Reconstructing Turbulent Flows Using Spatio-temporal Physical Dynamics

SHENGYU CHEN, University of Pittsburgh, United States
TIANSHU BAO, Vanderbilt University, United States
PEYMAN GIVI, CAN ZHENG, and XIAOWEI JIA, University of Pittsburgh, United States

Accurate simulation of turbulent flows is of crucial importance in many branches of science and engineering. Direct numerical simulation (DNS) provides the highest fidelity means of capturing all intricate physics of turbulent transport. However, the method is computationally expensive because of the wide range of turbulence scales that must be accounted for in such simulations. Large eddy simulation (LES) provides an alternative. In such simulations, the large scales of the flow are resolved, and the effects of small scales are modelled. Reconstruction of the DNS field from the low-resolution LES is needed for a wide variety of applications. Thus the construction of super-resolution methodologies that can provide this reconstruction has become an area of active research. In this work, a new physics-guided neural network is developed for such a reconstruction. The method leverages the partial differential equation that underlies the flow dynamics in the design of spatio-temporal model architecture. A degradation-based refinement method is also developed to enforce physical constraints and to further reduce the accumulated reconstruction errors over long periods. Detailed DNS data on two turbulent flow configurations are used to assess the performance of the model.

CCS Concepts: • **Computing methodologies → Neural networks;**

Additional Key Words and Phrases: Physics-guided neural network, turbulent flow

1 INTRODUCTION

Direct numerical simulation (DNS) of the Navier–Stokes equations is a brute-force computational method and is the method with the highest reliability for capturing turbulence dynamics [21]. The computational cost of such simulations is very expensive for flows with high Reynolds numbers. **Large eddy simulation (LES)** is a popular alternative, concentrating on the larger scale energy-containing eddies and filtering the small scales of transport [44]. In this way, LES can be conducted on coarser grids as compared to DNS, but obviously with less fidelity [40].

Machine learning, including **super-resolution (SR)** methods [41], have been advocated as a means of reconstructing highly resolved DNS from LES data. These methods have shown tremendous success in reconstructing high-resolution data in various commercial applications. The majority of current SR models use **convolutional network layers (CNNs)** [2] to extract representative spatial features and transform them through complex non-linear mappings to recover high-resolution images. Starting from the end-to-end convolutional SRCNN model [15], several investigators have explored the addition of other structural components such as skip-connections [1, 12, 47, 47, 61, 62], channel attention [61], adding adversarial training objectives [10, 11, 28, 32, 46, 50, 51, 53], and more recently, Transformer [42]-based SR methods [17, 18, 34, 37, 49, 59, 63].

Given their success in computer vision, SR methods are becoming increasingly popular in turbulence reconstruction [14, 19, 20, 35, 51, 57]. Despite their popularity, these methods face some limitations when it comes to representing continuous flow dynamics in the spatial and temporal fields using discrete data samples. Consequently, they can learn spurious patterns between sparse observations, which often lack generalizability. Additionally, the training of SR models is hindered by the scarcity of high-fidelity DNS data due to the required high computational cost of such simulations.

In this work, a novel method termed the **"continuous networks using differential equations" (CNDE)** is developed to improve the SR reconstruction. This development is by leveraging the underlying physical relationships to guide the learning of generalizable spatial and temporal patterns in the reconstruction process. The method consists of three components: the **Runge–Kutta transition unit (RKTU)**, the **temporally enhancing layer (TEL)**, and degradation-based refinement. The RKTU structure is designed based on the governing **partial differential equations (PDEs)** and is used for capturing continuous spatial and temporal dynamics of turbulent flows. The TEL structure is designed based on the **long-short-term memory (LSTM)** [24] model and is responsible for capturing long-term temporal dependencies. The degradation-based refinement is to adjust the reconstructed data by enforcing consistency with physical constraints.

Model appraisal is made by considering detailed datasets pertaining to two turbulent flow configurations: (1) a **forced isotropic turbulent (FIT)** flow [38] and (2) the **Taylor–Green vortex (TGV)** flow [5]. The results of the consistency assessments demonstrate the capability of the CNDE in terms of the reconstruction performance over space and time. The effectiveness of each component of the methodology is demonstrated qualitatively and quantitatively.

2 RELATED WORK

2.1 Super-Resolution

Single image super-resolution (SISR) via deep learning has been the subject of many investigations in computer vision. These methods derive their power primarily from the utilization of convolutional network layers [2], which extract spatial texture features and transform them through complex non-linear mappings to recover high-resolution data. One of the earliest SR methods for SISR is SRCNN [15], which learns an end-to-end mapping between coarse-resolution and high-resolution images by employing a series of convolutional layers. Another scheme is the skip-connection layers [1, 12, 45, 47, 62], which enable the bypassing of abundant low-frequency information and emphasize the relevant information to improve the stability of the optimization process in deep neural networks. Several investigators have explored the adversarial training objective by using the **generative adversarial network (GAN)** for SISR. For example, the SRGAN model [32] stacks the deep residual network to build a deeper generative network for image super-resolution and also introduces a discriminator network to distinguish between reconstructed images and real

images using an adversarial loss function. The ultimate goal is to train the generative network in a way that the reconstructed images cannot be easily distinguished by the discriminator. One major advantage of SRGAN is that the discriminator can help extract representative features from high-resolution data and enforce such features in the reconstructed images. Several variants of SRGAN are given in References [10, 11, 28, 46, 50, 51, 53].

The Transformer [48] has revolutionized **natural language processing (NLP)** by introducing self-attention mechanisms, allowing it to efficiently process long-range dependencies in the sequences of data. This method can effectively capture contextual information from the entire input sequence, leading to significant advancements in various NLP tasks like machine translation, sentiment analysis, and so on. The Transformer has also been introduced into the SISR problem [17, 18, 34, 37, 42, 49, 59, 63]. For example, Yang et al. [59] developed the TTSR model, which uses a learnable texture extractor to extract textures from low-resolution images and reference **high-resolution (HR)** images to recover target HR images. Lu et al. [37] developed the ESRT model, which optimizes the original Transformer to achieve competitive reconstruction performance with low computational cost.

2.2 Super-Resolution for Turbulent Flows

There is a significant interest in developing SR techniques for high-resolution flow reconstructions. Fukami et al. [19, 20, 35] created an improved CNN-based hybrid DSC/MS model to explore multiple scales of turbulence and capture the spatio-temporal turbulence dynamics. Liu et al. [35] developed another CNN-based model MTPC to simultaneously include spatial and temporal information to fully capture features in different time ranges. Xie et al. [57] introduced tempoGAN, which augments a GAN model with an additional discriminator network along with new loss functions that preserve temporal coherence in the generation of physics-based simulations of fluid flow. Deng et al. [14] demonstrated that both SRGAN and ESRGAN [51] can produce good reconstructions. Yang et al. [60] created an FSR model based on a back-projection network to achieve three-dimensional (3D) reconstruction. Xu et al. [58] introduced a Transformer-based SR method to build the SRTT model for capturing small-scale details of turbulent flow.

2.3 Physics-guided Machine Learning

Recent studies have shown promise in integrating physics into machine learning models for improved predictive performance [56]. These methods typically enforce physics in the loss function [8, 13, 22, 23, 26, 43] or use simulated data for pre-training and augmentation [7, 9, 23, 25, 36]. Hanson et al. [22] introduced ecological principles as physical constraints into the loss function to improve the lake surface water phosphorus prediction. Karpatne et al. [13] developed a hybrid machine learning and physics model to guarantee that the density of water at a lower depth is always greater than the density at any depth above. Jia et al. [26] and Read et al. [43] extended this idea by including an additional penalty for the violation of the energy conservation law. In the flow data reconstruction, Chen et al. [8] constructed a PGSRN method to enforce zero divergences of the velocity field in incompressible flows. Despite the promise of these methods, they may lead to slow convergence in optimization and performance degradation, especially when the physical relationships are complex or have uncertain parameters.

