

Prediction and control of two-dimensional decaying turbulence using generative adversarial networks

Jiyeon Kim¹, Junhyuk Kim² and Changhoon Lee^{1,3,†}

¹School of Mathematics and Computing, Yonsei University, Seoul 03722, Korea

²Korea Atomic Energy Research Institute, Daejeon 34057, Korea

³Department of Mechanical Engineering, Yonsei University, Seoul 03722, Korea

(Received 3 July 2023; revised 2 January 2024; accepted 16 January 2024)

An accurate prediction of turbulence has been very costly since it requires an infinitesimally small time step for advancing the governing equations to resolve the fast-evolving small-scale motions. With the recent development of various machine learning (ML) algorithms, the finite-time prediction of turbulence became one of promising options to relieve the computational burden. Yet, a reliable prediction of the small-scale motions is challenging. In this study, PredictionNet, a data-driven ML framework based on generative adversarial networks (GANs), was developed for fast prediction of turbulence with high accuracy down to the smallest scale using a relatively small number of parameters. In particular, we conducted learning of two-dimensional (2-D) decaying turbulence at finite lead times using direct numerical simulation data. The developed prediction model accurately predicted turbulent fields at a finite lead time of up to half the Eulerian integral time scale over which the large-scale motions remain fairly correlated. Scale decomposition was used to interpret the predictability depending on the spatial scale, and the role of latent variables in the discriminator network was investigated. The good performance of the GAN in predicting small-scale turbulence is attributed to the scale-selection and scale-interaction capability of the latent variable. Furthermore, by utilising PredictionNet as a surrogate model, a control model named ControlNet was developed to identify disturbance fields that drive the time evolution of the flow field in the direction that optimises the specified objective function.

Key words: machine learning, turbulence simulation

† Email address for correspondence: clee@yonsei.ac.kr

© The Author(s), 2024. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

1. Introduction

Turbulence is a multiscale nonlinear dynamic phenomenon frequently observed in various flows in nature and industry. Certain deterministic dynamic features such as coherent structures have been found in turbulence (Hussain 1986); however, the behaviour of turbulence in most flows is chaotic. These characteristics make the accurate prediction of turbulence challenging despite the existence of governing equations for turbulence, called the Navier–Stokes equations. With the qualitative and quantitative expansion of computing resources over the past decades, various numerical approaches have been proposed. Direct numerical simulation (DNS), a full-resolution approach that can provide the most detailed description, is restricted to low Reynolds numbers. Moreover, the spatial filtering strategy of large-eddy simulation (LES) and the temporal averaging approach of the Reynolds-averaged Navier–Stokes (RANS) model, which provide relatively fast solutions, lack reliable general closure models. Most importantly, these traditional numerical approaches are based on the temporal advancement of the governing partial differential equations, which is still costly to use in practice, even with ever-increasing computing power since the small-scale motions, essential in turbulence evolution, need to be properly resolved.

Recently, machine learning (ML) and other data-driven approaches have become popular in many areas of science, engineering and technology owing to their effectiveness and efficiency in dealing with complex systems. Certainly, efforts have been made to apply ML to turbulence problems, particularly in fluid dynamics (Brenner, Eldredge & Freund 2019; Duraisamy, Iaccarino & Xiao 2019; Brunton, Noack & Koumoutsakos 2020). Other studies have attempted to develop new closure models for turbulence models using ML, such as subgrid-scale models (Gamahara & Hattori 2017; Beck, Flad & Munz 2018; Maulik *et al.* 2018; Guan *et al.* 2021; Kim *et al.* 2022). Moreover, wall models for various flows in LES have been proposed (Yang *et al.* 2019; Zhou, He & Yang 2021; Bae & Koumoutsakos 2022; Dupuy, Odier & Lapeyre 2023; Lozano-Durán & Bae 2023; Vadrot *et al.* 2023), and improvements in closure models for RANS have been attempted (Duraisamy, Zhang & Singh 2015; Ling, Kurzawski & Templeton 2016; Parish & Duraisamy 2016; Singh, Medida & Duraisamy 2017; Wu, Xiao & Paterson 2018; Duraisamy *et al.* 2019; Zhu *et al.* 2019). Some attempts have yielded promising results; however, more exploration is required to secure reliable general closure models.

Another approach using ML to predict turbulence dynamics based on reduced-order modelling (ROM) has been proposed. With low-dimensional representations obtained through mathematical decomposition, such as proper orthogonal decomposition (Sirovich 1987), Koopman operator (Mezić 2005, 2013) and dynamic mode decomposition (Schmid 2010), or using recent network architectures such as autoencoder (AE) and convolutional neural networks (CNNs), the governing dynamics of latent space are trained using dynamic models, such as recurrent neural networks (RNNs). For example, Hennigh (2017) developed a model called Lat-Net based on the lattice Boltzmann method data by applying an AE structure. King *et al.* (2018) developed a compressed convolutional long short-term memory (LSTM) combining AE and LSTM and showed that dynamic approaches such as Lat-Net and their method are more effective in reflecting turbulence physics, compared with static approaches. Wang *et al.* (2018) and Mohan & Gaitonde (2018) efficiently predicted the coefficients of basis functions using LSTM after applying proper orthogonal decomposition to various flows. Mohan *et al.* (2019) extended the range of prediction to the forced homogeneous isotropic turbulence with two passive scalars. Srinivasan *et al.* (2019) confirmed the applicability of LSTM to wall-bounded near-wall turbulence using the nine-equation shear flow model. A recent study by Nakamura *et al.* (2021) successfully

applied nonlinear mode decomposition to predict the minimal turbulent channel flow using a CNN-based AE. Although ROM-based methods are efficient and easy to analyse, system characteristics such as nonlinear transients and multiscale phenomena can be easily lost during information compression when only the dominant modes are used. However, as complexity of ROM-based methods increases, models tend to capture most physical phenomena of turbulence. For example, as reported by Nakamura *et al.* (2021), approximately 1500 modes were found to be sufficient to fully reconstruct turbulence characteristics. Then a question arises on how many modes need to be considered and the number of modes to properly represent turbulence might be not as small as intended.

The most basic and popular ML models are artificial neural networks (ANNs), also known as multilayer perceptrons, which determine nonlinear functional relationships between the input and output data and are relatively easy to train (Beck & Kurz 2021). However, when the input is in the form of a field, CNNs effectively capture embedded spatial patterns or correlations. In our previous work on the prediction of turbulent heat transfer (Kim & Lee 2020*b*), a CNN successfully reconstructed the wall heat-flux distribution based on wall shear stresses. However, CNN-based models whose objective function is to minimise the pointwise mean-squared difference between the predicted and target fields sometimes produce blurry outputs (Kim & Lee 2020*a*; Kim *et al.* 2021). Conversely, generative adversarial networks (GANs) (Goodfellow *et al.* 2014), in which a generator (G) and discriminator (D) network are trained simultaneously in an adversarial manner such that G is trained to generate high-quality data whereas D is trained to distinguish generated data from target data, can produce better output than CNNs (Deng *et al.* 2019; Kim & Lee 2020*a*; Kim *et al.* 2021; Kim, Kim & Lee 2023). Lee & You (2019) also showed that GANs are better in long-term prediction of unsteady flow over a circular cylinder. Ravuri *et al.* (2021) applied a deep generative model to precipitation nowcasting and reported a much higher accuracy with small-scale features than other ML models and numerical weather-prediction systems. GANs appear to capture the statistical characteristics of fields better than CNNs; we focus on this capability of GAN in this study.

First, we selected two-dimensional (2-D) decaying homogeneous isotropic turbulence (DHIT), which is essential in fields such as weather forecasting (Shi *et al.* 2015; Rüttgers *et al.* 2019; Liu & Lee 2020; Ravuri *et al.* 2021); it is relatively simple; thus, its prediction can be performed at a reasonable cost. In addition, the study of 2-D turbulence was initially considered as a simplified version of three-dimensional (3-D) turbulence; however, it was studied extensively (Sommeria 1986; Brachet *et al.* 1988; McWilliams 1990; McWilliams, Weiss & Yavneh 1994; Jiménez, Moffatt & Vasco 1996) after its unique characteristics related to geophysical and astrophysical problems, such as strongly rotating stratified flow, are revealed (Alexakis & Doering 2006). The primary goal of this study was to develop a high-accuracy prediction model for 2-D DHIT called PredictionNet based on GANs, that produces the evolution of turbulence by reflecting spatiotemporal statistical characteristics. Successfully trained PredictionNet could predict 2-D turbulence more accurately in various aspects than a baseline CNN. Although proving why GANs are better in the prediction of turbulence statistics than CNNs is prohibitively hard, we performed various quantitative statistical analyses regarding the predictive accuracy depending on time and spatial scales to provide some clues on the working principle of the GAN model. By considering scale decomposition in the analysis of the behaviour of the latent variable, we discovered that the discriminator network of a GAN possesses a scale-selection capability, leading to the successful prediction of small-scale turbulence.

Second, flow control becomes feasible if accurate flow prediction is possible. The application of ML to turbulence control dates back to Lee *et al.* (1997), who used a

neural network for turbulence control to reduce drag in a turbulent channel flow. Recently, various studies that applied ML to flow control for drag reduction in turbulent channel flow (Han & Huang 2020; Park & Choi 2020; Lee, Kim & Lee 2023), drag reduction of flow around a cylinder (Rabault *et al.* 2019; Rabault & Kuhnle 2019; Tang *et al.* 2020) and object control (Colabrese *et al.* 2017; Verma, Novati & Koumoutsakos 2018) have been conducted, yielding successful control results. However, in this study, for a fundamental understanding of the control mechanism in 2-D turbulence, we considered determining the optimum disturbance field, which can modify the flow in the direction of optimising the specified objective function. Thus, we combined PredictionNet and ControlNet for specific purposes. A target where the flow control can be used meaningfully, such as maximising the propagation of the control effect of the time-evolved flow field, was set, and the results were analysed and compared with the results of similar studies (Jiménez 2018; Yeh, Meena & Taira 2021).

Following this introduction, § 2 describes the process of collecting datasets to be used for training and testing. In § 3, ML methodologies such as objective functions and network architectures are explained. The prediction and control results are subdivided and analysed qualitatively and quantitatively in § 4, and a conclusion is drawn in § 5.

2. Data collection

For decaying 2-D turbulence, which is our target for prediction and control, DNS was performed by solving the incompressible Navier–Stokes equations in the form of the vorticity transport equation without external forcing:

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \nu \frac{\partial^2 \omega}{\partial x_j \partial x_j}, \quad (2.1)$$

with

$$\frac{\partial^2 \psi}{\partial x_j \partial x_j} = -\omega, \quad (2.2)$$

where $\omega(x_1, x_2, t)$ is the vorticity field with $x_1 = x$ and $x_2 = y$ and ν is the kinematic viscosity. Here ψ denotes the stream function that satisfies $u_1 = u = \partial \psi / \partial y$ and $u_2 = v = -\partial \psi / \partial x$. A pseudo-spectral method with 3/2 zero padding was adopted for spatial discretisation. The computational domain size to which the bi-periodic boundary condition was applied was a square box of $[0, 2\pi)^2$, and the number of spatial grids, $N_x \times N_y$, was 128×128 . The Crank–Nicolson method for the viscous term and second-order Adams–Bashforth method for the convective term were used for temporal advancement. In Appendix A, it was proven that the pseudo-spectral approximation to ((2.1) and (2.2)) in a bi-periodic domain is equivalent to pseudo-spectral approximation to the rotational form of the 2-D Navier–Stokes equations. For the training, validation and testing of the developed prediction network, 500, 100 and 50 independent simulations with random initialisations were performed, respectively.

Training data were collected at discrete times, $t_i (= t_0 + i\delta t, i = 0, 1, 2, \dots, 100)$ and $t_i + T$ (T is the target lead time for prediction), where $\delta t (= 20\Delta t)$ is the data time step, and Δt denotes the simulation time step. We selected t_0 such that the initial transient behaviour due to the prescribed initial condition in the simulation had disappeared sufficiently; thus, the power-law spectrum of enstrophy ($\Omega(k) \propto k^{-1}$ where k is the wavenumber; Brachet *et al.* 1988) was maintained. During this period, the root-mean-square (r.m.s.) vorticity magnitude ω' and the average dissipation rate ε decay in the form $\sim t^{-0.56}$

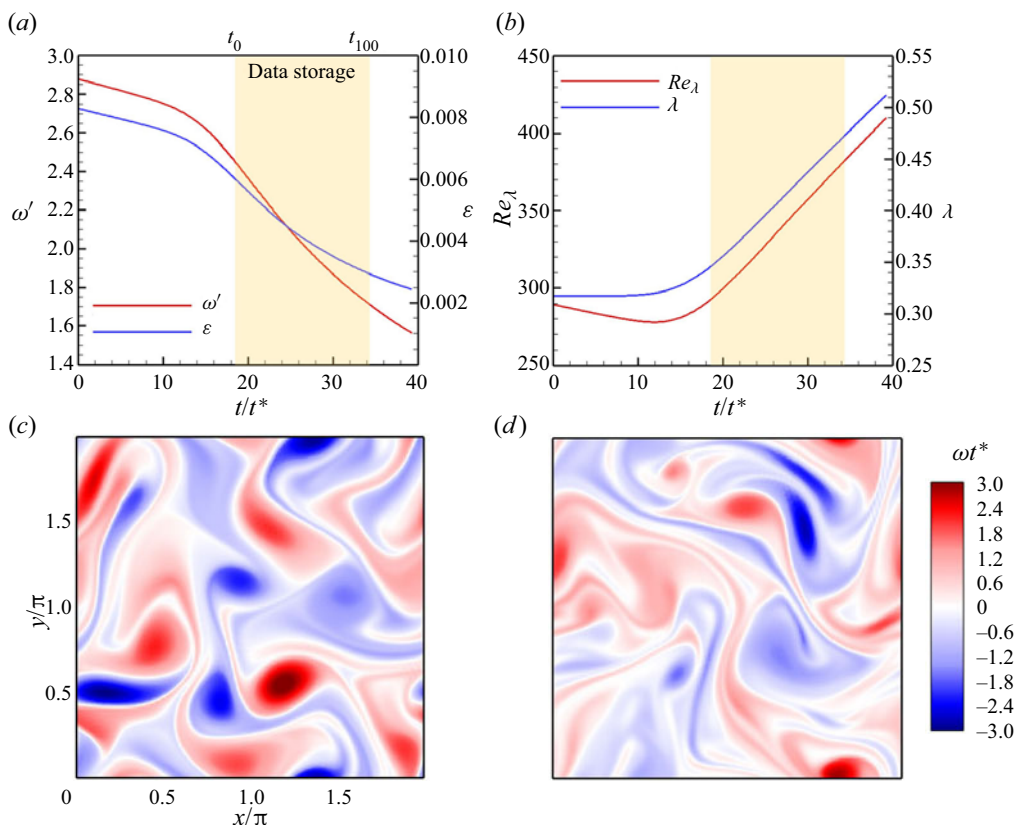


Figure 1. Range of selected data for training with a time interval of $100\delta t/t^*$ shown in the yellow region in (a) the vorticity r.m.s. and dissipation rate and in (b) Reynolds number and the Taylor length scale. Example vorticity fields at (c) t_0 and (d) t_{100} from a normalised test data.

and $\sim t^{-1.12}$, respectively, as shown in figure 1(a), whereas the Taylor length scale ($\lambda = u'/\omega'$ where u' is the r.m.s. velocity magnitude; Jiménez 2018) and the Reynolds number based on λ ($Re_\lambda = u'\lambda/\nu$), which are ensemble-averaged over 500 independent simulations, linearly increase as shown in figure 1(b). Therefore, 50 000 pairs of flow field snapshots were used in training, and for sufficient iterations of training without overfitting, data augmentation using a random phase shift at each iteration was adopted. Hereafter, the reference time and length scales in all non-dimensionalisations are $t^* = 1/\omega'_0$ and $\lambda^* = u'_0/\omega'_0$, respectively. The non-dimensionalised simulation and data time steps are $\Delta t/t^* = 0.00614$ and $\delta t/t^* = 0.123$, respectively. Here $t_{100} - t_0$ is approximately 2.5 times the Eulerian integral timescale of the vorticity field, as discussed in the following. Therefore, the vorticity fields at t_0 and t_{100} are decorrelated as shown in figure 1(c,d). In our training, all of these data spanning from t_0 to t_{100} were used without distinction such that the trained network covers diverse characteristics of decaying turbulence.

