent flow around a square cylinder. *Journal of Fluid Mechanics, 937*, A38. https://doi.org/10.1017/jfm.2022.133

HAMMOUD ET AL.