A means of imposing the physics is by considering the PDEs that govern the physical phenomena. In some cases, however, direct integration of the governing PDEs using standard numerical methods [54] can become prohibitively expensive. An alternative is to solve PDEs via neural operators [4, 16, 33]. For example, Li et al. [33] introduced the **Fourier neural operator (FNO)** to model PDEs for learning the mappings between infinite-dimensional spaces of functions using the integral operator. The integral operator of this approach is restricted to convolution and
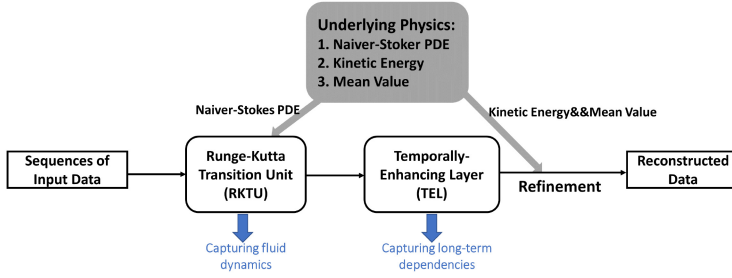
Fig. 1. The overall structure of the CNDE method.

instantiated through a linear transformation in the Fourier domain. However, the major limitation of neural operators for flow data reconstruction lies in their lack of explicit knowledge about the specific form of the underlying PDE (Naiver–Stokes equation). Neural operators directly learn the relation between input data and outputs without incorporating the intrinsic structure and physics encoded in the PDE. This can lead to inefficiencies and challenges in effectively capturing complex flow dynamics. An alternative direction is to embed the physics equations or relationships in the modeling structure [3, 29, 39]. One such example is the encoding of the Navier–Stokes equation in a recurrent unit, as demonstrated in our previous work [3]. However, this method may accumulate errors in long-term predictions, and it does not consider the use of LES data in reconstructing DNS data within the recurrent unit.

3 PROBLEM UNDER INVESTIGATION

In this work, the transport of unsteady, three-dimensional turbulent flows is the subject of main consideration. In all cases, the flow is assumed to be Newtonian and incompressible with a constant density. In the formulation, the space coordinate is identified by the vector $\mathbf{x} \equiv x,\ y,\ z$, and the time is denoted by t. The velocity field is denoted by $\mathbf{V}(\mathbf{x}, t)$, with its three components, $u(\mathbf{x}, t),\ v(\mathbf{x}, t),\ w(\mathbf{x}, t)$, along the three flow directions $x,\ y,\ z$, respectively. The pressure, the density, and the dynamic viscosity are denoted by $p(\mathbf{x}, t)$, $\rho(\mathbf{x}, t)$, and v, respectively. The latter two are assumed constant. The (dummy) parameters $\mathbf{Q}(\mathbf{x}, t)$ (as a vector), and/or Q (as a scalar) are used to denote a transport variable.

All of the flows considered are statistically homogeneous. High-resolution DNS and lower-resolution LES data are considered on $N_x \times N_y \times N_z$, and $M_x \times M_y \times M_z$ grid points, respectively. A box filter [27] is employed to create the LES data from the original DNS. All of the statistical averages, including the Reynolds-averaged values are obtained by data ensembled over the entire domain. In this way, the ensemble averages, denoted by an over-bar are defined by

$$\overline{\mathbf{Q(t)}} = \frac{1}{N_x \times N_y \times N_z} \sum_{i}^{N_x} \sum_{j}^{N_y} \sum_{k}^{N_z} \mathbf{Q}(i, j, k, t), \tag{1}$$

suitable for homogeneous flows. In the training process, the available DNS data are at a regular time interval δ, as $\mathbf{Q}^d = \{\mathbf{Q}^d(t)\}$ within the time $\{t_0, t_0 + \delta, \ldots, t_0 + K\delta\}$. The objective is to predict high-resolution DNS data after the historical data, at time $\{t_0 + (K+1)\delta, \ldots, t_0 + M\delta\}$. The variable $\mathbf{Q}^l(x, y, z, t)$ represents the low-resolution LES data at timestep t. Since the LES data can be created at a lower computational cost, they are used for both training and testing periods and at a higher frequency. The variable $\mathbf{Q}^l = \{\mathbf{Q}^l(t)\}$ denotes LES data within the time range $[t_0, t_0 + M\delta]$.

The CNDE framework consists of two structural components: the RKTU, and the TEL. The training is done in two phases: supervised super-resolution training, and degradation-based refinement. These are shown in Figure 1 and are described in order below.
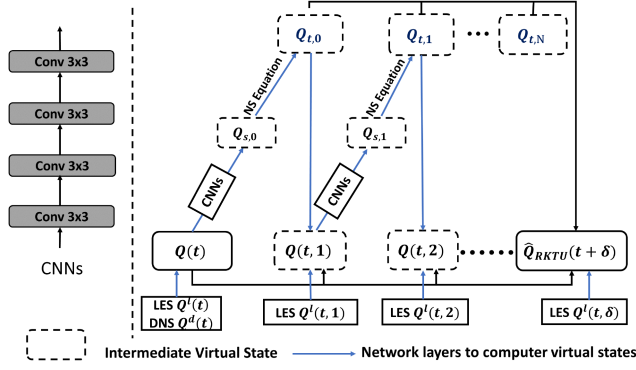
Fig. 2. The RKTU based on Naiver–Stokes equation for reconstructing turbulent flow data in the spatio-temporal field. $\mathbf{Q}_{s,n}$ and $\mathbf{Q}_{t,n}$ denote the spatial and temporal derivatives, respectively, at each intermediate timestep.

3.1 Runge–Kutta Transition Unit

The datasets $\mathbf{Q}$ pertaining to turbulent flows consist of the transport variables that interact with each other and evolve temporally and spatially. The traditional temporal models, e.g., LSTM [24], rely on large and consecutive training samples to capture the underlying patterns over time. However, the amount of high-fidelity DNS data is often limited. The RKTU structure is developed for reconstructing flow variables over a long period, given an initial DNS sample $\mathbf{Q}^d$ at t, and frequent low-resolution LES data samples $\mathbf{Q}^l$. The prediction follows an auto-regressive process in which the predicted DNS $\hat{\mathbf{Q}}^d(x, y, z, t)$ at time t, and frequent LES data $\mathbf{Q}^l$ from the current time to the next interval $[t, t + \delta]$ are used to predict the DNS at next timestep $\hat{\mathbf{Q}}^d(x, y, z, t + \delta)$.

The RKTU is based on the **RungeâĂŞKutta (RK)** discretization method [6]. The principal idea is to leverage the continuous physical relationship described by the underlying PDE to bridge the gap between the discrete data samples and the continuous flow dynamics. The scheme can be applied to any dynamical systems governed by deterministic PDEs. Consider the PDE of the target variables $\mathbf{Q}$ as expressed by

$$\mathbf{Q}_t = \mathbf{f}(t, \mathbf{Q}; \theta), \tag{2}$$

where $\mathbf{Q}_t$ denotes the temporal derivative of $\mathbf{Q}$, and $\mathbf{f}(t, \mathbf{Q}; \theta)$ is a non-linear function (parameterized by coefficient θ) that summarizes the current value of $\mathbf{Q}$ and its spatial variations. The turbulence data follow the Navier–Stokes equation for an incompressible flow. Thus, for $\mathbf{Q} \equiv \mathbf{V}(\mathbf{x}, t)$,

$$\nabla \cdot \mathbf{Q} = 0,$$
$$\mathbf{f}(\mathbf{Q}) = -\frac{1}{\rho}\nabla p + \nu\Delta\mathbf{Q} - (\mathbf{Q} \cdot \nabla)\mathbf{Q}. \tag{3}$$

The term ∇ denotes the gradient operator and $\Delta = \nabla \cdot \nabla$ on each of the components of the velocity. The independent variable t is omitted in the function $\mathbf{f}(\cdot)$, because $\mathbf{f}(\mathbf{Q})$ in the Navier–Stokes equation is for a specific time t (same with t in $\mathbf{Q}_t$). Figure 2 shows the overall structure of the method and involves a series of intermediate states $\{\mathbf{Q}(t, 0), \mathbf{Q}(t, 1), \mathbf{Q}(t, 2), \ldots, \mathbf{Q}(t, N)\}$. The temporal gradients are estimated at these states $\{\mathbf{Q}_{t,0}, \mathbf{Q}_{t,1}, \mathbf{Q}_{t,2}, \ldots, \mathbf{Q}_{t,N}\}$. Starting from $\mathbf{Q}(t, 0) = \mathbf{Q}(t)$, the RKTU estimates the temporal gradient as $\mathbf{Q}_{t,0}$ and then moves $\mathbf{Q}(t)$ toward the gradient direction to create the next intermediate state $\mathbf{Q}(t, 1)$. The process is repeated for N intermediate states. For the fourth-order RK method, as employed here, $N = 3$.