To select the target lead time T , we investigate the temporal autocorrelation function of the vorticity field, $\rho(s) (= \langle \omega(t)\omega(t+s) \rangle / \langle \omega(t)^2 \rangle^{1/2} \langle \omega(t+s)^2 \rangle^{1/2})$ for time lag s , as shown in figure 2(a), from which the integral time scale is obtained, $T_L = 4.53t^* = 36.9\delta t$. As it is meaningless to predict the flow field much later than one integral time scale, we selected four lead times to develop a prediction network: $10\delta t$, $20\delta t$, $40\delta t$ and

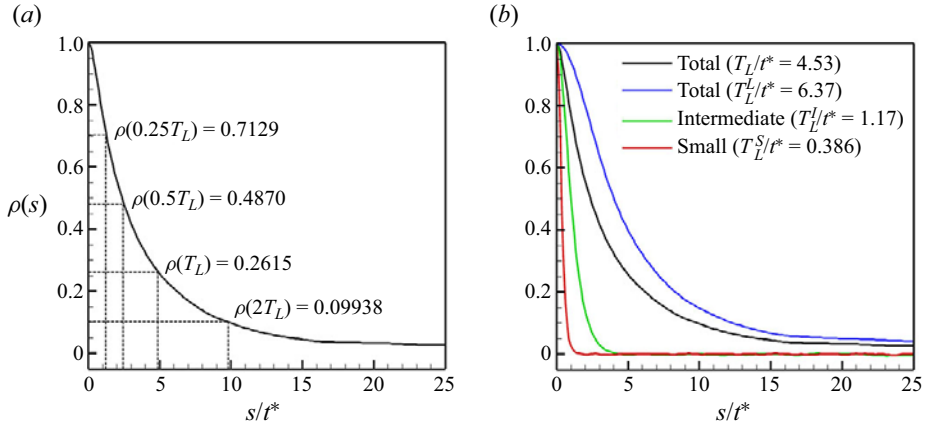


Figure 2. Distribution of the temporal autocorrelation function of (a) the whole vorticity field and (b) the scale-decomposed vorticity fields.

$80\delta t$, which are referred to as $0.25T_L$, $0.5T_L$, T_L and $2T_L$, respectively, even though $T_L = 36.9\delta t$. Figure 2(b) shows the correlation function for the scale-decomposed field of vorticity, where three decomposed fields are considered: a large-scale field consisting of the wavenumber components for $k \leq 4$, representing the energy (enstrophy)-containing range; an intermediate-scale field containing the inertial-range wavenumber components for $4 < k \leq 20$; and the small-scale field corresponding to the wavenumber components for $k > 20$ in the dissipation range. This clearly illustrates that the large-scale field persists longer than the total field with an integral time scale $T_L^L \simeq 1.4T_L$, whereas the intermediate- and small-scale fields quickly decorrelate with $T_L^I \simeq 0.25T_L$ and $T_L^S \simeq 0.09T_L$. These behaviours are responsible for the different prediction capabilities of each scale component, as discussed later.

The spatial two-point correlation function of vorticity $R_\omega(r, t)$ with the corresponding integral length scale L_t at three different times $t = t_0, t_0 + T_L$ and $t_0 + 2T_L$ is shown in figure 3. For the investigated time range, $R_\omega(r, t)$ decays sufficiently close to zero at $r = \pi$ (half domain length), even though L_t tends to increase over time from $0.876\lambda^*$ at t_0 to $1.03\lambda^*$ at t_{100} because of the suppression of the small-scale motions of 2-D turbulence. This marginally decaying nature in the periodic domain was considered in the design of the training network such that data at all grid points were used in the prediction of vorticity at one point, as discussed in the following section.

3. ML methodology

3.1. ML models and objective functions

Training ANNs is the process of updating weight parameters to satisfy the nonlinear relationships between the inputs and outputs as closely as possible. The weight parameters were optimised to minimise the prescribed loss function (in the direction opposite to the gradient) by reflecting nonlinear mappings. Loss functions are mainly determined by objective functions, and other loss terms are often added to improve training efficiency and model performance. In GANs, a generator (G) and discriminator (D) are trained simultaneously in an adversarial manner; parameters of G and D are iteratively and alternately updated to maximise $\log(1 - D(G(z)))$ for G and maximise $\log(D(x)) + \log(1 - D(G(z)))$ for D , respectively. This stands for the two-player min-max game with

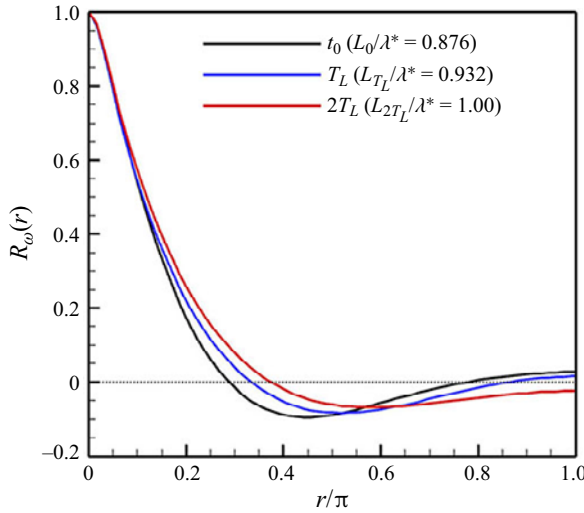


Figure 3. Ensemble-averaged two-point correlation functions of vorticity extracted from 500 training data.

a value function $V(G, D)$ given by

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p(x)}[\log D(x)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z)))] \tag{3.1}$$

where x and z are real data and random noise vectors, respectively. Operator \mathbb{E} denotes the expectation over some sampled data, and the expressions $x \sim p(x)$ and $z \sim p(z)$ indicate that x is sampled from the distribution of the real dataset $p(x)$ and z from some simple noise distribution $p(z)$ such as a Gaussian distribution, respectively. Thus, we can obtain a generator that produces more realistic images. Various GANs have been developed rapidly since their introduction (Mirza & Osindero 2014; Arjovsky, Chintala & Bottou 2017; Gulrajani *et al.* 2017; Karras *et al.* 2017; Mescheder, Geiger & Nowozin 2018; Park *et al.* 2019; Zhu *et al.* 2020). Among these, a conditional GAN (cGAN) (Mirza & Osindero 2014) provides additional information y , which can be any type of auxiliary information, as a condition for the input of the generator and discriminator to improve the output quality of the generator, as follows:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p(x)}[\log D(x | y)] + \mathbb{E}_{z \sim p(z)}[\log(1 - D(G(z | y)))] \tag{3.2}$$

Furthermore, there are two adaptive methods that can stabilise the training process, solve the problem of the vanishing gradient in which the discriminator is saturated, and prevent mode collapse, a phenomenon in which the distribution of generated samples is restricted to a specific small domain, even though the generator does not diverge. First, (3.2) is modified using the earth-mover (EM; or Wasserstein-1) distance combined with the Kantorovich–Rubinstein (KR) duality (Villani 2009), which is called Wasserstein GAN (WGAN) (Arjovsky *et al.* 2017), as follows:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p(x)}[D(x | y)] - \mathbb{E}_{z \sim p(z)}[D(G(z | y))] \tag{3.3}$$

This modification is made based on thorough examinations of various ways of measuring the distance between the real (data) distribution (p_r) and the model distribution (p_g),

including the total variation distance, Kullback–Leibler divergence and Jensen–Shannon divergence. EM distance can be expressed as the final form of (3.4) by the KR duality:

$$EM(p_r, p_g) = \sup_{\|f\|_L \leq 1} \mathbb{E}_{x \sim p_r}[f(x)] - \mathbb{E}_{x \sim p_g}[f(x)], \tag{3.4}$$

where the supremum is taken over all the 1-Lipschitz functions for the set of real data $\mathcal{X}, f : \mathcal{X} \rightarrow \mathbb{R}$. Simply put, the model g that wants to make p_g close to p_r represents the generator, and f corresponds to the discriminator that is optimised to make the distance between p_r and p_g larger. Thus, it can be melted down to the form of (3.3), and the discriminator output can be viewed as latent codes of real and generated data. Then, a gradient penalty (GP) loss term is added to obtain the final form of WGAN-GP (Gulrajani *et al.* 2017). We intend to develop a model capable of predicting the dynamic behaviour of turbulence with high predictive accuracy by reflecting statistical aspects, employing a cGAN with WGAN-GP for PredictionNet and comparing the results with a baseline CNN.

PredictionNet (the generator of cGAN) is a network that predicts the vorticity field after a specific lead time T from the input field at t as follows:

$$\text{Pred}(X(t)) = Y^* \approx X(t + T) = Y, \tag{3.5}$$

where X, Y^* and Y represent the real data, prediction results and prediction targets, respectively (here, the prediction target Y is independent of the additional input y in (3.2) and (3.3), which are general descriptions of the value function of each GAN model). Originally, the input of a generator for turbulence prediction would be a noise field, Z , conditioned on the previous timestep’s vorticity field from (3.3), expressed as $\text{Pred}(Z, X) = Y^*$. However, the desired prediction network is a deterministic mapping function along the time horizon T ; thus, the noise input is excluded in (3.5). In our cGAN application, the generator input (X) is also used as a condition in the training of the discriminator through concatenation. Therefore, the target Y and the generator output Y^* conditioned on X are used as inputs to the discriminator in the first and second terms of (3.3) as $D(Y, X)$ and $D(Y^*, X)$, respectively. This allows the statistical characteristics of the input (X) as well as the target (Y) to be reflected on the training of the discriminator and eventually the generator through competitive training. Moreover, the most easily imaginable and generally employed optimisation target for prediction task is the following objective function based on the pointwise difference:

$$\underset{w_p}{\text{argmin}} \|Y^* - Y\|, \tag{3.6}$$

with the trainable parameters of PredictionNet w_p , and $\|\cdot\|$ represents a distance norm of any type. This data loss can be a powerful regulariser for the generator in GANs at the beginning of training procedure. Therefore, we formulated the final form of the loss function of cGAN, including the GP and drift loss term, as follows:

$$L_G = \gamma \mathbb{E}_{X \sim p(X)} \left[\|Y^* - Y\|_2^2 \right] - L_{adv}, \quad L_{adv} = L_{false}, \tag{3.7}$$

for the generator and

$$L_D = -L_{true} + L_{false} + \alpha L_{gp} + \beta L_{drift}, \tag{3.8}$$

for the discriminator, where

$$\left. \begin{aligned} L_{true} &= \mathbb{E}_{X \sim p(X)} [D(Y, X)], & L_{false} &= \mathbb{E}_{X \sim p(X)} [D(Y^*, X)], \\ L_{gp} &= \mathbb{E}_{X \sim p(X)} \left[(\|\nabla_{Y'} D(Y', X)\|_2 - 1)^2 \right], & L_{drift} &= (\mathbb{E}_{X \sim p(X)} [D(Y, X)])^2, \end{aligned} \right\} \tag{3.9}$$

where $Y' = Y + \delta(Y^* - Y)$ with δ between zero and one. The simplest L2 norm distance was used for the data loss. The role of L_{drift} was to restrict the order of discriminator outputs (keeping them from drifting too far from zero) with a small weight β . Its original form in Karras *et al.* (2017) is similar to the L2 regularisation on $D(Y, X)$ as $L_{drift} = \mathbb{E}_{X \sim p(X)} [(D(Y, X))^2]$, but we modified it to the above form, in which regularisation can be applied average-wise during the backpropagation to robustly use hyperparameters α and β regardless of the lead time. Accordingly, $\beta = 0.001$ was used equally for all lead times, and α and γ were fixed at 10 and $100/(N_x \times N_y)$, respectively, by fine tuning. There exists a separate optimum hyperparameter setting for each lead time; however, we verified that our hyperparameter setting showed no significant difference in performance from the optimum settings. In addition, we verified that it worked properly for lead times ranging from $1\delta t$ to $100\delta t$. On the other hand, the baseline CNN is an identical network to PredictionNet without the adversarial learning. For the loss function of it, an L2 regularisation was added to (3.6) using the L2 norm distance to improve the performance and make the optimisation process efficient, as follows:

$$L_{CNN} = \sigma_1 \mathbb{E}_{X \sim p(X)} \left[\|Y^* - Y\|_2^2 \right] + \sigma_2 R(w_p), \quad R(w_p) = \frac{1}{2} \|w_p\|_2^2, \quad (3.10)$$

where σ_1 is set to $1/(N_x \times N_y)$, and σ_2 modulates the strength of regularisation and is fixed at 0.0001 based on case studies. Simplified configurations of the CNN and cGAN are shown in figure 4(a,b).

ControlNet uses a pre-trained PredictionNet that contains the dynamics of our data as a surrogate model to generate disturbance fields that change the evolution of the flow field in a direction suitable for a target, as follows:

$$\text{Control}(X(t)) = \Delta X, \quad \text{Pred}(X(t) + \Delta X) = \tilde{Y}. \quad (3.11)$$

Here ΔX represents the disturbance field added to the input field $X(t)$ and \tilde{Y} is the disturbance-added prediction at time $t + T$. An important implication of ControlNet in this study is that it is a model-free method without restrictions, except for the strength of ΔX . The objective function to be maximised includes the change in the vorticity field at a later time, as follows:

$$\operatorname{argmax}_{w_c} \|Y^* - \tilde{Y}\|, \quad (3.12)$$

where w_c are the weight parameters of ControlNet. In the process of training of ControlNet, the weight parameters of ControlNet are updated in the direction maximising the change in the vorticity field at the target lead time. The already trained PredictionNet with fixed weight parameters is used in the prediction of controlled field as a surrogate model in training of ControlNet. Therefore, once the training of ControlNet is successful through maximisation of the loss based on the change in the vorticity field, the trained ControlNet can produce an optimum disturbance field. Whether the generated disturbance field is globally optimum, however, is not guaranteed. For the final form of the loss function, a spectral gradient loss term is additionally used to remove nonphysical noise (i.e. smoothing effect), and a CNN model is applied:

$$L_{control} = -\mathbb{E}_{X \sim p(X)} \left[\|Y^* - \tilde{Y}\|_2^2 \right] + \theta L_{grad}, \quad L_{grad} = \mathbb{E}_{X \sim p(X)} \left[\|\nabla_x (\Delta X)\|_2^2 \right], \quad (3.13)$$

where ∇_x denotes the gradient with respect to the coordinate directions. The coefficient θ controls the strength of smoothing effect, and it is adjusted depending on the order of data loss (see § 4.4). Figure 4(c) shows a simplified configuration of ControlNet.