For the starting data point $\mathbf{Q}(t)$, an augmentation mechanism is adopted by combining the DNS and LES data, $\mathbf{Q}(t) = W^d\mathbf{Q}^d(t) + W^l\mathbf{Q}^l(t)$, where W^d and W^l are trainable model parameters, and $\mathbf{Q}^l(t)$ is the up-sampled LES data with the same resolution as DNS. The RKTU estimates the first temporal gradient $\mathbf{Q}_{t,0} = \mathbf{f}(\mathbf{Q}(t))$ using the Navier–Stokes equation and computes the next intermediate state variable $\mathbf{Q}(t,1)$ by moving the flow data $\mathbf{Q}(t)$ along the direction of temporal derivatives. Given frequent LES data, the intermediate states $\mathbf{Q}(t,n)$ are also augmented by using LES data $\mathbf{Q}^l(t,n)$, as $\mathbf{Q}(t,n) = W^d\mathbf{Q}(t,n) + W^l\mathbf{Q}^l(t,n)$, and they follow the same process to move $\mathbf{Q}(t)$ along the estimated gradient $\mathbf{Q}_{t,n}$ to compute the next intermediate states $\mathbf{Q}(t,n+1)$,

$$
\begin{aligned}
\mathbf{Q}(t,1) &= \mathbf{Q}(t) + \delta\frac{\mathbf{Q}_{t,0}}{2}, \\
\mathbf{Q}(t,2) &= \mathbf{Q}(t) + \delta\frac{\mathbf{Q}_{t,1}}{2}, \\
\mathbf{Q}(t,3) &= \mathbf{Q}(t) + \delta\mathbf{Q}_{t,2}.
\end{aligned}
\tag{4}
$$

The temporal derivative $\mathbf{Q}_{t,3}$ is then computed from the last intermediate state by $\mathbf{f}(\mathbf{Q}(t,3))$. According to Equation (4), the intermediate LES data $\mathbf{Q}^l(t,n)$ are selected as $\mathbf{Q}^l(t,1) = \mathbf{Q}^l(t+\delta/2)$, $\mathbf{Q}^l(t,2) = \mathbf{Q}^l(t+\delta/2)$, and $\mathbf{Q}^l(t,3) = \mathbf{Q}^l(t+\delta)$. Finally, RKTU combines all the intermediate temporal derivatives as a composite gradient to calculate the final prediction of next step flow data $\hat{\mathbf{Q}}_{\text{RKTU}}(t+\delta)$,

$$
\hat{\mathbf{Q}}_{\text{RKTU}}(t+\delta) = \mathbf{Q}(t) + \sum_{n=0}^{N} w_n\mathbf{Q}_{t,n},
\tag{5}
$$

where $\{w_n\}_{n=1}^{N}$ are the trainable model parameters.

The RKTU requires the temporal derivatives in the Navier–Stokes equation. The RKTU estimates the temporal derivatives through the function $\mathbf{f}(\cdot)$. According to Equation (3), the evaluation of $\mathbf{f}(\cdot)$ requires explicitly estimation of the first-order and second-order spatial derivatives. One of the most popular approaches for evaluating spatial derivatives is through **finite difference methods (FDMs)** [54]. However, the discretization in FDMs can cause larger errors for locations with complex dynamics. The RKTU structure, as depicted in Figure (2), utilizes CNN layers to replace the FDMs. The CNNs have the inherent capability to learn additional non-linear relationships from data and capture the spatial derivatives required in the Navier–Stokes equation. After estimating the first-order and second-order spatial derivatives, they are used in Equation (3) to obtain the temporal derivative $\mathbf{Q}_{t,n}$.

The padding strategies for CNNs also need to be considered. Standard padding strategies (e.g., zero padding) do not satisfy the spatial boundary conditions of the flows considered here. These conditions describe how the flow data interact with the external environment. With the assumption of homogeneous turbulence, periodic boundary conditions are imposed on all three flow directions. Thus, periodic data augmentation is made for each of the six faces (of the 3D cubic data) with an additional two layers of data before feeding it to the model.

3.2 Temporally Enhancing Layer

The RKTU can capture the data in the spatial and temporal field between a pair of consecutive data points, but it may cause large reconstruction errors in the long-time prediction if the time interval δ is large. Temporal models, such as LSTM [24], and **temporal convolutional network (TCN)** [31] are widely used to capture the long-term dependencies in time-series prediction. In this case, the LSTM model is incorporated in a TEL to further enhance the RKTU to capture long-term temporal dependencies. This TEL structure can be replaced by other existing temporal models
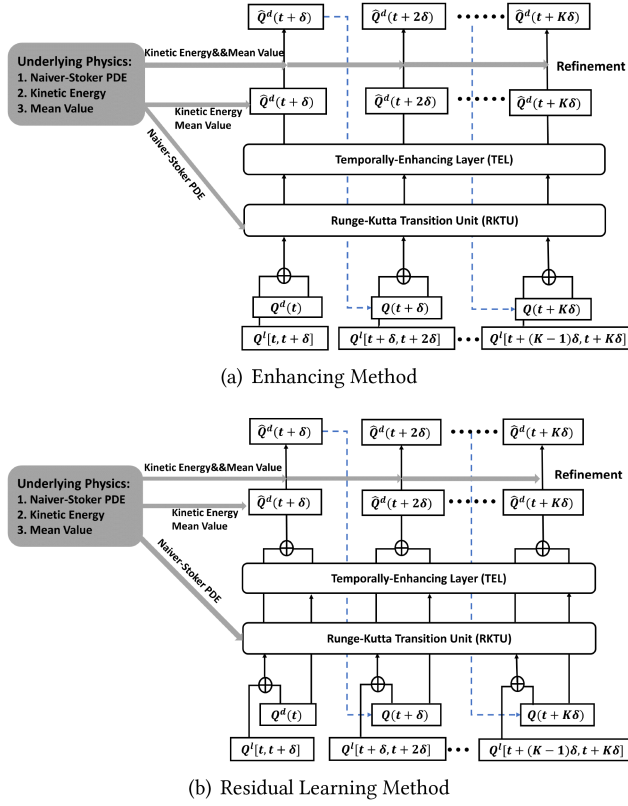
(a) Enhancing Method

(b) Residual Learning Method

Fig. 3. Details of CNDE for reconstructing $\mathbf{Q}$. Parts (a) and (b) depict two different approaches to incorporating the TEL component. The solid lines represent the forwarding process in CNDE-based methods. The blue dashed lines represent the reconstructed flow data $\hat{\mathbf{Q}}^d$ as used as input for the reconstruction of the next timestep.

such as TCN. Figure 3 shows two different approaches for integrating the TEL structure with the RKTU structure. In the first enhancing method shown in Figure 3(a), the RKTU output flow data $\hat{\mathbf{Q}}_{\text{RKTU}}$ are fed to the TEL structure, which is essentially an LSTM layer. After further processing through the TEL structure, the model produces the reconstructed flow data $\hat{\mathbf{Q}}^d(t)$. Given the true DNS data $\mathbf{Q}^d(t)$ in the training set, the reconstructed loss $\mathcal{L}_{\text{recon}}$ can be expressed using the **mean squared error (MSE)** loss,

$$\mathcal{L}_{\text{recon}} = \sum_t \text{MSE}(\hat{\mathbf{Q}}^d(t), \mathbf{Q}^d(t)). \tag{6}$$

The second method uses the TEL structure to complement the output of the RKTU structure, i.e., learning the residual of the RKTU output, as shown in Figure 3(b). In the training process, both true DNS data $\mathbf{Q}^d$ at time $\{t, \ldots t + (K-1)\delta\}$ and RKTU output $\hat{\mathbf{Q}}_{\text{RKTU}}$ are used to produce the corresponding temporal output feature $\hat{\mathbf{Q}}_{\text{TEL}}$ at time $\{t + \delta, \ldots, t + K\delta\}$. Then in the testing process, this method uses only the initial true DNS data $\mathbf{Q}^d$ in time $t + K\delta$ and the next series of predicted DNS data $\hat{\mathbf{Q}}^d$ as the DNS input to generate $\hat{\mathbf{Q}}_{\text{TEL}}$. Finally, this method adopts a linear combination to combine the RKTU output $\hat{\mathbf{Q}}_{\text{RKTU}}$ and corresponding TEL output $\hat{\mathbf{Q}}_{\text{TEL}}$ to obtain

the final reconstructed output $\hat{\mathbf{Q}}^d$, which can be represented as

$$\hat{\mathbf{Q}}^d(t) = w_r^t \hat{\mathbf{Q}}_{\text{RKTU}}(t) + w_t^t \hat{\mathbf{Q}}_{\text{TEL}}(t), \tag{7}$$

where w_r^t and w_t^t are trainable parameters. Finally, the reconstructed loss $\mathcal{L}_{\text{recon}}$ can also be represented by Equation (6).