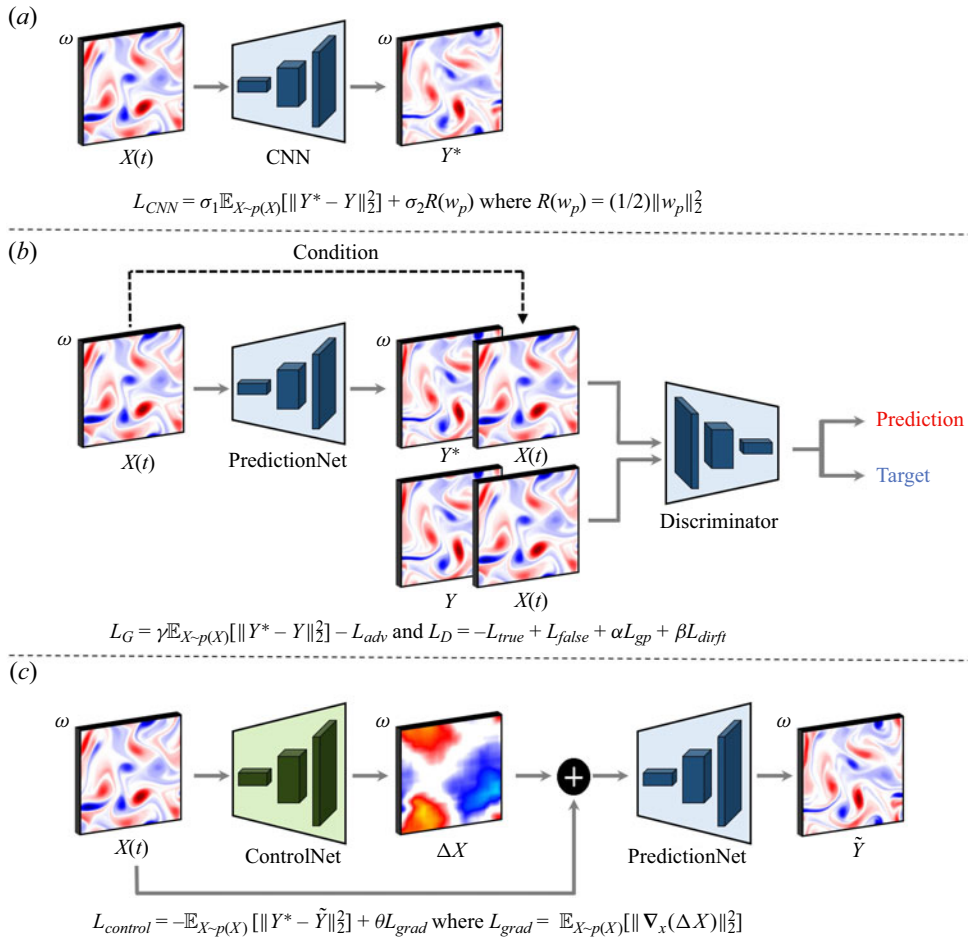


Figure 4. Simplified network schematics of (a) the baseline CNN, (b) cGAN-based PredictionNet and (c) ControlNet.

3.2. Network architecture

A typical multiscale architecture of a CNN was used for both PredictionNet and ControlNet, as shown in figure 5(a). The architecture consists of six convolutional blocks, each composed of three convolutional layers with 3×3 filter kernels (Conv. 3×3), called Convblk- m , named after their feature maps with different resolutions ($128/2^m \times 128/2^m$ ($m = 0, 1, 2, 3, 4, 5$)). Here, the average pooled inputs $X^{(m)}$ ($m = 1, 2, 3, 4$) and the original input $X^{(0)}$ are concatenated to the feature map tensor at the beginning of each Convblk- m to be used as feature maps. One node point of a pooled input $X^{(m)}$ contains the compressed information of 2^m points of $X^{(0)}$. Using such an architecture enables all the spatial grid information of an input field, X , to be used to calculate a specific output point, even for the lowest-resolution feature maps. For example, the receptive field of Convblk-5 is $(2^5 \times 7) \times (2^5 \times 7)$, which means that all the spatial grid points of $X^{(0)}$ are used to calculate a point in the output. In addition, by sequentially using low- to high-resolution feature maps, the network can learn large-scale motion in the early stages, and then fill in the detailed features of small-scale characteristics in the deepest layers. As mentioned

Prediction and control of 2-D decaying turbulence using GANs

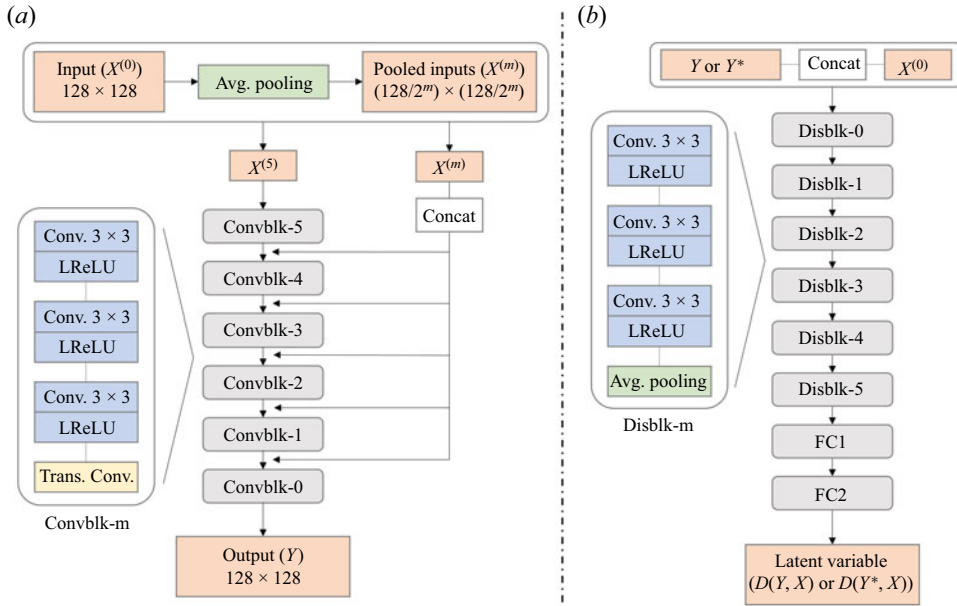


Figure 5. Network architecture of (a) PredictionNet (the generator of cGAN) and ControlNet and (b) discriminator of cGAN.

in § 2, the purpose of designing networks is to increase the prediction accuracy and determine the global optimum by considering the spectral properties of the flow. However, depending on the flow features or characteristics of the flow variables, we can expect that accurate results can be obtained even using only a small input domain with a size similar to L for cost-effectiveness. Furthermore, periodic padding was used to maintain the size of the feature maps after convolutional operations, because the bi-periodic boundary condition was applied spatially. The feature maps generated from Convblk-($m+1$) must be extended through upscaling to be concatenated and used with the following pooled input $X^{(m)}$ (i.e. doubling the size via upsampling). In this process, a transposed convolution is used to minimise the non-physical phenomena. After upscaling, $X^{(m)}$ is concatenated to the feature map tensor and then Convblk- m operations are performed. Finally, after the original input field $X^{(0)}$ is concatenated, the operations of Convblk-0 are performed, and the output of resolution 128×128 is generated depending on the network, prediction result $\text{Pred}(X) = Y^*$, or disturbance field $\text{Control}(X) = \Delta X$. Detailed information on the architectures of PredictionNet and ControlNet, including the number of feature maps used in each Convblk, are presented in table 1. A leaky rectified linear unit (LReLU) is used as an activation function to add nonlinearities after each convolutional layer. This was not applied to the last layer of Convblk-0, to prevent nonlinearity in the final output.

The discriminator of PredictionNet to which cGAN is applied has a symmetric architecture for the generator (PredictionNet), as shown in figure 5(b). Here X^0 is concatenated with the target or the prediction result for the input of the discriminator as a condition. In contrast to the generator, convolutional operations are performed from high to low resolutions, named with each convolutional block as Disblk- m . Average pooling was used for downsampling to half the size of the feature maps. After the operation of Disblk-5, its output feature maps passed through two fully connected layers (with output dimensions of $1 \times 1 \times 256$ and $1 \times 1 \times 1$) to return a scalar value. The numbers of feature maps used for each Disblk are presented in table 1. The baseline CNN model takes the

Convblk-m	Resolution	Number of feature maps	Disblk-m	Resolution	Number of feature maps
Convblk-5	4×4	64, 64, 64	Disblk-0	128×128	16, 16, 32
Convblk-4	8×8	64, 64, 64	Disblk-1	64×64	32, 32, 64
Convblk-3	16×16	64, 64, 64	Disblk-2	32×32	64, 64, 64
Convblk-2	32×32	64, 64, 64	Disblk-3	16×16	64, 64, 64
Convblk-1	64×64	32, 32, 32	Disblk-4	8×8	64, 64, 64
Convblk-0	128×128	16, 16, 1	Disblk-5	4×4	64, 64, 64

Table 1. Number of feature maps used at each convolutional layer of Convblks and Disblks.

same architecture as PredictionNet but is trained without adversarial training through the discriminator.

4. Results

4.1. PredictionNet: prediction of the vorticity field at a finite lead time

In this section, we discuss the performance of PredictionNet in predicting the target vorticity field at $t + T$ using the field at t as the input. The considered lead times T are $0.25T_L$, $0.5T_L$, T_L and $2T_L$ based on the observation that the autocorrelation of the vorticity at each lead time dropped to 0.71, 0.49, 0.26 and 0.10, respectively (figure 2a). The convergence behaviour of the L2 norm-based data loss of the baseline CNN and cGAN models for various lead times in the process of training is shown for 100 000 iterations with a batch size of 32 in figure 6. It took around 2 and 5.4 h for CNN and cGAN, respectively, on a GPU machine of NVIDIA GeForce RTX 3090. Although the baseline CNN and PredictionNet have the exact same architecture (around 520 000 trainable parameters based on table 1), cGAN includes a discriminator to be trained comparable to the generator with many complex loss terms; thus, the training time is almost tripled. Both models yielded good convergence for lead times of up to $0.5T_L$, whereas overfitting was observed for $T = T_L$ and $2T_L$ in both models. Since the flow field is hardly correlated with the input field [correlation coefficient (CC) of 0.26 and 0.10], it is naturally more challenging to train the model that can reduce a mean-squared error (MSE)-based data loss, a pointwise metric that is independent of the spatial distribution. One notable observation in figure 6 is that for lead times beyond T_L , CNN appears to exhibit better performance than cGAN, as evidenced by its smaller converged data loss. However, as discussed later, the pointwise accuracy alone cannot encompass all the aspects of turbulence prediction due to its various statistical properties. In addition, although the only loss term of CNN is the pointwise MSE except the weight regularisation, the magnitude of its converged data loss also remains significant.

As an example of the test of the trained network, for unseen input data, the fields predicted by cGAN and CNN were compared against the target data for various lead times, as shown in figure 7. In the test of cGAN, only PredictionNet was used to generate or predict flow field at the target lead time. Hereinafter, when presenting performance results for the instantaneous field, including figure 7 and all subsequent statistics, we only brought the results for input at t_0 . This choice was made because the difficulty of tasks that the models have to learn is much larger at earlier time since flow contains more small-scale structures than later times due to decaying nature. Therefore, performance test for input data at t_0 is sufficient for test of the trained network. For short lead-time predictions, such as $0.25T_L$, both cGAN and CNN showed good performance. However, for a lead time

Prediction and control of 2-D decaying turbulence using GANs

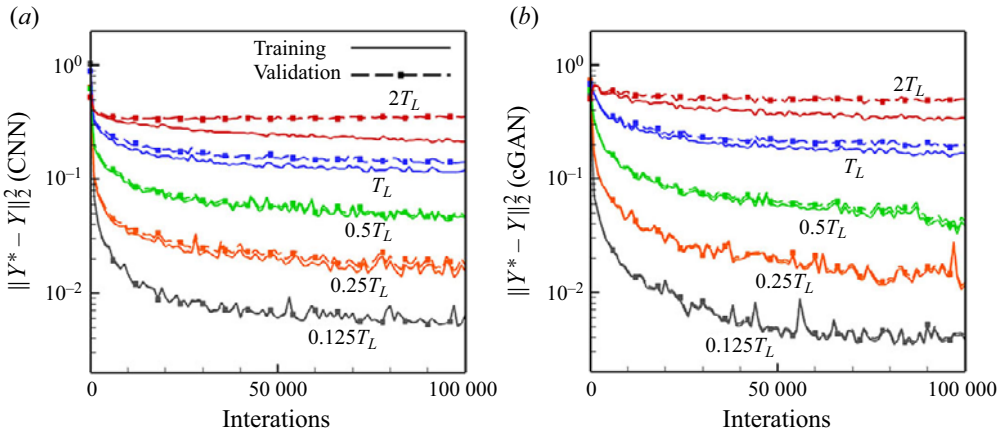


Figure 6. Convergence of data loss depending on the lead time of (a) baseline CNN and (b) cGAN. The order of the converged value increases as the lead time increases. In addition, relatively large overfitting was observed in the case of $T = 2T_L$.

of $0.5T_L$, the prediction by the CNN started displaying a blurry image, whereas cGAN’s prediction maintained small-scale features relatively better than CNN. This blurry nature of the CNN worsened the predictions of T_L and $2T_L$. Although cGAN’s prediction also deteriorated as the lead time increased, the overall performance of cGAN, particularly in capturing small-scale variations in the field, was much better than that of the CNN even for T_L and $2T_L$.

A quantitative comparison of the performances of the cGAN and CNN against the target in the prediction of various statistics is presented in table 2. The CC between the prediction and target fields or, equivalently, the MSE by the cGAN shows a better performance than that by the CNN for lead times of up to $0.5T_L$, even though both the cGAN and CNN predicted the target field well. For T_L , the CC by cGAN and CNN was 0.829 and 0.868, respectively, indicating that the predictions by both models were not poor, whereas the predictions by both models were quite poor for $2T_L$. Once again, for lead times larger than T_L , CNN shows better performance on the pointwise metrics according to the trend of data loss. Conversely, the statistics related to the distribution of vorticity, such as the r.m.s. value or standardised moments ($\hat{\mu}_n = \mu_n/\sigma^n$, where $\mu_n = \langle (X - \langle X \rangle)^n \rangle$ is the n th central moments and $\sigma = \sqrt{\mu_2}$ is the standard deviation), clearly confirm that the cGAN outperforms the CNN for all lead times, and the prediction by the cGAN was much closer to the target than that of the CNN, even for T_L and $2T_L$. Furthermore, the prediction of the Kolmogorov length scale ($\eta = (v^3/\varepsilon)^{1/4}$) and dissipation ($\varepsilon = 2\nu \langle S_{ij}S_{ij} \rangle$, $\varepsilon' = \varepsilon t^{*3}/\lambda^{*2}$, where S_{ij} is the strain rate tensor) by cGAN was more accurate than that by the CNN, as presented in table 2. All these qualitative and quantitative comparisons clearly suggest that the adversarial learning of the cGAN tends to capture the characteristics of turbulence data better than the CNN, which minimises the pointwise MSE only. The superiority of the cGAN is more pronounced for large lead times, for which small-scale turbulence features are difficult to predict.

Detailed distributions of probability density functions (p.d.f.s) of vorticity, two-point correlation functions and enstrophy spectra ($\Omega(k) = \pi k \sum_{|k|=k} |\hat{\omega}(k)|^2$ where $\hat{\omega}(k, t) = \mathcal{F}\{\omega(x, t)\}$) of the target and prediction fields by the cGAN and CNN are compared for all lead times in figure 8. Both the cGAN and CNN effectively predicted the target p.d.f. distribution for lead times of up to T_L , whereas the tail part of the p.d.f. for $2T_L$

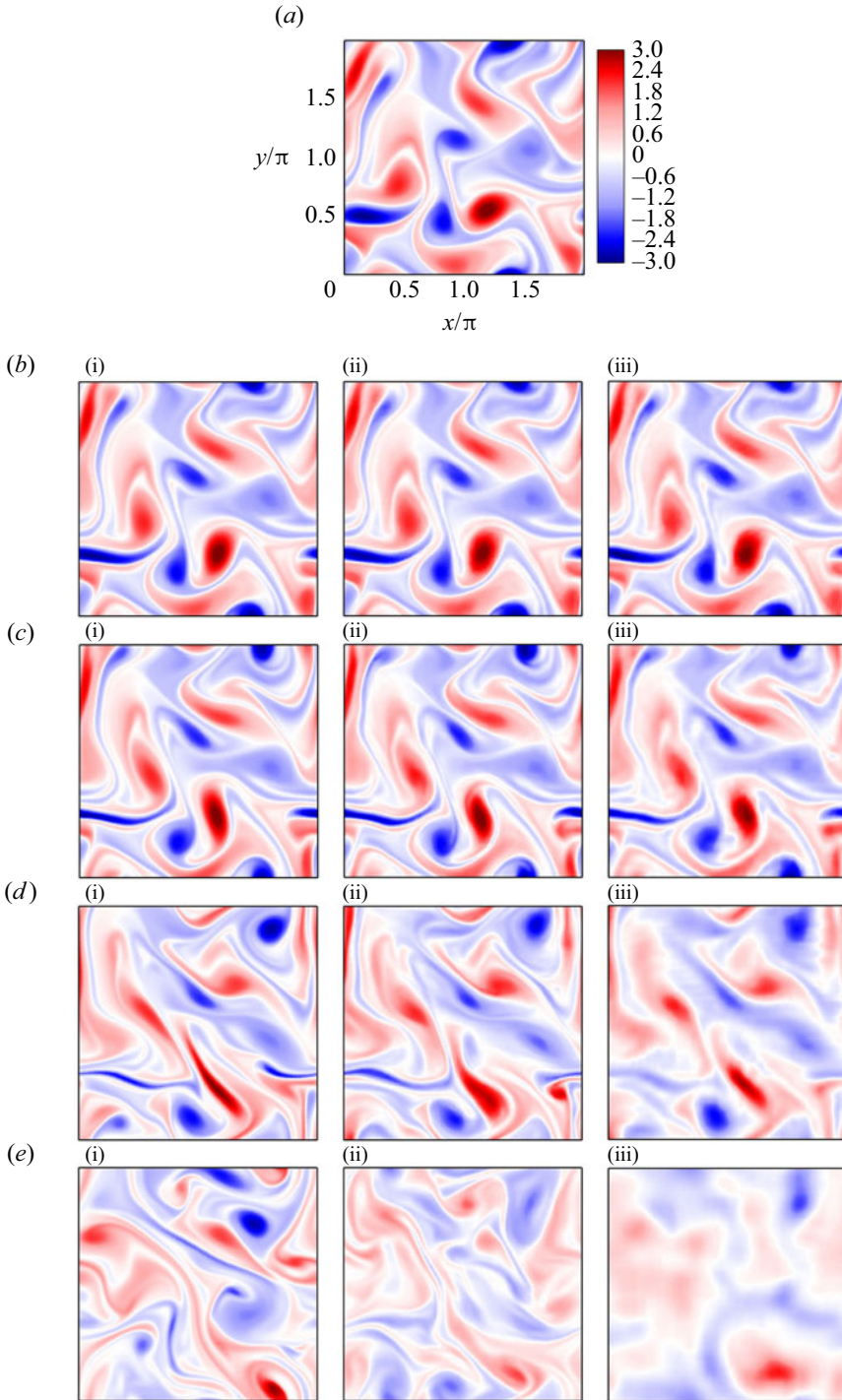


Figure 7. Visualised examples of the performance of the cGAN and CNN for one test dataset. (a) Input field at t_0 , prediction results at the lead time (b) $0.25T_L$, (c) $0.5T_L$, (d) T_L and (e) $2T_L$. Panels (b i,c,i,d,i,e), (b ii,c ii,d ii,e ii) and (b iii,c iii,d iii,e iii) show the target DNS, cGAN predictions and CNN predictions, respectively.

Lead time T ($\rho(T)$)	CC	MSE	r.m.s.	$\hat{\mu}_4$	$\hat{\mu}_6$	$\hat{\mu}_8$	$\hat{\mu}_{10}$	η/λ^*	$\varepsilon' (\times 10^{-2})$
0.125 T_L (0.8774)	Target	—	0.9871	2.963	13.70	80.34	543.3	0.1443	2.001
	cGAN	3.32×10^{-3}	0.9813	2.957	13.64	79.82	539.0	0.1446	1.978
	CNN	6.05×10^{-3}	0.9813	2.961	13.67	79.95	539.1	0.1447	1.978
0.25 T_L (0.7129)	Target	—	0.9728	3.001	14.15	84.87	587.8	0.1453	1.945
	cGAN	1.09×10^{-2}	0.9643	3.016	14.36	87.29	614.7	0.1459	1.911
	CNN	1.85×10^{-2}	0.9603	3.019	14.42	88.25	627.6	0.1463	1.895
0.5 T_L (0.4870)	Target	—	0.9440	3.084	15.14	95.05	690.5	0.1475	1.832
	cGAN	0.0537	0.9357	3.108	15.56	101.2	776.2	0.1482	1.800
	CNN	0.0557	0.8917	3.152	16.05	105.5	806.6	0.1518	1.636
T_L (0.2615)	Target	—	0.8877	3.268	17.46	120.6	966.9	0.1522	1.621
	cGAN	0.262	0.8603	3.261	17.40	121.5	1010	0.1546	1.523
	CNN	0.196	0.7946	3.413	19.49	146.8	1307	0.1610	1.301
2 T_L (0.09938)	Target	—	0.7883	3.648	22.90	189.1	1827	0.1615	1.279
	cGAN	0.708	0.6976	3.569	22.15	189.2	2000	0.1717	1.002
	CNN	0.562	0.5795	4.064	29.73	303.4	3865	0.1889	0.696

Table 2. Quantitative comparison of the performance of the cGAN and CNN against the target in the prediction of the CC, MSE and r.m.s., and the n th standardised moments, Kolmogorov length scale (η) and dissipation rate (ε') depending on the lead time for the input at t_0 . All the dimensional data such as MSE, r.m.s. and ε' are provided just for the relative comparison.

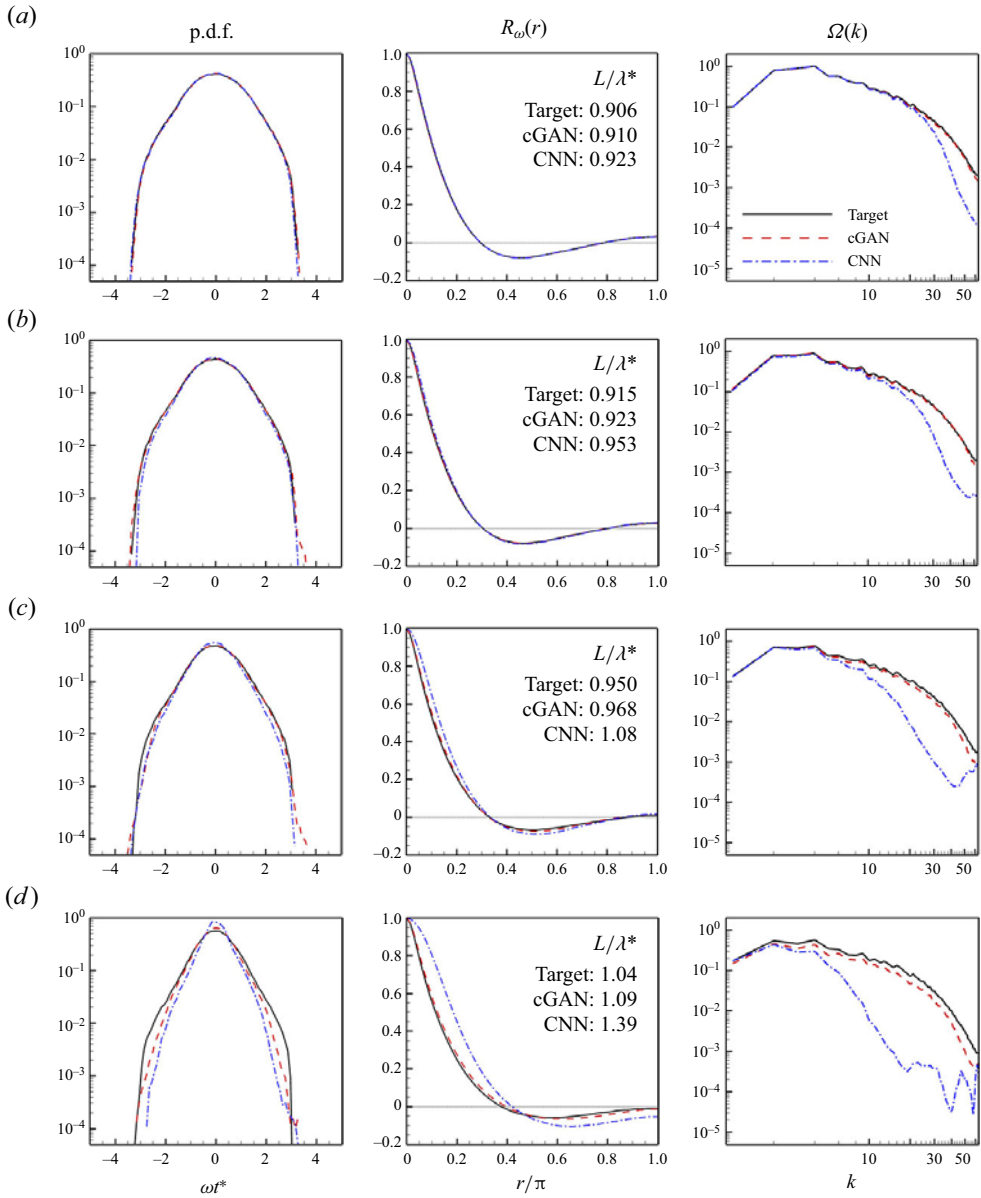


Figure 8. Comparison of statistics such as the p.d.f., two-point correlation and entropy spectrum of the target and prediction results at different lead times (a) $0.25T_L$, (b) $0.5T_L$, (c) T_L and (d) $2T_L$ for the same input time point t_0 .

was not effectively captured by the cGAN and CNN. The difference in performance between the cGAN and CNN is more striking in the prediction of $R_\omega(r)$ and $\Omega(k)$. The values of $R_\omega(r)$ and $\Omega(k)$ obtained by the cGAN almost perfectly match those of the target for all lead times, whereas those predicted by the CNN show large deviations from those of the target distribution progressively with the lead time. In the prediction of the correlation function, the CNN tends to produce a relatively slow decaying distribution compared with the cGAN and the target, resulting in a much larger integral length scale

for all lead times. In particular, it is noticeable that even for a short lead time of $0.25T_L$, the CNN significantly underpredicts $\Omega(k)$ in the high-wavenumber range, and this poor predictability propagates toward the small-wavenumber range as the lead time increases, whereas cGAN produces excellent predictions over all scale ranges, even for $2T_L$. This evidence supports the idea that adversarial learning accurately captures the small-scale statistical features of turbulence.

To quantify in more detail the differences in the prediction results by cGAN and CNN models, scale decomposition was performed by decomposing the fields in the wavenumber components into three regimes: large-, intermediate- and small-scale fields, as in the investigation of the temporal correlation in [figure 2\(b\)](#). The decomposed fields predicted for the lead time of T_L by the cGAN and CNN, for example, were compared with those of the target field, as shown in [figure 9](#). As shown in [figure 2\(b\)](#), the large-scale field ($k \leq 4$) persists longer than the total fields with an integral time scale 1.4 times larger than that of the total field, whereas the intermediate-scale field ($4 < k \leq 20$) and the small-scale field ($20 < k$) decorrelate more quickly than the total field with integral time scales one-fourth and one-twelfth of that of the total field, respectively. Given that the intermediate- and small-scale fields of the target field are almost completely decorrelated from the initial field at a lead time of T_L as shown in [figure 2\(b\)](#), the predictions of those fields by the cGAN shown in [figure 9\(c,d\)](#) are excellent. The cGAN generator generates a small-scale field that not only mimics the statistical characteristics of the target turbulence but is also consistent with the large-scale motion of turbulence through adversarial learning. A comparison of the decomposed fields predicted by the cGAN and CNN for other lead times is presented in [Appendix B](#). For small lead times, such as $0.25T_L$ and $0.5T_L$, the cGAN predicted the small-scale fields accurately, whereas those produced by the CNN contained non-negligible errors. For $2T_L$, it is difficult to predict using both models even with the large-scale field. On the other hand, the CC of decomposed fields between the target and predictions, provided in [table 4 \(Appendix B\)](#), did not demonstrate the superiority of the cGAN over the CNN. This is attributed to the fact that pointwise errors such as the MSE or CC are predominantly determined by the behaviour of large-scale motions. The pointwise MSE is the only error used in the loss function of the CNN, whereas the discriminator loss based on the latent variable is used in the loss function of the cGAN in addition to the MSE. This indicates that the latent variable plays an important role in predicting turbulent fields with multiscale characteristics.

In the prediction of the enstrophy spectrum, the cGAN showed excellent performance, compared with the CNN, by accurately reproducing the decaying behaviour of the spectrum in the small-scale range, as shown in [figure 8](#). The average phase error between the target and predictions by the cGAN and CNN is shown in [figure 10](#). For all lead times, the phase error of the small-scale motion approaches $\pi/2$, which is the value for a random distribution. For short lead times $0.25T_L$ and $0.5T_L$, the cGAN clearly suppressed the phase error in the small-scale range compared with the CNN, whereas for T_L and $2T_L$, both the cGAN and CNN poorly predicted the phase of the intermediate- and small-scale motions, even though the cGAN outperformed the CNN in predicting the spectrum.

The performance of PredictionNet in the prediction of velocity fields is presented in [Appendix C](#), where several statistics, including the p.d.f. of velocity, two-point correlation and p.d.f. of the velocity gradient are accurately predicted for all lead times. The superiority of the cGAN over the CNN was also confirmed.