3.3 Physical Constraints and Refinements

3.3.1 Physical Constraints. For a more accurately reconstructed field, some additional constraints are imposed on the data. Two such constraints are imposed by considering the consistency of (i) the mean velocity field and (ii) the kinetic energy of turbulence. For (i), the loss function $\mathcal{L}_{\text{mean}}$ between reconstructed data $\hat{\mathbf{Q}}^d$ and true DNS data $\mathbf{Q}^d$ is considered,

$$\mathcal{L}_{\text{mean}} = |\overline{\mathbf{Q}^d} - \overline{\hat{\mathbf{Q}}^d}|. \tag{8}$$

For (ii), the kinetic energy,], defined as

$$\mathcal{K} = \frac{1}{2}(u^2 + v^2 + w^2), \tag{9}$$

is monitored. For this, the loss function is $\mathcal{L}_{\text{kinetic}}$ is

$$\mathcal{L}_{\text{kinetic}} = |\mathcal{K}(\mathbf{Q}^d) - \mathcal{K}(\hat{\mathbf{Q}}^d)|, \tag{10}$$

where $\mathcal{K}(\mathbf{Q}^d)$ and $\mathcal{K}(\hat{\mathbf{Q}}^d)$ denote the kinetic energy of $\mathbf{Q}^d$ and $\hat{\mathbf{Q}}^d$, respectively. The overall loss function $\mathcal{L}$ is

$$\mathcal{L} = \alpha_0 \mathcal{L}_{\text{recon}} + \alpha_1 \mathcal{L}_{\text{mean}} + \alpha_2 \mathcal{L}_{\text{kinetic}}, \tag{11}$$

is considered in which α_0, α_1, and α_2 represent the hyperparameters to control the balance amongst the three constituents.

3.3.2 Degradation-based Refinement. As shown in Figure 3, the scheme preserves the physical constraints in the training process and also employs these constraints in the degradation-based test-time refinement process. The objective is to mitigate accumulated errors and structural distortions over long-term prediction by enforcing the physical consistency. The refinement process includes the same set of the loss function: the degradation loss $\mathcal{L}_{\text{deg}}$, the equal-mean loss $\mathcal{L}'_{\text{mean}}$, and the kinetic energy loss $\mathcal{L}'_{\text{kinetic}}$ loss. Since it is not possible to access true DNS data during the testing phase, the difference between true DNS $\mathbf{Q}^d$ and the reconstructed data $\hat{\mathbf{Q}}^d$ cannot be directly minimized. Thus, to protect the overall structure of flow data, a reverse degradation process is employed by using a separate convolutional network for mapping reconstructed data $\hat{\mathbf{Q}}^d$ to the corresponding low-resolution LES data $\hat{\mathbf{Q}}^l$. The loss $\mathcal{L}_{\text{deg}}$ between $\hat{\mathbf{Q}}^l$ and real LES data $\mathbf{Q}^l$ is

$$\mathcal{L}_{\text{deg}} = \text{MSE}(\hat{\mathbf{Q}}^l, \mathbf{Q}^l). \tag{12}$$

Also, the mean values from the true DNS cannot be used in the equal-mean loss function. Therefore, the corresponding values from the LES data are used as an approximation. As such, the equal-mean loss $\mathcal{L}'_{\text{mean}}$ between the reconstructed flow data $\hat{\mathbf{Q}}^d$ and the true LES data $\mathbf{Q}^l$ can be directly minimized,

$$\mathcal{L}'_{\text{mean}} = |\overline{\mathbf{Q}^l} - \overline{\hat{\mathbf{Q}}^d}|. \tag{13}$$

Similarly, the exact kinetic energy of flow data is not available during the testing period. These values are taken from the DNS in the training data,

$$\mathcal{L}'_{\text{kinetic}} = |\mathcal{K}(\hat{\mathbf{Q}}^d) - \tilde{\mathcal{K}}|. \tag{14}$$

The final refinement loss function is in the same format $\mathcal{L}' = \alpha_0 \mathcal{L}_{\text{deg}} + \alpha_1 \mathcal{L}'_{\text{mean}} + \alpha_2 \mathcal{L}'_{\text{kinetic}}$. The loss $\mathcal{L}'$ is adopted to directly adjust the state of reconstructed data for 10 epochs at each test-time timestep and yield an improved reconstruction performance.

4 MODEL APPRAISAL

4.1 Flows Considered

To assess the performance of the proposed methodology, the datasets pertaining to two turbulent flows are considered: a FIT flow [38] and the TGV [5] flow. In both cases, the mean velocity is zero, $\overline{\mathbf{V}}(t) = 0$, and the Reynolds number is large enough for the flow to exhibit turbulent characteristics.

The FIT data [38] are publicly available from the Johns Hopkins University. This dataset contains the original DNS of forced isotropic turbulence on a $1,024 \times 1,024 \times 1,024$ collocation points. The flow is forced by injecting energy into the flow at small waver numbers. The DNS data contain $5,024$ timesteps with time intervals of 0.002 s and includes both the velocity and the pressure fields. The original DNS data are downsampled to $64 \times 64 \times 64$ grids. The LES data are created on $16 \times 16 \times 16$ grids. The loss $\mathcal{L}'_{\text{kinetic}}$ is not considered for this flow.

The TGV [5] is an incompressible flow. The evolution of the TGV includes vorticity stretching and the consequent production of small-scale, dissipating eddies. A box flow, with a cubic periodic domain of $[-\pi, \pi]$ (in all three directions), is considered, with the following initial conditions:

$$
\begin{aligned}
u(x, y, z, 0) &= \sin(x)\cos(y)\cos(z), \\
v(x, y, z, 0) &= -\cos(x)\sin(y)\cos(z), \\
w(x, y, z, 0) &= 0.
\end{aligned}
\tag{15}
$$

The LES and DNS resolutions are $32 \times 32 \times 65$ and $128 \times 128 \times 65$, respectively. Both LES and DNS data are produced along the 65 equally spaced grid points along the z axis.

4.2 Comparative Assessments

4.2.1 CNDE Method and Baselines. The performance of the CNDE method is evaluated and compared with several existing methods for image SR and turbulent flow downscaling. Specifically, the proposed CNDE-based methods, CNDE-E (enhancing-based TEL method) and CNDE-R (residual learning-based TEL method),[1] were implemented. Additionally, four popular SR methods, namely SRCNN [15], RCAN [61], HDRN [47], and SRGAN [32]; two popular dynamic fluid downscaling methods, DCS/MS [19] and FSR [60]; and FNO [33] are used as baselines. To better verify the effectiveness of each of the model's components, four additional baselines are included: **convolutional transition network (CTN)**, RKTU, CNDEp-E, and CNDEp-R. The CTN is created by combining SRCNN and LSTM [24]. CNDEp-E and CNDEp-R are similar to CNDE-E and CNDE-R, but they are created without using the degradation-based refinement process.

By comparing the CTN with the RKTU, the objective is to demonstrate the advantages of the RKTU in spatio-temporal DNS reconstruction. By comparing the RKTU with the CNDEp-based methods, the goal is to show the effectiveness of introducing the TEL structure. The advantages of the refinement process are demonstrated by comparing the CNDEp-based and CNDE-based methods.

4.2.2 Experimental Designs. The proposed methods and the baselines are tested on both the FIT and the TGV datasets. The models are trained by using the FIT data from a consecutive one-second period with a time interval $\delta = 0.02$ s and a total of 50 timesteps and then apply the trained model into the next 0.4-s period (a total of 20 timesteps) for performance evaluation. For the TGV dataset,

---

[1]The source code is at https://drive.google.com/drive/folders/15PhF_q1HcJpXZIvxnR1mMbd8hbkxBbT_?usp=share_link

the models use a consecutive 40-s period with a time interval $\delta = 2$ s for training and the next 40 s of data for testing.