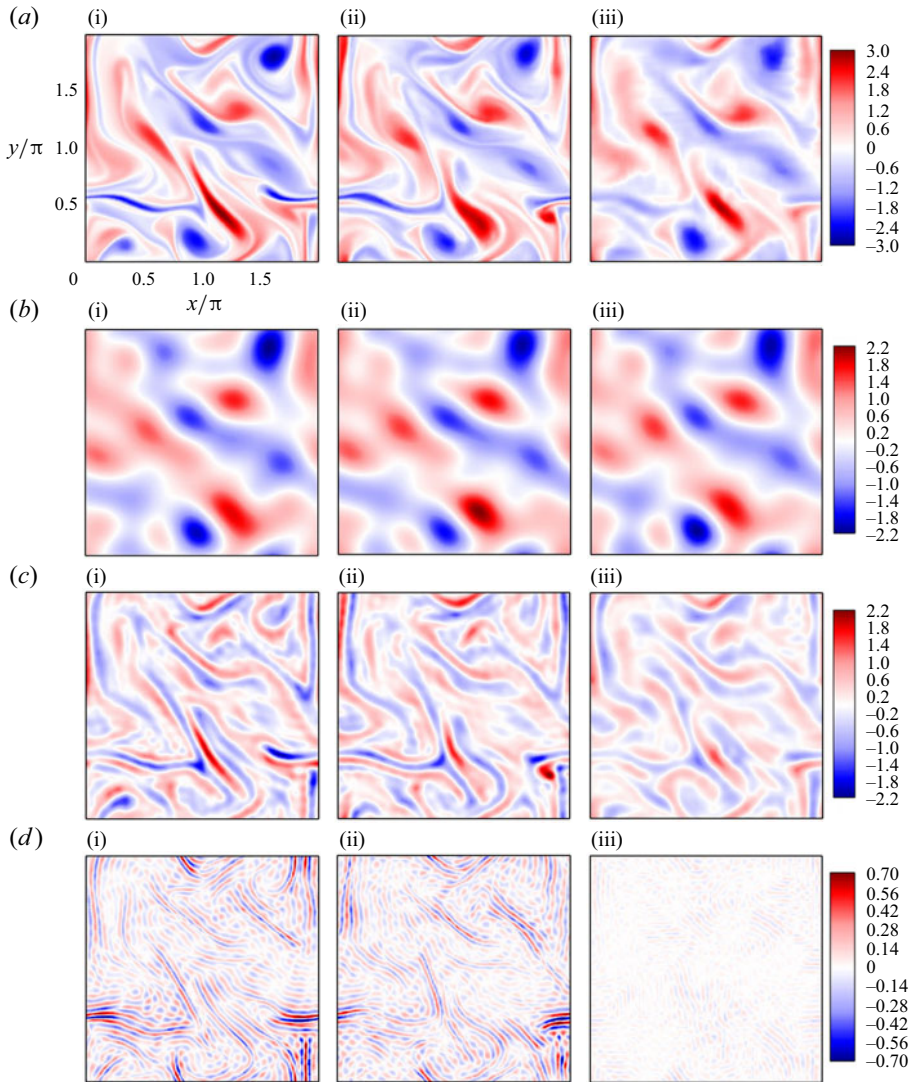


Figure 9. Prediction results of the scale-decomposed field for lead time T_L : (a) original fields, (b) large scales ($k \leq 4$), (c) intermediate scales ($4 < k \leq 20$) and (d) small scales ($k > 20$). Panels (a i,b i,c i,d i), (a ii,b ii,c ii,d ii) and (a iii,b iii,c iii,d iii) display the target DNS fields, cGAN predictions and CNN predictions, respectively.

4.2. Role of the discriminator in turbulence prediction

In the previous section, we demonstrated that adversarial learning using a discriminator network effectively captured the small-scale behaviour of turbulence, even when the small-scale field at a large lead time was hardly correlated with that of the initial field. In this section, we investigate the role of the discriminator in predicting turbulence through the behaviour of the latent variable, which is the output of the discriminator. Although several attempts have been made to disclose the manner in which adversarial training affects the performance of the generator and the meaning of the discriminator output (Creswell *et al.* 2018; Yuan *et al.* 2019; Goodfellow *et al.* 2020), there have been no

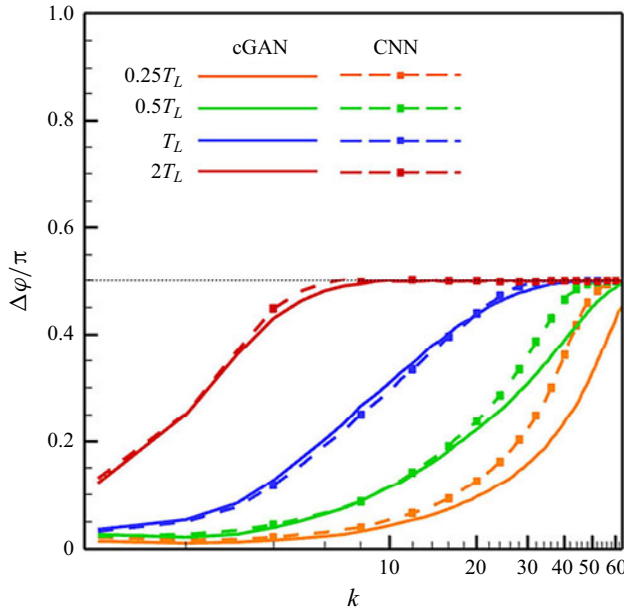


Figure 10. Prediction results of the phase error of each model depending on T .

attempts to reveal the implicit role of the latent variable in recovering the statistical nature of turbulence in the process of a prediction.

In the current cGAN framework, the discriminator network is trained to maximise the difference between the expected value of the latent variable of the target field $D(Y, X)$ and that of the prediction $D(Y^*, X)$, whereas the generator network is trained to maximise $D(Y^*, X)$ and to minimise the pointwise MSE between Y and Y^* . Therefore, through the adversarial learning process, the discriminator is optimised to distinguish Y^* from Y , whereas the generator evolves to produce Y^* by reflecting on the optimised latent variable of the discriminator and minimising the MSE between Y and Y^* . The input field X is used as a condition in the construction of the optimal latent variable in the discriminator network. The behaviour of the expected value of the latent variable of the target and generated fields (L_{true} and L_{false}) and their difference ($L_{true} - L_{false}$) as learning proceeds for all lead times are illustrated in figure 11. Because of the regularisation of the latent variable by $L_{drift} (= L_{true}^2)$ in the loss function of the discriminator, both L_{true} and L_{false} remain around zero, although they sometimes oscillate significantly. As the learning proceeds, $L_{true} - L_{false}$ quickly decays initially owing to the suppression of the MSE and continues to decrease to zero for $0.25T_L$ and $0.5T_L$ but to a finite value for T_L and $2T_L$. The intermittent transition of $L_{true} - L_{false}$ between a finite value and zero for $0.25T_L$ and $0.5T_L$ clearly suggests that the generator and discriminator operate competitively in an adversarial manner. As the generator improves, the difference monotonically decreases and occasionally exhibits sudden suppression. When sudden suppression occurs, the discriminator evolves to distinguish Y^* from Y by finding a better criterion latent variable, resulting in a sudden increase in the difference. Ultimately, when the generator can produce Y^* that is indistinguishable from Y by any criterion proposed by the discriminator, the difference converges to zero, as shown in the cases of $0.25T_L$ and $0.5T_L$. However, for T_L and $2T_L$, such an event never occurs; $L_{true} - L_{false}$ monotonically decreases and tends toward a finite value, implying that the discriminator wins and can distinguish Y^* from Y .

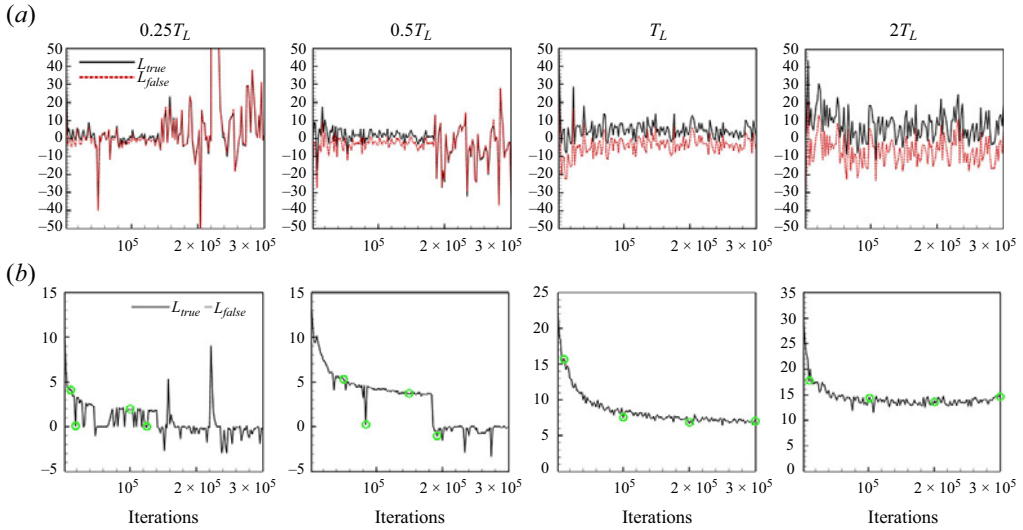


Figure 11. Evolution over training iteration of (a) L_{true} and L_{false} and (b) their difference evaluated every 2000 iterations.

Although the generator cannot produce Y^* to beat the discriminator, Y^* retains the best possible prediction.

To understand the mechanism in more detail that the discriminator uses to distinguish a generated field from a target field, the behaviour of the distribution of the latent variable during the learning process was investigated, because the latent variable plays a key role in the discriminator network. The distribution of the discriminator output of the generated field $D(Y^*, X)$ at four iterations during the learning process is marked with a green circle in figure 11(b) for all lead times and is compared with that of the target field $D(Y, X)$ in figure 12. The distributions of latent variable of the scale-decomposed target field $D(Y^S, X)$, $D(Y^I, X)$ and $D(Y^L, X)$ are also compared for analysis, where Y^S , Y^I and Y^L are scale-decomposed from Y in the same manner, as shown in figure 2. Here, an additional 500 fields, in addition to the 50 test fields, were used to extract a smooth distribution of the latent variables of the target field and scale-decomposed target fields. For easy comparison, the mean value of the latent variable for the target field was shifted to zero.

For all lead times, as learning proceeded, the distributions of $D(Y, X)$ and $D(Y^*, X)$ became narrower, with the mean values of $D(Y, X)$ and $D(Y^*, X)$ becoming closer to each other (the gap between the two vertical lines decreases). This indicates that, as the generator improves in producing Y^* closer to Y , the discriminator becomes more efficient in distinguishing Y^* from Y because only the mean values are compared in the discrimination process and the narrower distributions yield the sharper mean values. Even when the mean values of $D(Y, X)$ and $D(Y^*, X)$ are almost the same (18 000 and 124 000 iterations for $0.25T_L$ and 84 000 and 190 000 iterations for $0.5T_L$; see figure 12a,b), a more optimal discriminator at later iterations yields a narrower distribution. The collapse of the distribution of $D(Y, X)$ and $D(Y^*, X)$ in the second column for $0.25T_L$ and $0.5T_L$ occurs when the discriminator falls into a local optimum during training, implying that the predicted and target fields are indistinguishable by the discrimination criterion right at that iteration. However, as the criterion is jittered in the next iterations, the two fields become distinct again as shown in the third column. Eventually, when the two fields become indistinguishable by any criterion after sufficient iterations, almost perfect collapse of very

Prediction and control of 2-D decaying turbulence using GANs

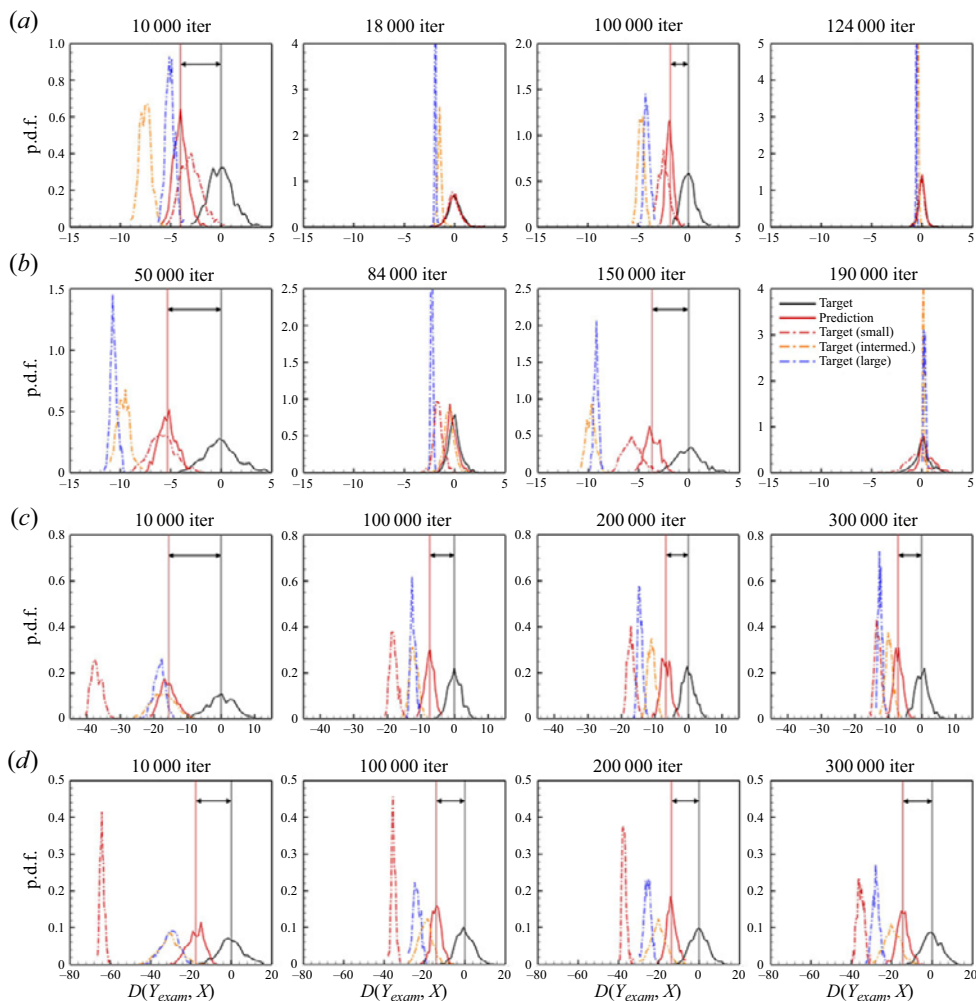


Figure 12. Distributions of discriminator output (latent variable) for various fields at several iterations: (a) $T = 0.25T_L$, (b) $T = 0.5T_L$, (c) $T = T_L$ and (d) $T = 2T_L$. All distributions are shifted by the target mean to fix the target distribution at the zero mean. The vertical black solid line indicates the target mean (zero) and the red line is the prediction mean. When the mean difference between the target and prediction is smaller than 0.5, the vertical lines are omitted for visibility.

narrow distributions is achieved as shown in the fourth column for $0.25T_L$ (the collapse shown in the fourth column for $0.5T_L$ occurs in another local optimum). For T_L and $2T_L$, for which perfect training of the discriminator was not achieved, however, distributions of $D(Y, X)$ and $D(Y^*, X)$ hardly change with iteration although the mean values are getting closer to each other very slowly.

In the comparison with the scale-decomposed target field, we observe that for $0.25T_L$ and $0.5T_L$, the distribution of the latent variable of the small-scale target field among all the scale-decomposed target fields is closest to that of the target field and generated field, whereas that of the intermediate-scale target field is closest to that of the target and generated fields for T_L and $2T_L$. This suggests that a small-scale (or intermediate) field in the target field plays the most critical role in discriminating the generated field from the target field for $0.25T_L$ and $0.5T_L$ (or T_L and $2T_L$). If the generator successfully produces

Y^* similar to Y for $0.25T_L$ and $0.5T_L$, and the predictions for T_L and $2T_L$ are incomplete, the small-scale fields for T_L and $2T_L$ are useless for discriminating the generated field from the target field, and the intermediate-scale field is partially useful. Considering the possible distribution of the functional form of the latent variable may provide a clue to this conjecture. The latent variable, which is the output of the discriminator network shown in [figure 5\(b\)](#), and a function of the input fields Y (or Y^*) and X , can be written as

$$D(Y, X) = \sum_{k_x, k_y} \hat{D}^Y(k_x \Delta x, k_y \Delta y; w_d, \epsilon_L) \hat{Y}(k_x, k_y) + \sum_{k_x, k_y} \hat{D}^X(k_x \Delta x, k_y \Delta y; w_d, \epsilon_L) \hat{X}(k_x, k_y), \tag{4.1}$$

where \hat{Y} and \hat{X} are the Fourier coefficients of Y and X , respectively. Here Δx , Δy , w_d and ϵ_L are the grid sizes in each direction, weights of the discriminator network and slope of the LReLU function for the negative input, respectively. This linear relationship between the input and output is a consequence of the linear structure of the discriminator network: the convolution, average pooling, and full connection operations are linear, and the LReLU function is a piecewise linear function such that either one or ϵ_L is multiplied by the function argument, depending on its sign. Although \hat{D}^Y is an undetermined function, a possible shape can be conjectured. For $0.25T_L$ and $0.5T_L$, \hat{D}^Y of the optimised discriminator network has more weight in the small-scale range than in other scale ranges such that the latent variable is more sensitive to the small-scale field; thus, it better discriminates the generated field from the target field. Similarly, for T_L and $2T_L$, \hat{D}^Y has more weight in the intermediate-scale range than in the other ranges, and discrimination is limited to the intermediate-scale field because the small-scale field is fully decorrelated; thus, it is no longer useful for discrimination. Although the manner in which \hat{D}^X conditionally influences learning is unclear, the scale-dependent correlation of the target field with the initial input field appears to be captured by \hat{D}^X . More importantly, it is conjectured that through \hat{D}^X , the large-scale motions of the input field play an important role in determining the small-scale motions of the generated field at finite lead times since the enstrophy cascades towards the small scales in 2-D turbulence. This scale-selective and scale-interaction feature of the discriminator appears to be the key element behind the successful prediction of the small-scale characteristics of turbulence at finite lead times. Here, the generator implicitly learns system characteristics embedded in data to deceive the discriminator with such features; thus, it is extremely challenging to provide an exact mechanism for how prediction performance is comprehensively enhanced, particularly in terms of physical and statistical attributes through adversarial training. However, we clearly showed that such a successful prediction of turbulence, even down to small scales, is possible through adversarial learning and not by the use of a CNN, which enforces suppression of the pointwise MSE only.

One last question that may arise in this regard is whether introducing physical constraints explicitly into the model, rather than using implicit competition with an additional network, could also be effective. To explore this, we incorporated the enstrophy spectrum as an explicit loss term into the objective function of the CNN model to address the performance issues associated with small-scale features. As shown in the relevant results presented in [Appendix D](#), adding the enstrophy loss alone did not lead to better performance although the enstrophy spectrum was better predicted. This confirms that the adoption of an implicit loss based on the latent variable in cGAN is effective in reflecting the statistical nature of turbulence, particularly the small-scale characteristics and the interactions between different scales. Once again, it is stressed that recovering

the statistical nature of the small-scale motions along with learning the scale interaction through the enstrophy cascade between the large-scale and small-scale motions can accurately predict turbulence in all scales even at finite lead times and that cGAN-based algorithm possesses such capability.

4.3. PredictionNet: single vs recursive prediction

In the previous section, we observed that the prediction performance for large lead times deteriorated naturally because of the poor correlation between the target field and the initial input field. An improved prediction for large lead times may be possible if recursive applications of the prediction model for short lead times are performed. Conversely, recursive applications may result in the accumulation of prediction errors. Therefore, there is an optimal model that can produce the best predictions. In this regard, since the performance of CNN models itself falls short of PredictionNet across all lead times, a more significant error accumulation is expected to arise. Thus, we would focus on analysing the recursive prediction results using PredictionNet. Nevertheless, we also presented the results of CNN models, not only to highlight the improvements through recursive prediction for large lead-time prediction but also to compare how much further enhancement is achieved when utilising PredictionNet. A similar concept was tested for weather prediction (Daoud, Kahl & Ghorai 2003).

For example, for the prediction of a large lead time $2T_L$, four base models trained to predict the lead times $0.125T_L$, $0.25T_L$, $0.5T_L$ and T_L were recursively applied 16, 8, 4 and 2 times, respectively, and compared against the single prediction as shown in figure 13. All recursive applications produced better predictions than the corresponding single prediction, and among them, four recursive applications of the cGAN model for $0.5T_L$ yielded the best results. Eight recursive applications of CNN model for $0.25T_L$, however, produces the best prediction. For all lead times, the best prediction was sought; the performances of all recursive predictions by cGAN and CNN are compared in table 3 in terms of performance indices, such as the CC and MSE and statistical quantities, including r.m.s., $\hat{\mu}_4$ and dissipation rate. Here, we observed that the performance difference depending on the input time point varies significantly depending on the base model; thus, the time-averaged values for CC and MSE from t_0 to t_{100} as input are presented to ensure fair comparison, unlike § 4.1. The CC and MSE values are plotted in figure 14. As expected, there exists an optimum base model producing the best performance for each lead time. Recursive prediction was more effective for large lead times (T_L and $2T_L$) than for short lead times ($0.25T_L$ and $0.5T_L$) for cGAN model. For $2T_L$, the CC improved from 0.4530 to 0.7394 and MSE decreased from 0.506 to 0.250 through the best recursive applications of cGAN model. The predicted statistics exhibited a consistent improvement. For T_L , the recursive applications show similar improvements. However, for $0.25T_L$ and $0.5T_L$, even though the recursive applications produced slightly better performance in terms of the CC and MSE, the statistics predicted by the single prediction are more accurate than those of the recursive prediction. However, recursive applications of CNN model do not show a monotonic behaviour in CC or MSE; for T_L , eight applications of CNN model trained for $0.125T_L$ yield the best prediction, whereas two applications of CNN model for T_L produces the best prediction for $2T_L$. Finally, the prediction of the enstrophy spectrum by recursive prediction also yielded an improvement over the single prediction, as shown in figure 15. In particular, it is noticeable that eight recursive applications of CNN model trained for $0.25T_L$ yielded relatively better spectrum than the single prediction. However, the performance of prediction was not good enough as shown in figure 13(c).

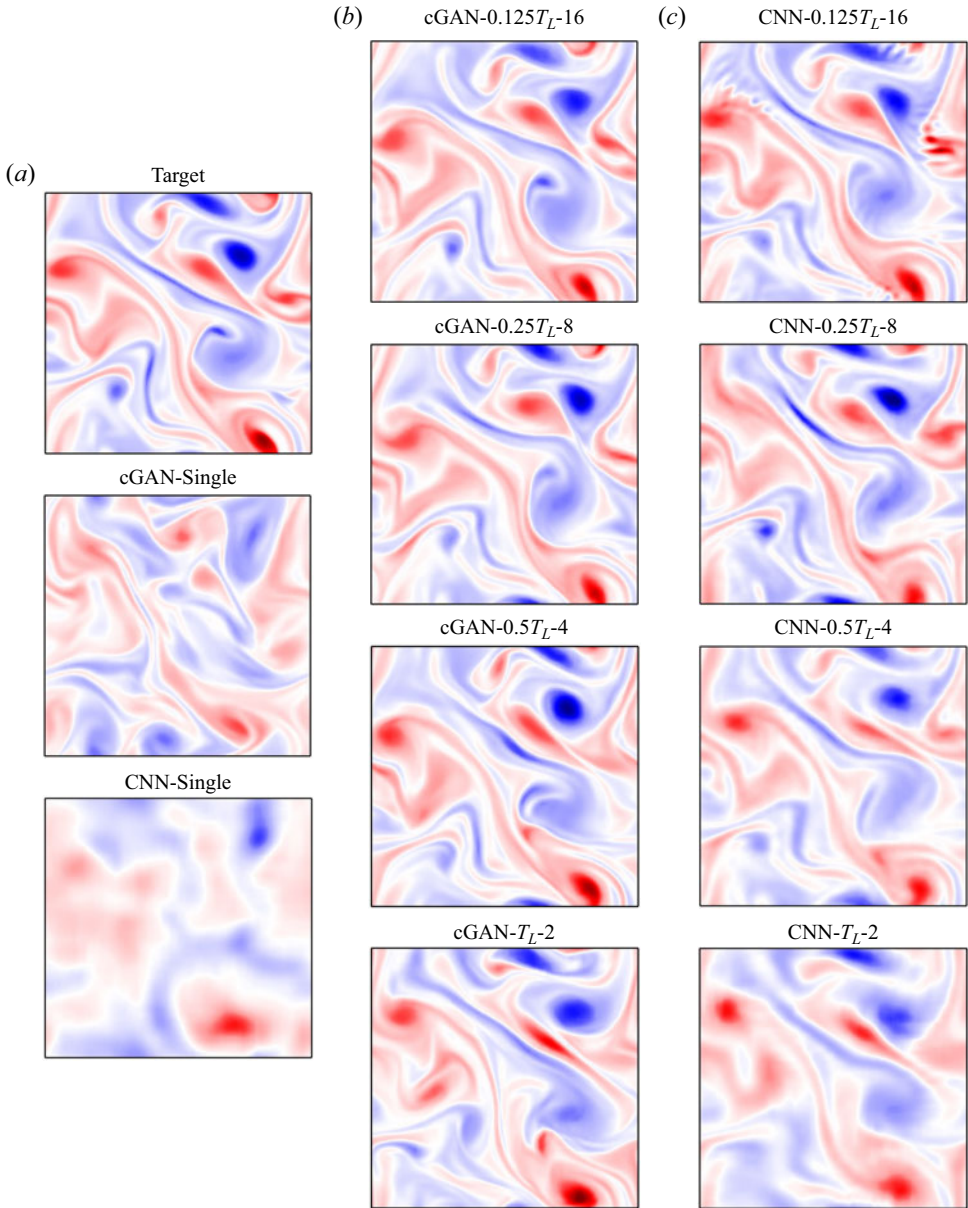


Figure 13. Visualised prediction results for $T = 2T_L$ of (a) target DNS field and single predictions using cGAN and CNN, (b) cGAN recursive predictions and (c) CNN recursive predictions.

It is noteworthy that for the prediction at T_L and $2T_L$, the best recursive prediction is made using two and four consecutive applications of the model trained for $0.5T_L$, respectively, whereas for the prediction at $0.5T_L$, any recursive prediction does not improve the performance significantly. This indicates that the prediction model for the lead time up to $0.5T_L$ is almost ‘complete’ in that the small-scale motions at up to $0.5T_L$ can be entirely recovered from the input data. A plausible interpretation of this observation is that the time required for the enstrophy to cascade to the smallest scale in the current 2-D turbulence is within $0.5T_L$. Therefore, the prediction at T_L or $2T_L$ is inevitably incomplete.

	Recursive model	CC		MSE		r.m.s.		$\hat{\mu}_4$		$\varepsilon' (\times 10^{-2})$	
		cGAN	CNN	cGAN	CNN	cGAN	CNN	cGAN	CNN	cGAN	CNN
$2T_L$	Target	—	—	—	—	0.788	—	3.65	—	1.28	—
	0.125 T_L -16	0.625	0.757	0.341	0.253	0.736	0.764	3.51	3.28	1.12	1.20
	0.25 T_L -8	0.672	0.558	0.300	0.419	0.746	0.722	3.66	3.59	1.15	1.07
	0.5T_L-4	0.739	0.705	0.250	0.259	0.776	0.625	3.75	3.52	1.24	0.80
	T_L -2	0.708	0.766	0.281	0.209	0.758	0.656	3.61	3.53	1.18	0.88
Single	0.453	0.523	0.506	0.386	0.698	0.580	3.57	4.06	1.00	0.70	
T_L	Target	—	—	—	—	0.888	—	3.27	—	1.62	—
	0.125 T_L -8	0.891	0.941	0.126	0.0708	0.854	0.828	3.20	3.12	1.50	1.41
	0.25 T_L -4	0.916	0.866	0.0971	0.157	0.860	0.819	3.30	3.34	1.52	1.38
	0.5T_L-2	0.924	0.919	0.0907	0.0936	0.875	0.751	3.32	3.30	1.57	1.16
	Single	0.855	0.887	0.172	0.130	0.860	0.795	3.26	3.41	1.52	1.30
$0.5T_L$	Target	—	—	—	—	0.944	—	3.08	—	1.83	—
	0.125 T_L -4	0.979	0.983	0.0277	0.0228	0.924	0.887	3.06	3.18	1.76	1.62
	0.25T_L-2	0.983	0.971	0.0227	0.0383	0.928	0.882	3.11	3.29	1.77	1.60
	Single	0.971	0.970	0.0390	0.0404	0.936	0.892	3.11	3.15	1.80	1.64
	Target	—	—	—	—	0.973	—	3.00	—	1.94	—
$0.25T_L$	0.125T_L-2	0.995	0.993	7.41 $\times 10^{-3}$	9.51 $\times 10^{-3}$	0.962	0.921	2.99	3.21	1.90	1.75
	Single	0.994	0.990	8.58 $\times 10^{-3}$	0.0144	0.964	0.960	3.02	3.02	1.91	1.90
$0.125T_L$	Target	—	—	—	—	0.987	—	2.96	—	2.00	—
	Single	0.998	0.997	2.74 $\times 10^{-3}$	4.75 $\times 10^{-3}$	0.981	0.981	2.96	2.96	1.98	1.98

Table 3. Statistical results of recursive predictions for various values of T including the single prediction at the last entries. Best prediction for each lead time by cGAN is bold-faced.

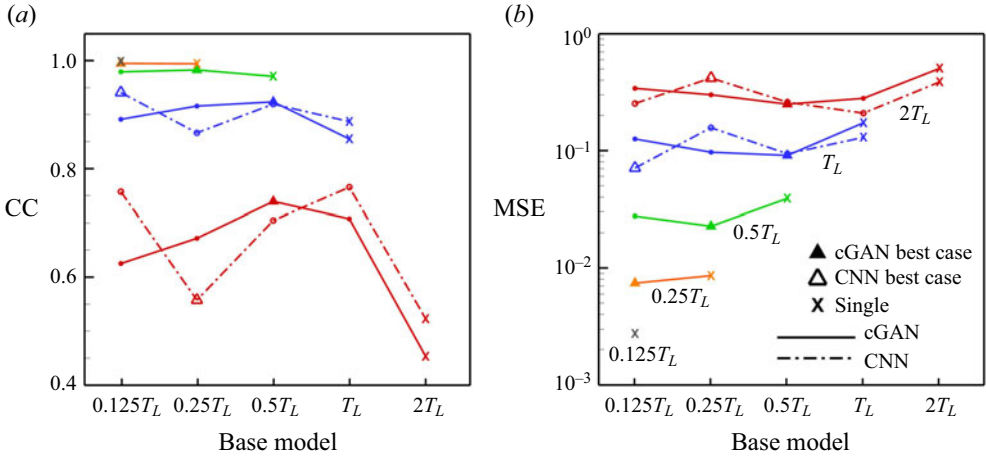


Figure 14. Plots of the CC and MSE of the recursive predictions in terms of the lead time of the base recursive model. Only T_L and $2T_L$ cases are included for CNN for visibility.

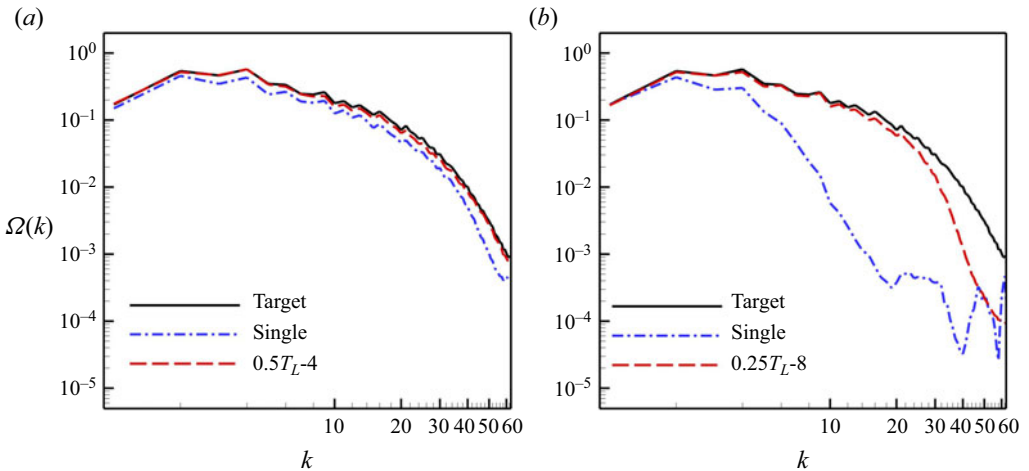


Figure 15. Enstrophy spectra of the best case of recursive prediction compared with the target and single prediction for $T = 2T_L$ using (a) cGAN and (b) CNN.