The performance of DNS reconstruction is evaluated by using two different metrics, **structural similarity index measure (SSIM)** [52] and dissipation [55]. SSIM is used to appraise the similarity between reconstructed data and target DNS on three aspects—luminance, contrast, and overall structure. The higher value of SSIM indicates better reconstruction performance. The dissipation operator is used to assess the performance of capturing the flow gradients. The dissipation of each of the three components of the velocity vector (u, v, and w) are evaluated. The dissipation operator is defined by

$$\chi(Q) \equiv \nabla Q \cdot \nabla Q = \left(\frac{\partial Q}{\partial x}\right)^2 + \left(\frac{\partial Q}{\partial y}\right)^2 + \left(\frac{\partial Q}{\partial z}\right)^2. \tag{16}$$

The dissipation is used to measure the difference of flow gradient between the true DNS and generated data. This is represented by $|\chi(Q^d) - \chi(\hat{Q}^d)|$. The lower value of this difference indicates better performance. Compared with our previous work [3], the performance assessment is expanded by considering a new pixelwise evaluation metric (dissipation) and a physical validation method based on the kinetic energy.

4.2.3 Environmental Settings and Implementation Details. The method is implemented via Tensorflow 2 with a GTX3080 GPU. The model is first trained in 500 epochs with ADAM optimizer [30] from an initial learning rate of 0.001. In the refinement step, the learning rate is lowered to 0.0005, and the training rate is iterated by 10 epochs. All the hidden variables and gating variables are in 32 dimensions. The values of α_0, α_1, and α_2 are set as 1, 0.1, and 0.1, respectively.

4.3 Reconstruction Performance

4.3.1 Quantitative Results. Table 1 and Table 2 summarize the average performance over the first 10 timesteps in the testing phase on both the FIT dataset and the TGV dataset. Compared with the baselines, CNDE-based methods perform the best in both evaluations obtaining the highest SSIM value and lowest dissipation difference. Several observations are made: (1) When comparing the CNDE-based methods with SR baselines, the DCS/MS, FNO, and FSR models, it is observed that these baseline methods cannot recover the overall flow well and get worse performance in terms of SSIM and dissipation difference. (2) Compared with the SRCNN, the CTN, which uses the LSTM model, shows a significant improvement in both evaluations. This confirms the effectiveness of a temporal model (e.g., LSTM) in capturing temporal dependency. (3) The comparison among the CTN, RKTU, CNDEp-based methods, and CNDE-based methods, indicates significant improvements by incorporating each of the three components (RKTU, TEL, and refinement). In particular, the refinement method brings the most significant improvement in terms of SSIM and dissipation differences.

4.3.2 Temporal Analysis. In the temporal analysis of the FIT dataset, the performance for reconstruction is measured for each step during a 0.4-s period (20 timesteps) in the testing phase. The performance change using the SSIM and the dissipation difference is shown in Figures 4 and 5, respectively. These figures indicate that (1) with larger time intervals between training data and prediction data, the performance becomes worse. In general, the CNDE-based methods are more stable over a long period and show a much better performance than other methods. (2) The temporal model (e.g., LSTM) results in significant improvements in long-term predictions. (3) The CNDE-based methods outperform the CNDEp-based methods, which demonstrate the effectiveness of test-time refinement in reducing the prediction bias in long-term prediction. (4) The CNDEp-based methods yield a better performance after the 5th timesteps compared with the

Table 1. Reconstruction Performance (Measured by SSIM, and
Dissipation Difference) on (u, v, w) Channels by Different
Methods in the FIT Dataset

Method	SSIM	Dissipation Difference
SRCNN	(0.859, 0.851, 0.851)	(0.301, 0.303, 0.303)
RCAN	(0.861, 0.859, 0.859)	(0.299, 0.301, 0.300)
HDRN	(0.861, 0.860, 0.862)	(0.298, 0.298, 0.297)
FSR	(0.861, 0.860, 0.861)	(0.299, 0.297, 0.296)
DCS/MS	(0.861, 0.862, 0.862)	(0.298, 0.295, 0.294)
SRGAN	(0.862, 0.861, 0.863)	(0.296, 0.294, 0.294)
FNO	(0.874, 0.875, 0.874)	(0.265, 0.266, 0.273)
CTN	(0.881, 0.880, 0.881)	(0.253, 0.254, 0.254)
RKTU	(0.898, 0.899, 0.898)	(0.260, 0.261, 0.259)
CNDEp-E	(0.909, 0.909, 0.907)	(0.244, 0.243, 0.245)
CNDEp-R	(0.904, 0.905, 0.905)	(0.249, 0.248, 0.248)
CNDE-E	(0.927, 0.921, 0.922)	(0.193, 0.194, 0.197)
CNDE-R	(0.921, 0.919, 0.920)	(0.196, 0.196, 0.200)

The performance is measured by the average results of the first 10
timesteps.

Table 2. Reconstruction Performance (Measured by SSIM, and
Dissipation Difference) on (u, v, w) Channels by Different Methods
in the TGV Dataset

Method	SSIM	Dissipation Difference×10
SRCNN	(0.602, 0.603, 0.626)	(0.083, 0.087, 0.079)
RCAN	(0.627, 0.622, 0.631)	(0.073, 0.074, 0.071)
HDRN	(0.638, 0.638, 0.641)	(0.072, 0.072, 0.068)
FSR	(0.646, 0.648, 0.649)	(0.070, 0.073, 0.066)
DSC/MS	(0.647, 0.649, 0.649)	(0.070, 0.071, 0.065)
SRGAN	(0.661, 0.658, 0.666)	(0.068, 0.067,0.058)
FNO	(0.645, 0.646, 0.648)	(0.072, 0.071, 0.072)
CTN	(0.623, 0.624, 0.627)	(0.093, 0.096, 0.087)
RKTU	(0.708, 0.708, 0.688)	(0.049, 0.046, 0.043)
CNDEp-E	(0.724, 0.723, 0.708)	(0.046, 0.041, 0.039)
CNDEp-R	(0.720, 0.719, 0.701)	(0.046, 0.045, 0.040)
CNDE-E	(0.938, 0.918, 0.876)	(0.031, 0.032, 0.026)
CNDE-R	(0.917, 0.909, 0.877)	(0.033, 0.034, 0.028)

The performance is measured by the average results of the first 10
timesteps.

temporal baseline CTN model. This indicates the advantage of the RKTU structure in the long-term prediction. (5) The CNDEp-E slightly outperforms the CNDEp-R in the long-term prediction. A similar observation is made by comparing two versions of CNDE-based methods.

In Figures 6 and 7, the results for the TGV are presented. A better performance of the model developed here is indicated via the SSIM and dissipation differences. Several observations are made: (1) The CNDE-based methods using refinement perform much better than CNDEp-based methods and DCS/MS. Moreover, the performance of the CNDEp-based methods becomes worse than the baseline DCS/MS after the fifth timestep. This is because of the variability of TGV data over larger time intervals ($\delta = 2$ s) and the testing data are very different from the initial data point. This causes
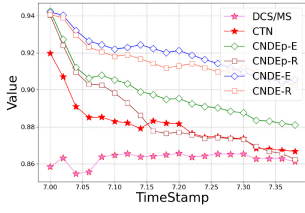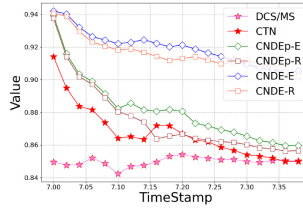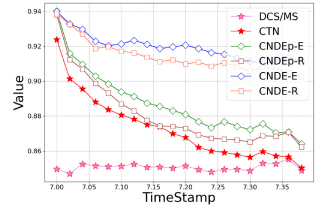
(a) u Channel.   (b) v Channel.   (c) w Channel.

Fig. 4. Change of SSIM values produced by different models from the 1st (7 s) to 20th (7.4 s) timesteps in the FIT dataset.
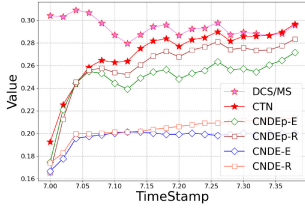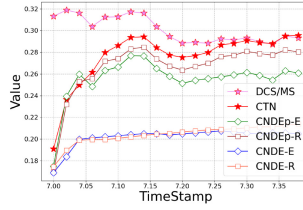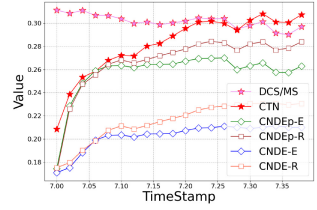

(a) u Channel.   (b) v Channel.   (c) w Channel.

Fig. 5. Change of dissipation difference by different models from the 1st (7 s) to 20th (7.4 s) timesteps in the FIT dataset.
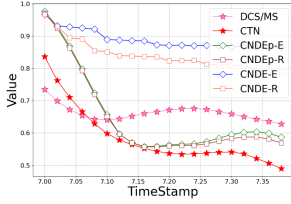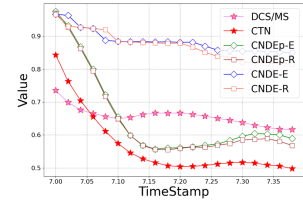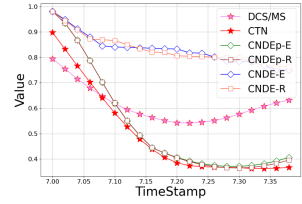

(a) u Channel.   (b) v Channel.   (c) w Channel.

Fig. 6. Change of SSIM values produced by different models from the 1st (80s) to 20th (120s) timesteps in the TGV dataset.
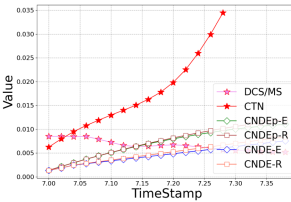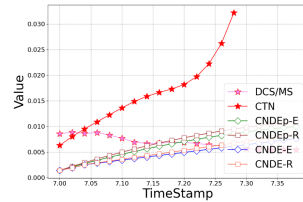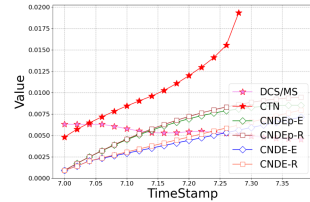

(a) u Channel.   (b) v Channel.   (c) w Channel.

Fig. 7. Change of dissipation difference produced by different models from the 1st (80s) to 20th (120s) timesteps in the TGV dataset.

the CNDEp-based methods to fail in capturing the correct flow dynamic without refinement. It also indicates the advantages of the refinement method for adjusting the state of flow data in the long-term prediction. (2) The CTN almost fails to capture the flow dynamics after the fifth timestep, and thus the CTN is not suitable for this dataset.

Fig. 8. Reconstructed w channel by each method on a sample testing slice along the z dimension in the FIT dataset. The reconstruction results are 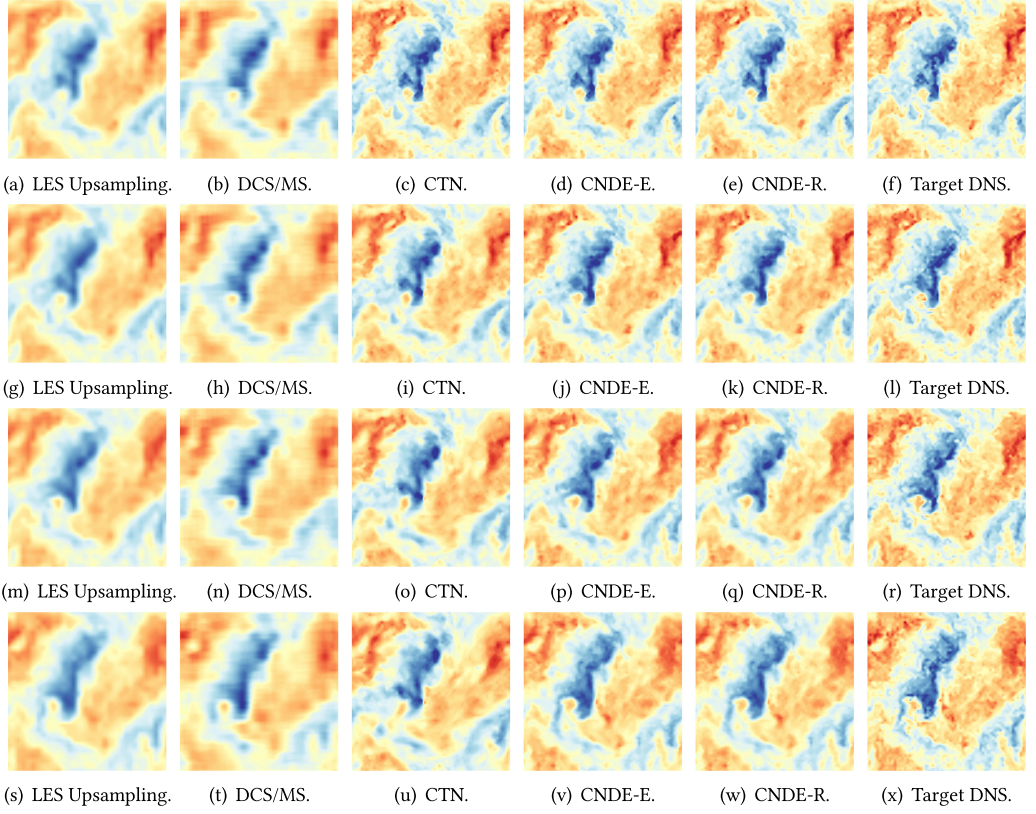shown at the 1st (7 s), 5th (7.1 s) 10th (7.2 s), and 20th (7.4 s) in (a)–(f), (g)–(l), (m)–(r), and (s)–(x), respectively.

4.3.3   Visualization. In Figures 8, the reconstructed data are shown at multiple (1st, 5th, 10th, and 20th) timesteps after the training period. For each timestep, the slices of the w component at a specified z value are shown. In the 1st step, both the CNDE-based methods and the baseline CTN model yield ideal reconstruction results. This is because the test data are similar to the training data at the last timestep. In contrast, the baseline DSC/MS [19] leads to a poor performance starting from early time. Beginning at the 5th timestep, the CNDE-based methods perform better than the baselines. A more significant difference is observed at the 20th timestep. All the baselines almost fail to capture the correct flow transport pattern. The CNDE-based methods yield a much better performance in the late stage. Similar observations are made on the TGV dataset as shown in Figure 9.

4.3.4   Validation via Physical Metrics. The model performance is also assessed via of long-term prediction of the turbulent kinetic energy. Figure 10 show the energies corresponding to the target DNS, and the reconstructed flow data by the baselines and the CNDE-based methods for both the FIT and the TGV flows. The results in Figure 10(a) for the FIT dataset indicate the following: (1) The CNDE-based methods in general perform better than the baseline method DCS/MS and CTN. Even without using the refinement process, the CNDEp-based methods outperform the DCS/MS and CTN models. CNDE-based methods can follow the underlying physical rule well in the long-term prediction. (2) The performance of CNDEp-based methods becomes very poor

(a) LES Upsampling. (b) DCS/MS. (c) CTN. (d) CNDE-E. (e) CNDE-R. (f) Target DNS.

(g) LES Upsampling. (h) DCS/MS. (i) CTN. (j) CNDE-E. (k) CNDE-R. (l) Target DNS.

(m) LES Upsampling. (n) DCS/MS. (o) CTN. (p) CNDE-E. (q) CNDE-R. (r) Target DNS.