4.4. ControlNet: maximisation of the propagation of the control effect

In this section, we present an example of flow control that uses the high-accuracy prediction model developed in the previous section as a surrogate model. By taking advantage of the prediction capability of the model, we can determine the optimum control input that can yield the maximum modification of the flow in the specified direction with a fixed control input strength. In particular, we trained ControlNet to find a disturbance field that produces the maximum modification in the vorticity field over a finite time period with the constraint of a fixed r.m.s. of the disturbance field. Similar attempts to control 2-D decaying turbulence were made by Jiménez (2018) and Yeh *et al.* (2021). Jiménez (2018) modified a part of a vorticity field to identify dynamically significant sub-volumes in 2-D decaying turbulence. The influence on the future evolution of the flow was defined as significance using the L2 norm of the velocity field. Then, the

significance of the properly labelled regions was compared by turning the vorticity of specific regions on and off. They showed that vortices or vortex filaments are dynamically significant structures, and interstitial strain-dominated regions are less significant. In contrast, Yeh *et al.* (2021) developed a method called network-broadcast analysis based on network theory by applying Katz centrality (Katz 1953) to identify key dynamical paths along which perturbations amplify over time in 2-D decaying turbulence. Two networks (composed of nodes and edges, not neural networks), the Biot–Savart (BS) network and Navier–Stokes (NS) network, with different adjacency matrices were investigated. In the former case, vortex structures similar to those in Jiménez (2018), and in the latter case, opposite-sign vortex pairs (vortex dipoles) were the most effective structures for perturbation propagation. However, both studies confined the control disturbance field either by localising the modification for control by dividing the domain into cell units (Jiménez 2018), or by considering the perturbation using the leading singular vector of an adjacency matrix calculated from a pulse in a predefined shape (Yeh *et al.* 2021). However, the disturbance field considered in ControlNet is free of shape, and the only constraint is the strength of the disturbance field in terms of the fixed r.m.s. of the input field.

As a surrogate model for the time evolution of the disturbance-added initial field, PredictionNet trained for a lead time of $0.5T_L$ was selected for the control test. ControlNet is trained to generate an optimum disturbance field ΔX that maximises the difference between the disturbance-added prediction $\tilde{Y} (= \text{Pred}(X + \Delta X))$ and the original (undisturbed) prediction Y^* . If the strength of ΔX is too large, causing $X + \Delta X$ to deviate from the range of dynamics of the pre-trained PredictionNet, the surrogate model will not function properly, resulting in \tilde{Y} being different from the ground-truth solution $N(X + \Delta X)$. Here, $N()$ is the result of the Navier–Stokes simulation, with the input as the initial condition. Here \tilde{Y} , using disturbances with various strengths of the r.m.s. of the input field ($\Delta X_{rms} = 0.1, 0.5, 1, 5, 10, 20\%$ of X_{rms}), were compared with the corresponding $N(X + \Delta X)$. We confirmed that PredictionNet functions properly within the tested range (Appendix E). Therefore, the effect of the disturbance strength was not considered; thus, we only present the 10% case. Coefficient θ related to the smoothing effect was fixed at 0.5 for $\Delta X_{rms} = 0.1X_{rms}$ through parameter calibration. Figure 16 shows the data loss trend of ControlNet for $\Delta X_{rms} = 0.1X_{rms}$, which is maximised as the training progresses, confirming successful training.

Figure 17 presents a visualisation of the effect of the control input on one of the test data. Figure 17(a) shows the input vorticity field and corresponding prediction at $T = 0.5T_L$. Figure 17(b) shows the disturbance field generated by ControlNet ΔX_C , disturbance-added prediction, and difference between the undisturbed and disturbance-added predictions, demonstrating that the control effect is effectively propagated and amplified at $T = 0.5T_L$. The MSE between \tilde{Y}_C and Y^* is 0.757, indicating a substantial change. A prominent feature of ΔX_C is its large-scale structure. To verify whether the propagation of the control effect of \tilde{Y}_C was maximised under the given conditions, we considered three additional disturbance fields. Inspired by the results of Jiménez (2018) and the claim of Yeh *et al.* (2021) regarding their BS network, in which vortices have the greatest effect on the future evolution of the base flow, we considered a 10% scaled input disturbance $\Delta X_{SI} (= 0.1X)$. Another disturbance field due to a single Taylor vortex ΔX_{TV} extracted from the NS network of Yeh *et al.* (2021), which is best for amplifying the control effect when applied to a vortex dipole rather than the main vortex structures, is considered in the following form:

$$\Delta X_{TV}(x, y) = \epsilon \delta(x_c, y_c) = \epsilon(2/r_\delta - r^2/r_\delta^3) \exp[-r^2/(2r_\delta^2)], \quad (4.2)$$

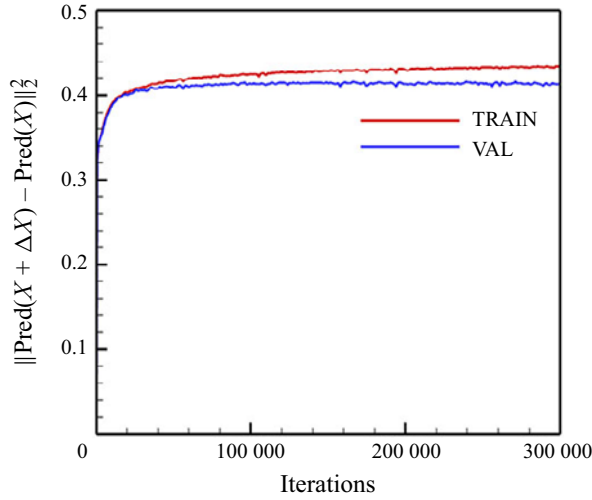


Figure 16. Convergence of the data loss of ControlNet for $0.5T_L$. Here $\Delta X_{rms} = 0.1X_{rms}$.

where $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$ with vortex centre (x_c, y_c) . Here ϵ represents the amplitude of the Taylor vortex, and it is adjusted to maintain the r.m.s. of the disturbance at $0.1X_{rms}$. We set r_δ to $r_\delta = \pi/k_{max}$, where $k_{max} = 4$ denotes the wavenumber with the maximum value in the enstrophy spectra. Finally, the disturbance field in the form of a Taylor–Green vortex ΔX_{TG} approximating the disturbance field of ControlNet is considered as follows:

$$\Delta X_{TG}(x, y) = \epsilon \sin\left(\frac{1}{\sqrt{2}}(x - x_c - y + y_c)\right) \sin\left(\frac{1}{\sqrt{2}}(x - x_c + y - y_c)\right), \quad (4.3)$$

indicating the largest vortex in the $[0, 2\pi)^2$ -domain. For ΔX_{TV} and ΔX_{TG} , the optimum location of (x_c, y_c) that yielded the greatest propagation was determined through tests using PredictionNet.

The disturbance-added predictions \tilde{Y}_{SI} and \tilde{Y}_{TV} and their differences from the undisturbed prediction Y^* using ΔX_{SI} and ΔX_{TV} located at the optimum position as disturbances, respectively, are shown in figure 17(c,d). The difference fields show that these disturbances do not significantly change the field. MSE values for \tilde{Y}_{SI} and \tilde{Y}_{TV} against the undisturbed prediction are 0.039 and 0.402, respectively, which are much smaller than the corresponding value of 0.757 for \tilde{Y}_C , as shown in figure 18(a). From this comparison, it can be conjectured that because the goal of control is to maximise the pointwise MSE of the vorticity against the undisturbed prediction, the large-scale disturbance of the vorticity is more effective in displacing and deforming the vorticity field. The enstrophy spectra of the disturbance fields shown in figure 18(b) confirm that ΔX_C has a peak value at $k = 1$, representing the largest permissible scale in the given domain, whereas ΔX_{TV} and ΔX_{SI} have peak values at $k = 2$ and $k = 4$, respectively, under the constraint that the r.m.s. value of all disturbances is $0.1X_{rms}$. This observation leads to the consideration of the largest Taylor–Green vortex-type disturbance ΔX_{TG} in the domain given by (4.3). The distribution of the optimum location of ΔX_{TG} shown in figure 17(e), yielding the greatest propagation coincides with that of ΔX_C shown in figure 17(b) (except for the sign owing to the symmetry in the propagation effect, as shown in the third panel of figure 17b,e). The MSE of \tilde{Y}_{TG} against Y^* was 0.726, which was slightly smaller than 0.757 of \tilde{Y}_C

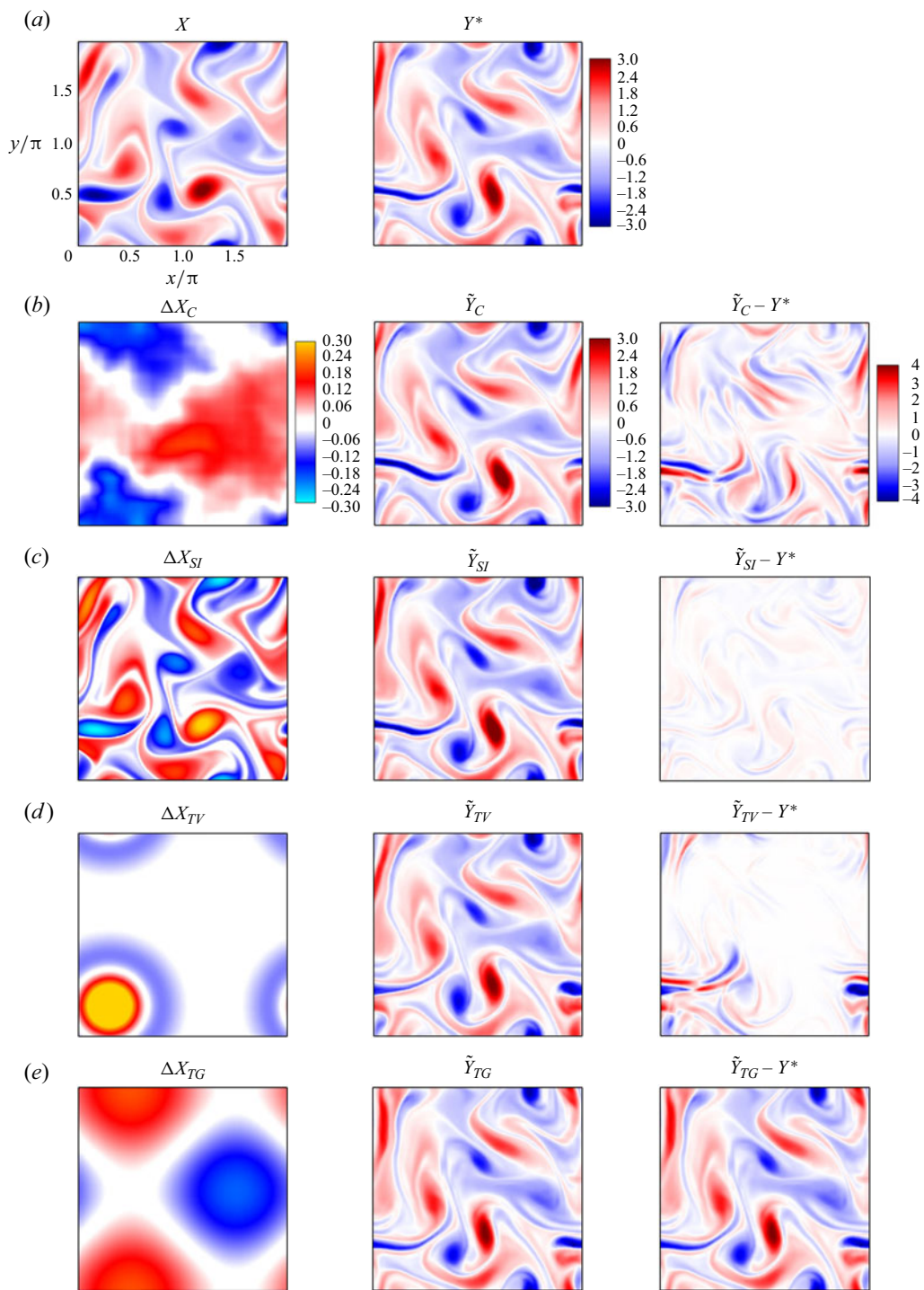


Figure 17. Visualised example of disturbance fields: (a) input and the undisturbed prediction, (b) the optimum disturbance (Control(X)), disturbance-added prediction (Pred($X + \Delta X_C$)) and the difference. Comparison cases of (c) ΔX_{SI} , (d) ΔX_{TV} and (e) ΔX_{TG} .

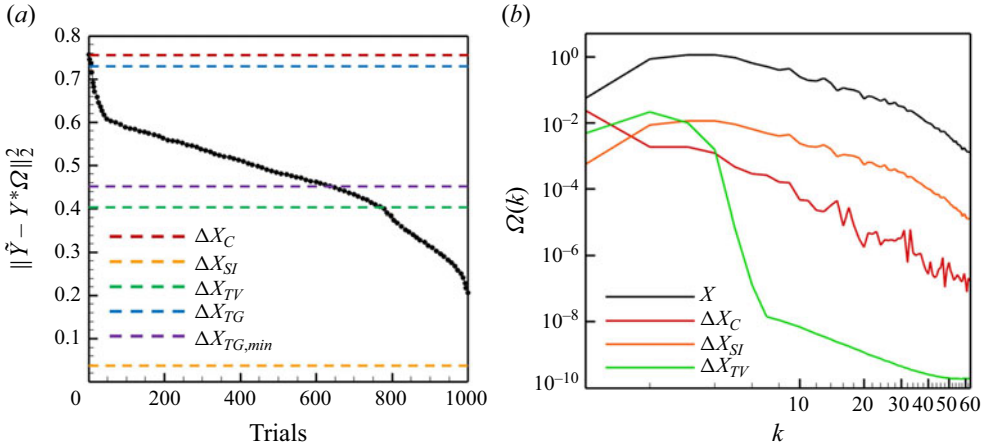


Figure 18. (a) Comparison of MSEs between various \tilde{Y} and Y^* using different disturbance fields. Black dots denote the sorted MSE of phase-shifted ΔX_C . The first node of black dots (red dashed line) shows the result of the optimum ΔX_C . (b) Enstrophy spectra of the input X , ΔX_C , ΔX_{SI} and ΔX_{TV} .

(figure 18a). All these comparisons suggest that the largest-scale disturbance is the most effective in modifying the vorticity field through displacement or distortion of the given vorticity field. To verify whether the location of the largest-scale disturbance ΔX_C was indeed globally optimal, tests were conducted with a phase-shifted ΔX_C . We consider two types of phase shifting for the wavenumber components $\hat{\Delta X}_C \exp(ik_x l_x + ik_y l_y)$: one with randomly selected l_x and l_y uniformly applied to all wavenumbers and the other with randomly selected phases differently applied to each wavenumber. The test results confirm that the maximum MSE was obtained for ΔX_C (without a phase shift), with the minimum MSE found at 0.2 among the 1000 trial disturbances considered with randomly selected phases, as shown in figure 18(a). Here $\Delta X_{TG,min}$ corresponds to one of the ΔX_{TG} tests yielding the minimum modification with an MSE of approximately 0.45. The wide range of MSE for the largest-scale disturbance indicates that the location of the largest-scale disturbance is important for modifying the flow field.