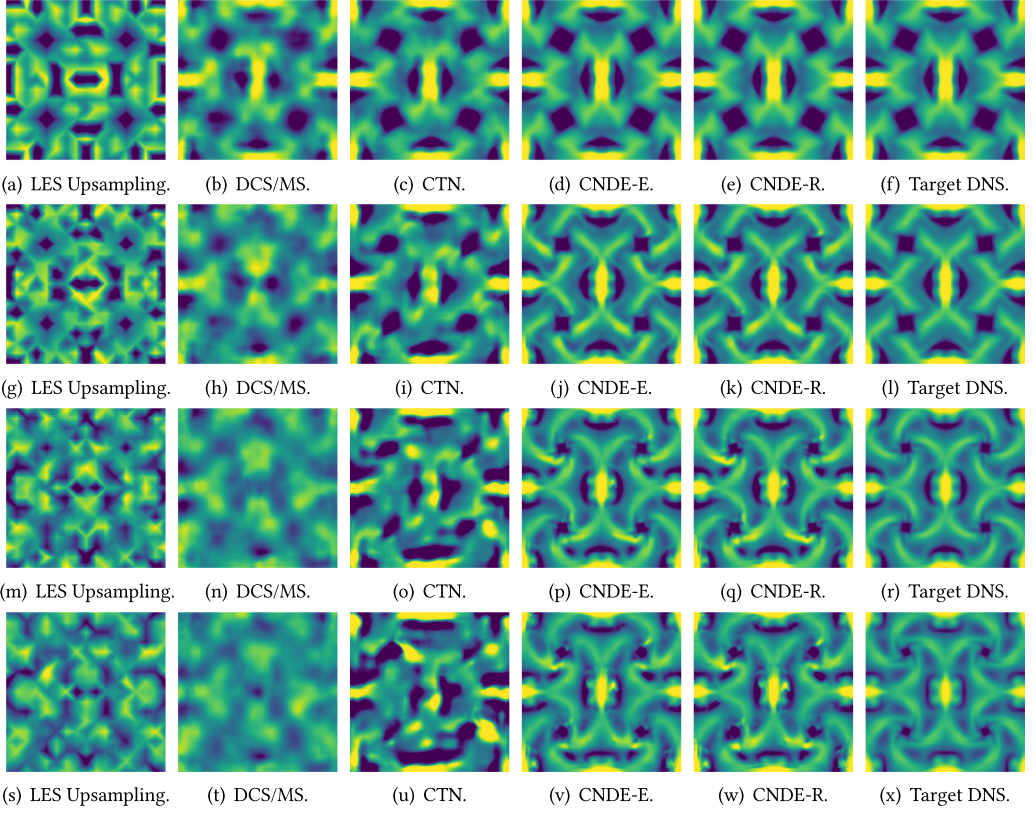(s) LES Upsampling. (t) DCS/MS. (u) CTN. (v) CNDE-E. (w) CNDE-R. (x) Target DNS.

Fig. 9. Reconstructed w channel by each method on a sample testing slice along the z dimension in the TGV dataset. The reconstruction results are shown at the 1st (80 s), 5th (90 s), 10th (100 s), and 15th (110 s) in (a)–(f), (g)–(l), (m)–(r), and (s)–(x), respectively.

(a) FIT data.
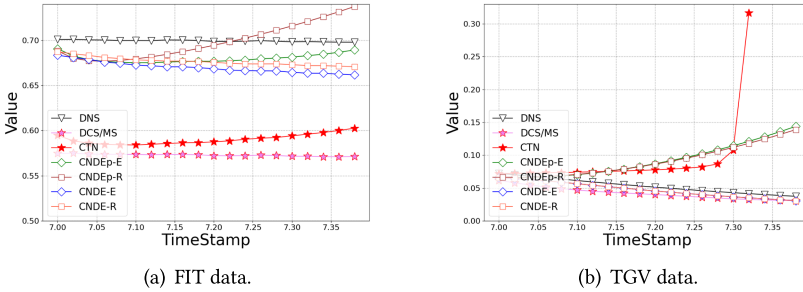
(b) TGV data.

Fig. 10. Change of kinetic energy produced by the reference DNS and different models in both the FIT and the TGV datasets, respectively.

after the 8th timestep. This is because the accumulated error gets amplified in every timestep. The results in Figure 10(b) yield similar conclusions.

5   SUMMARY AND CONCLUDING REMARKS

A novel SR methodology, termed CNDE, is developed to reconstruct high-resolution flow data in spatial and temporal fields. The model is used in the setting of unsteady, incompressible,

Newtonian turbulent flow under spatially homogeneous conditions. The SR method is to generate the high-resolution DNS field from low resolution, LES data. An RKTU is developed to leverage the physical knowledge embodied in the Navier–Stokes equation to capture the spatial resolution and the temporal dynamics of the flow. A TEL is constructed to capture long-term temporal dynamics. A degradation-based refinement method is developed to adjust the reconstructed data over time by enforcing the consistency with physical constraints. The performance of the model is assessed in the setting of two flow configurations via flow visualization and statistical analysis. The results demonstrate the superiority of the CNDE for spatio-temporal reconstruction of the flow. The model's constituents, the RKTU and the refinement methods can be used as building blocks to enhance existing deep learning models.

Despite its demonstrated capabilities, there are two limitations associated with the CNDE model in its current form. (1) The CNN layers are used to estimate spatial derivatives, which can introduce bias due to the approximation and due to data overfitting. (2) The method is, thus far, tailored and appraised for specific flows. Therefore, its generality cannot be warranted for other applications; especially in the absence of sufficient DNS data. Future work is recommended to find alternative ways to evaluate the spatial derivatives more accurately and to improve the model's transferability.

ACKNOWLEDGMENTS

REFERENCES

[1] Namhyuk Ahn, Byungkon Kang, and Kyung-Ah Sohn. 2018. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*. 252–268.

[2] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. 2017. Understanding of a convolutional neural network. In *Proceedings of the International Conference on Engineering and Technology (ICET'17)*. IEEE, 1–6.

[3] Tianshu Bao, Shengyu Chen, Taylor T. Johnson, Peyman Givi, Shervin Sammak, and Xiaowei Jia. 2022. Physics guided neural networks for spatio-temporal super-resolution of turbulent flows. In *Proceedings of the 38th Conference on Uncertainty in Artificial Intelligence*.

[4] Oussama Boussif, Yoshua Bengio, Loubna Benabbou, and Dan Assouline. 2022. MAgnet: Mesh agnostic neural PDE solver. *Adv. Neural Inf. Process. Syst.* 35 (2022), 31972–31985.

[5] Marc E. Brachet, D. Meiron, S. Orszag, B. Nickel, R Morf, and Uriel Frisch. 1984. The Taylor-green vortex and fully developed turbulence. *J. Stat. Phys.* 34, 5 (1984), 1049–1063.

[6] John Butcher. 2007. Runge-Kutta methods. *Scholarpedia* 2, 9 (2007), 3147.

[7] Shengyu Chen, Nasrin Kalanat, Yiqun Xie, Sheng Li, Jacob A. Zwart, Jeffrey M. Sadler, Alison P. Appling, Samantha K. Oliver, Jordan S. Read, and Xiaowei Jia. 2023. Physics-guided machine learning from simulated data with different physical parameters. *Knowl. Inf. Syst.* 65, 8 (2023), 3223–3250.

[8] Shengyu Chen, Shervin Sammak, Peyman Givi, Joseph P. Yurko, and Xiaowei Jia. 2021. Reconstructing high-resolution turbulent flows using physics-guided neural networks. In *Proceedigns of the IEEE International Conference on Big Data (Big Data'21)*. IEEE, 1369–1379.

[9] Shengyu Chen, Jacob A Zwart, and Xiaowei Jia. 2022. Physics-guided graph meta learning for predicting water temperature and streamflow in stream networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2752–2761.

[10] Yu Chen, Ying Tai, Xiaoming Liu, Chunhua Shen, and Jian Yang. 2018. Fsrnet: End-to-end learning face super-resolution with facial priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2492–2501.

[11] Wenlong Cheng, Mingbo Zhao, Zhiling Ye, and Shuhang Gu. 2021. Mfagan: A compression framework for memory-efficient on-device super-resolution gan. arXiv:2107.12679. Retrieved from https://arxiv.org/abs/2107.12679

[12] Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. 2019. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR'19)*. 11057–11066. https://doi.org/10.1109/CVPR.2019.01132

[13] Arka Daw, Anuj Karpatne, William D. Watkins, Jordan S. Read, and Vipin Kumar. 2022. Physics-guided neural networks (pgnn): An application in lake temperature modeling. In *Knowledge Guided Machine Learning*. Chapman & Hall/CRC, 353–372.

[14] Zhiwen Deng, Chuangxin He, Yingzheng Liu, and Kyung Chun Kim. 2019. Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys. Fluids* 31, 12 (2019), 125111.

[15] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. 2014. Learning a deep convolutional network for image super-resolution. In *Proceedings of the 13th European Conference on Computer Vision (ECCV'14), Part IV 13*. Springer, 184–199.

[16] Léonard Equer, T Konstantin Rusch, and Siddhartha Mishra. 2023. Multi-scale message passing neural pde solvers. arXiv:2302.03580. Retrieved from https://arxiv.org/abs/2302.03580

[17] Chaowei Fang, Dingwen Zhang, Liang Wang, Yulun Zhang, Lechao Cheng, and Junwei Han. 2022. Cross-modality high-frequency transformer for mr image super-resolution. In *Proceedings of the 30th ACM International Conference on Multimedia*. 1584–1592.

[18] Jinsheng Fang, Hanjiang Lin, Xinyu Chen, and Kun Zeng. 2022. A hybrid network of cnn and transformer for lightweight image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1103–1112.

[19] Kai Fukami, Koji Fukagata, and Kunihiko Taira. 2019. Super-resolution reconstruction of turbulent flows with machine learning. *J. Fluid Mech.* 870 (2019), 106–120.

[20] Kai Fukami, Koji Fukagata, and Kunihiko Taira. 2021. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *J. Fluid Mech.* 909 (2021), A9.

[21] P. Givi. 1994. Spectral and random vortex methods in turbulent reacting flows. In *Turbulent Reacting Flows*, P. A. Libby and F. A. Williams (Eds.). Academic Press, London, UK, Chapter 8, 475–572.

[22] Paul C. Hanson et al. 2020. Predicting lake surface water phosphorus dynamics using process-guided machine learning. *Ecol. Model.* 430 (2020), 109136.

[23] Erhu He, Yiqun Xie, Licheng Liu, Weiye Chen, Zhenong Jin, and Xiaowei Jia. 2023. Physics guided neural networks for time-aware fairness: An application in crop yield prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 14223–14231.

[24] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.* 9, 8 (1997), 1735–1780.

[25] Xiaowei Jia, Shengyu Chen, Can Zheng, Yiqun Xie, Zhe Jiang, and Nasrin Kalanat. 2023. Physics-guided graph diffusion network for combining heterogeneous simulated data: An application in predicting stream water temperature. In *Proceedings of the SIAM International Conference on Data Mining (SDM'23)*. SIAM, 361–369.

[26] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. 2019. Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 558–566.

[27] Cai Jing, Yang Jinsheng, and Ding Runtao. 2000. Fuzzy weighted average filter. In *Proceedings of the 5th International Conference on Signal Processing and the 16th World Computer Congress (WCC-ICSP'00)*, Vol. 1. IEEE, 525–528.

[28] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2017. Progressive growing of gans for improved quality, stability, and variation. arXiv:1710.10196. Retrieved from https://arxiv.org/abs/1710.10196

[29] Ankush Khandelwal, Shaoming Xu, Xiang Li, Xiaowei Jia, Michael Stienbach, Christopher Duffy, John Nieber, and Vipin Kumar. 2020. Physics guided machine learning methods for hydrology. arXiv:2012.02854. Retrieved from https://arxiv.org/abs/2012.02854

[30] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980

[31] Colin Lea, Rene Vidal, Austin Reiter, and Gregory D. Hager. 2016. Temporal convolutional networks: A unified approach to action segmentation. In *European Conference on Computer Vision*. Springer, 47–54.

[32] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4681–4690.

[33] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. 2020. Fourier neural operator for parametric partial differential equations. arXiv:2010.08895. Retrieved from https://arxiv.org/abs/2010.08895

[34] Zhengyu Liang, Yingqian Wang, Longguang Wang, Jungang Yang, and Shilin Zhou. 2022. Light field image super-resolution with transformers. *IEEE Sign. Process. Lett.* 29 (2022), 563–567.

[35] Bo Liu, Jiupeng Tang, Haibo Huang, and Xi-Yun Lu. 2020. Deep learning methods for super-resolution reconstruction of turbulent flows. *Phys. Fluids* 32, 2 (2020), 025105.

[36] Licheng Liu, Shaoming Xu, Jinyun Tang, Kaiyu Guan, Timothy J. Griffis, Matthew D. Erickson, Alexander L. Frie, Xiaowei Jia, Taegon Kim, Lee T. Miller, et al. 2022. KGML-ag: A modeling framework of knowledge-guided machine learning to simulate agroecosystems: A case study of estimating N 2 O emission using data from mesocosm experiments. *Geosci. Model Dev.* 15, 7 (2022), 2839–2858.

[37] Zhisheng Lu, Juncheng Li, Hong Liu, Chaoyan Huang, Linlin Zhang, and Tieyong Zeng. 2022. Transformer for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 457–466.

[38] Wan Minping, Shiyi Chen, Gregory Eyink, Charles Meneveau, Perry Johnson, Eric Perlman, Randal Burns, Yi Li, Alex Szalay, and Stephen Hamilton. 2012. Forced Isotropic Turbulence data set (Extended).

[39] Nikhil Muralidhar, Jie Bu, Ze Cao, Long He, Naren Ramakrishnan, Danesh Tafti, and Anuj Karpatne. 2020. PhyNet: Physics guided neural networks for particle drag force prediction in assembly. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, 559–567.

[40] Arash G. Nouri, Mehdi B Nik, Pope Givi, Daniel Livescu, and Stephen B. Pope. 2017. Self-contained filtered density function. *Phys. Rev. Fluids* 2, 9 (2017), 094603.

[41] Sung Cheol Park, Min Kyu Park, and Moon Gi Kang. 2003. Super-resolution image reconstruction: A technical overview. *IEEE Sign. Process. Mag.* 20, 3 (2003), 21–36.

[42] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *International Conference on Machine Learning*. PMLR, 4055–4064.

[43] Jordan S. Read, Xiaowei Jia, Jared Willard, Alison P. Appling, Jacob A. Zwart, Samantha K. Oliver, Anuj Karpatne, Gretchen J. A. Hansen, Paul C. Hanson, William Watkins, et al. 2019. Process-guided deep learning predictions of lake water temperature. *Water Resour. Res.* 55, 11 (2019), 9173–9190.

[44] Pierre Sagaut. 2005. *Large Eddy Simulation for Incompressible Flows: An Introduction.* Springer Science & Business Media.

[45] Ying Tai, Jian Yang, and Xiaoming Liu. 2017. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3147–3155.

[46] Uddeshya Upadhyay and Suyash P. Awate. 2019. Robust super-resolution GAN, with manifold-based and perception loss. In *Proceedings of the IEEE 16th International Symposium on Biomedical Imaging (ISBI'19)*. IEEE, 1372–1376.

[47] Vinh Van Duong, Thuc Nguyen Huu, Jonghoon Yim, and Byeungwoo Jeon. 2021. A fast and efficient super-resolution network using hierarchical dense residual learning. In *Proceedings of the IEEE International Conference on Image Processing (ICIP'21)*. IEEE, 1809–1813.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Adv. Neural Inf. Process. Syst.* 30 (2017).

[49] Shunzhou Wang, Tianfei Zhou, Yao Lu, and Huijun Di. 2022. Detail-preserving transformer for light field image super-resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 2522–2530.

[50] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. 2018. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 606–615.

[51] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. 2018. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the ECCV Workshops*.

[52] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* 13, 4 (2004), 600–612.

[53] Zhang Wenlong, Liu Yihao, Chao Dong, and Yu Qiao. 2021. RankSRGAN: Generative adversarial networks with ranker for image super-resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 10 (2021), 1–1.

[54] Wikipedia contributors. 2022. Finite Difference Method. .Retrieved January 25, 2023 from https://en.wikipedia.org/w/index.php?title=Finite_difference_method&oldid=1126400243.

[55] Wikipedia contributors. 2022. Laplace Operator. .Retrieved January 23, 2023 from https://en.wikipedia.org/w/index.php?title=Laplace_operator&oldid=1127277109.

[56] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. 2022. Integrating scientific knowledge with machine learning for engineering and environmental systems. *Comput. Surv.* 55, 4 (2022), 1–37.

[57] You Xie, Erik Franz, Mengyu Chu, and Nils Thuerey. 2018. tempogan: A temporally coherent, volumetric gan for super-resolution fluid flow. *ACM Trans. Graph.* 37, 4 (2018), 1–15.

[58] Qin Xu, Zijian Zhuang, Yongcai Pan, and Binghai Wen. 2023. Super-resolution reconstruction of turbulent flows with a transformer-based deep learning framework. *Phys. Fluids* 35, 5 (2023).

[59] Fuzhi Yang, Huan Yang, Jianlong Fu, Hongtao Lu, and Baining Guo. 2020. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5791–5800.

[60] Zhen Yang, Hua Yang, and Zhouping Yin. 2023. Super-resolution reconstruction for the three-dimensional turbulence flows with a back-projection network. *Phys. Fluids* 35, 5 (2023).

[61] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. 2018. Image super-resolution using very
      deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV'18)*.
      286–301.
[62] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. 2018. Residual dense network for image super-
      resolution. arXiv:1802.08797. Retrieved from https://arxiv.org/abs/1802.08797
[63] Wenbin Zou, Tian Ye, Weixin Zheng, Yunchen Zhang, Liang Chen, and Yi Wu. 2022. Self-calibrated efficient trans-
      former for lightweight super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern
      Recognition*. 930–939.