To confirm the optimality of ΔX_C in real flow, full Navier–Stokes simulations were performed further than $0.5T_L$ using the disturbance-added initial conditions for ΔX_C , ΔX_{SI} , ΔX_{TV} and ΔX_{TG} . The visualisation results of the propagation over the time horizon are shown in figure 19, and the behaviour of the normalised RMSE over time is shown in figure 19(e). Therefore, the propagation by large-scale disturbances such as ΔX_C and ΔX_{TG} is much more effective than that by ΔX_{SI} and ΔX_{TV} even up to longer than T_L . Moreover, the surrogate model functions properly for $\Delta X_{rms} = 0.1X_{rms}$ based on the comparison between the results at $0.5T_L$ in figure 19 and those in the third column (difference fields) in figure 17 for each disturbance. The RMSE of ΔX_C and ΔX_{TG} behave almost indistinguishably as shown in figure 19(e) for all test periods.

As shown in figure 18(a), the location of a large-scale disturbance causes a difference in the modification of the flow field. To investigate this in more detail, the vorticity contours of the flow field and the velocity vectors of the disturbances yielding the greatest change, ΔX_C and ΔX_{TG} , and the disturbance producing the least change, $\Delta X_{TG,min}$ are plotted together in figure 20. The velocity vector distributions of ΔX_C (figure 20a) and ΔX_{TG} (figure 20b) are similar, whereas those of $\Delta X_{TG,min}$ (figure 20c) differ significantly. Careful observation shows that the maximum modification occurs when the velocity of the disturbances is applied in the direction normal to the elongation direction of the

Prediction and control of 2-D decaying turbulence using GANs

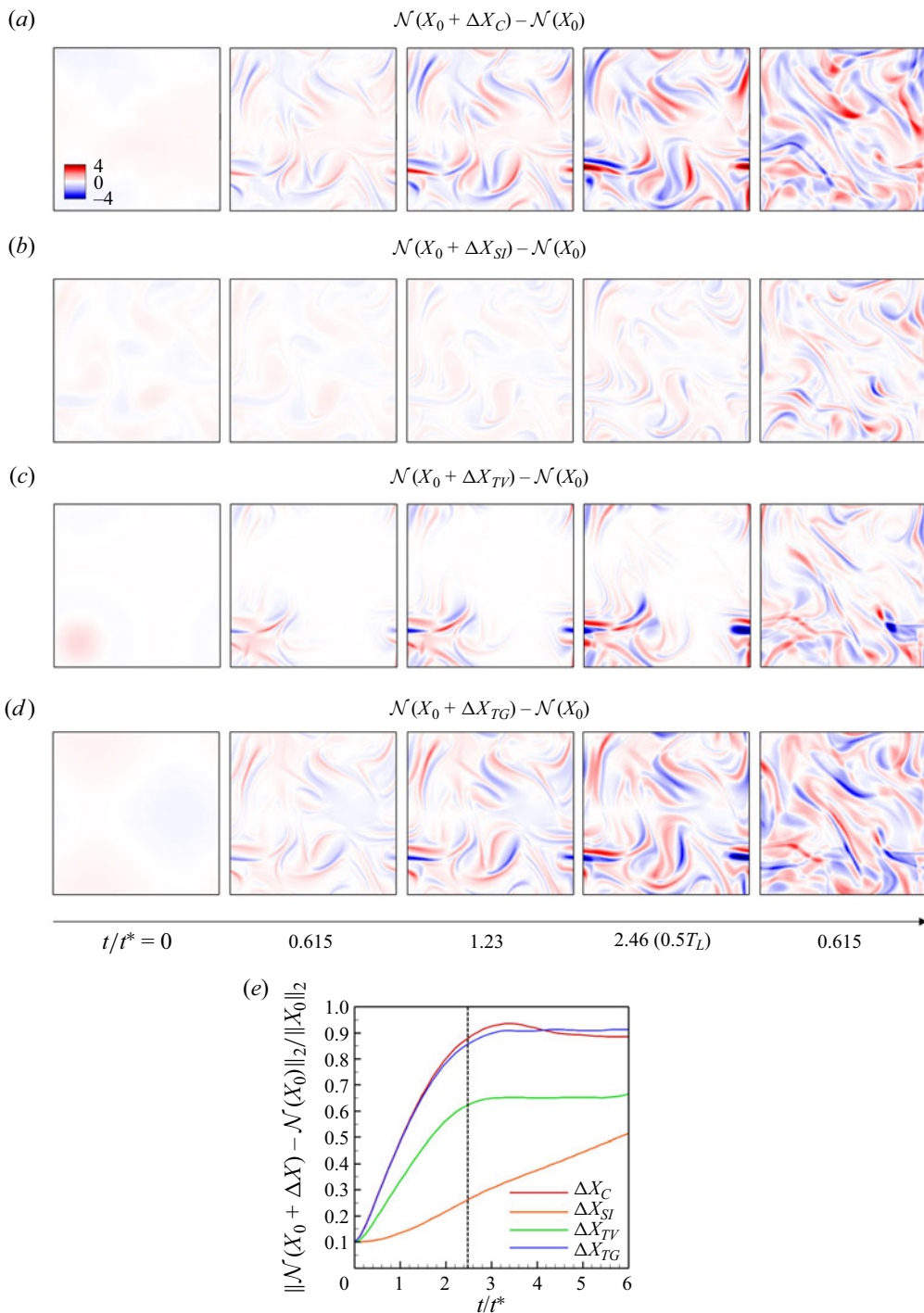


Figure 19. Simulated propagations of the control effect along with the time horizon presented by differences between the disturbance-added simulations and the original simulation: (a) ΔX_C , (b) ΔX_{SI} , (c) ΔX_{TV} and (d) ΔX_{TG} . The RMSE result of the propagation of each disturbance is shown in (e) after being normalised by the input r.m.s.

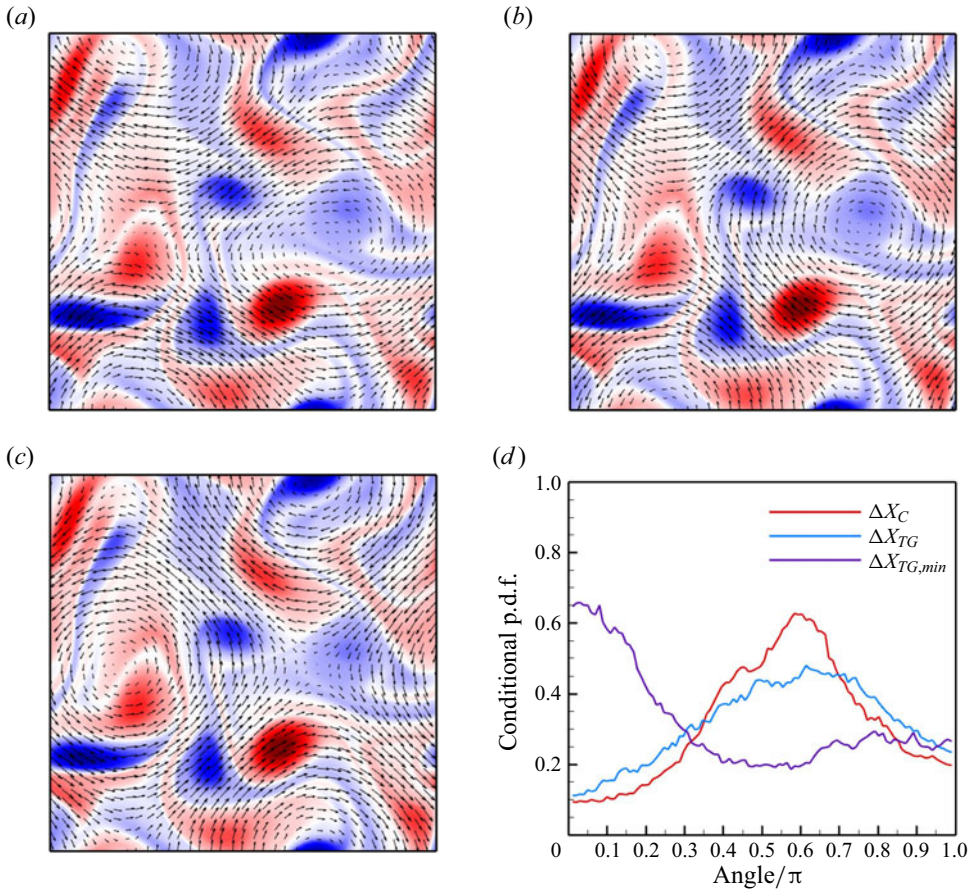


Figure 20. Distribution of the velocity vector field with the input vorticity contours for (a) ΔX_C , (b) ΔX_{TG} and (c) $\Delta X_{TG,min}$. (d) Conditional p.d.f. of the angle between the velocity vectors of the input and disturbance.

strong vortex region, whereas the flow field is minimally changed when the velocity of the disturbances is in the elongation direction of the strong vortex patch. The conditional p.d.f. of the angle between the velocity vectors of the input field X and disturbance fields is obtained under the condition that the velocities of the input and disturbance are greater than their own r.m.s. values. Figure 20(d) shows that the peak is found at approximately 0.6π for ΔX_C and ΔX_{TG} , and zero for $\Delta X_{TG,min}$, confirming this trend, given that the velocity vectors circumvent the vortex patches.

5. Conclusion

In this study, the dynamics prediction of 2-D DHIT was performed using a cGAN-based deep learning model. The developed prediction network, called PredictionNet, produced highly accurate predictions up to a lead time of half the Eulerian integral time scale. A quantitative comparison of the prediction performance with that of the baseline CNN model, which is an identical network to PredictionNet but trained in a different framework, proved the superiority of the GAN-based model. In particular, the small-scale turbulence characteristics, which were not properly predicted by the CNN model, were captured well by the cGAN-based model. A detailed investigation of the behaviour of the latent variable,

which is the output of the discriminator network, suggests that the discriminator network evolves through training to possess a scale-selection capability; thus, it can effectively distinguish the generated turbulence from the true turbulence. The minimisation of the pointwise MSE loss tended to capture large-scale motions only, whereas competitive optimisation of the discriminator network using the latent variable led to the recovery of small-scale motions. More specifically, PredictionNet generated the small-scale fields that not only mimic the statistical characteristics of the target turbulence but are also consistent with the large-scale motion of turbulence. On the other hand, we observed that an attempt to maximise an explicitly added physical loss term in the CNN framework reflecting the statistical feature of the small-scale motions failed to perform well. All of these suggest that our model based on cGAN is capable of capturing both the statistical nature of the small-scale motions and the interaction between the large-scale and small-scale motions.

In addition, recursive predictions were tested using high-accuracy base models for short lead times to improve the predictive accuracy for long lead times. As shown in [figure 13](#), four recursive applications of the prediction model trained for $0.5T_L$ yielded significantly better predictions than the single prediction for $2T_L$. However, more recursive applications of the prediction model for shorter lead times did not always guarantee improvement, indicating that an optimum recursive model exists for long lead times, which is the model trained for $0.5T_L$. Therefore, the prediction model for up to $0.5T_L$ is complete in that the small-scale motions are entirely recovered from the input, suggesting that the time scale of the enstrophy cascade from the largest scale to the smallest scale in current 2-D turbulence is within $0.5T_L$.

Flow control was conducted as an example of application of the developed high-accuracy prediction model. Using the developed prediction model for $0.5T_L$ as a surrogate model, a control network was trained to provide an optimum disturbance field that maximised the modification at a given lead time. When the pointwise mean-squared difference between the disturbance-added prediction and undisturbed prediction at $0.5T_L$ was maximised, the optimum disturbance turned out to be of the largest scale fitting the domain. This maximum propagation of the control effect appeared to be achieved through the translation of elongated vortices in a direction orthogonal to the elongation direction. Although the optimum disturbances were found in a model-free manner (the only constraint is the condition for r.m.s.), our models converged well to the optimum solution despite the high probability of falling into local optima because of the high degree of freedom.

We have demonstrated that it is possible to predict 2-D turbulent fields well in all scales at finite lead times up to a half the Eulerian integral time scale using cGAN-based deep learning. An accurate prediction of the fast-evolving small-scale motions at such a long lead time is indeed possible owing to the competitive learning of cGAN based on the latent variable. In addition to learning of the statistical nature of the small-scale motions, our model reflects the large-scale motions in the generation of the small-scale motions through conditioning process such that the generated small-scale motions are consistent with the large-scale motions. This suggests that as long as the large-scale motions remain temporally correlated, the corresponding small-scale motions can be well predicted. In current 2-D turbulence, $0.5T_L$ appears to be the limitation of the lead time for successful prediction. This is in a sense consistent with successful super-resolution reconstructions in which the small-scale motions can be well reconstructed using only the large-scale motion data if using GAN-based learning (Kim *et al.* 2021). Such a strong tie between the large-scale and the small-scale motions is naturally attributed to the enstrophy cascade in two dimensions and energy cascade in three dimensions through which the behaviour of the small-scale motions is predominantly determined by the large-scale motions.

This also supports the feasibility of LES modelling in which the sub-grid scale model is constructed using the resolved scale data only. This finding of the scale interaction in deep learning application could be useful for improving the prediction performance of weather forecasting using ML algorithms attempted recently (Bi *et al.* 2023; Lam *et al.* 2023; Park & Lee 2023).

Our investigation was restricted to 2-D DHIT. Therefore, a natural extension would be toward more general turbulent flows such as inhomogeneous 2-D flows or even 3-D turbulence. Application to 3-D homogeneous isotropic turbulence would be straightforward, even though there is a cost issue because training would require much more time than in 2-D flows. However, it is worthwhile investigating whether the scale-selection and scale-interaction capability of the discriminator network works in 3-D isotropic turbulence. We have shown that a generative model such as GAN is more effective in learning turbulence characteristics than a CNN. However, GAN is not the only generative model. Recently, a diffusion model (Sohl-Dickstein *et al.* 2015; Ho, Jain & Abbeel 2020; Shu, Li & Farimani 2023) was proposed as a generative model in which training is much more stable than GAN-based model. A comparative study of GAN and the diffusion models in the prediction of turbulence would be an interesting topic for future research.

Our sample code and datasets, which support the findings of this study, are openly available on a public Github repository: <https://github.com/jiyeon914/TurbGAN>.

Funding. This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP 2022R1A2C2005538).

Declaration of interests. The authors report no conflict of interest.

Author ORCIDs.

 Jiyeon Kim <https://orcid.org/0000-0001-6668-1951>;

 Junhyuk Kim <https://orcid.org/0000-0001-9662-9265>;

 Changhoon Lee <https://orcid.org/0000-0002-9813-8117>.

Appendix A. Pseudo-spectral approximation of the vorticity-streamfunction formulation

In two dimensions, the governing equations for the vorticity-streamfunction formulation are

$$\frac{\partial \omega}{\partial t} + \frac{\partial(u\omega)}{\partial x} + \frac{\partial(v\omega)}{\partial y} = \nu \left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right), \tag{A1}$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega, \tag{A2}$$

with

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x}. \tag{A3a,b}$$

In a biperiodic domain, the discrete Fourier representation of vorticity field is

$$\omega(x, y) = \sum_{\kappa_x} \sum_{\kappa_y} \hat{\omega}(\kappa_x, \kappa_y, t) \exp(i\kappa_x x) \exp(i\kappa_y y), \tag{A4}$$

Prediction and control of 2-D decaying turbulence using GANs

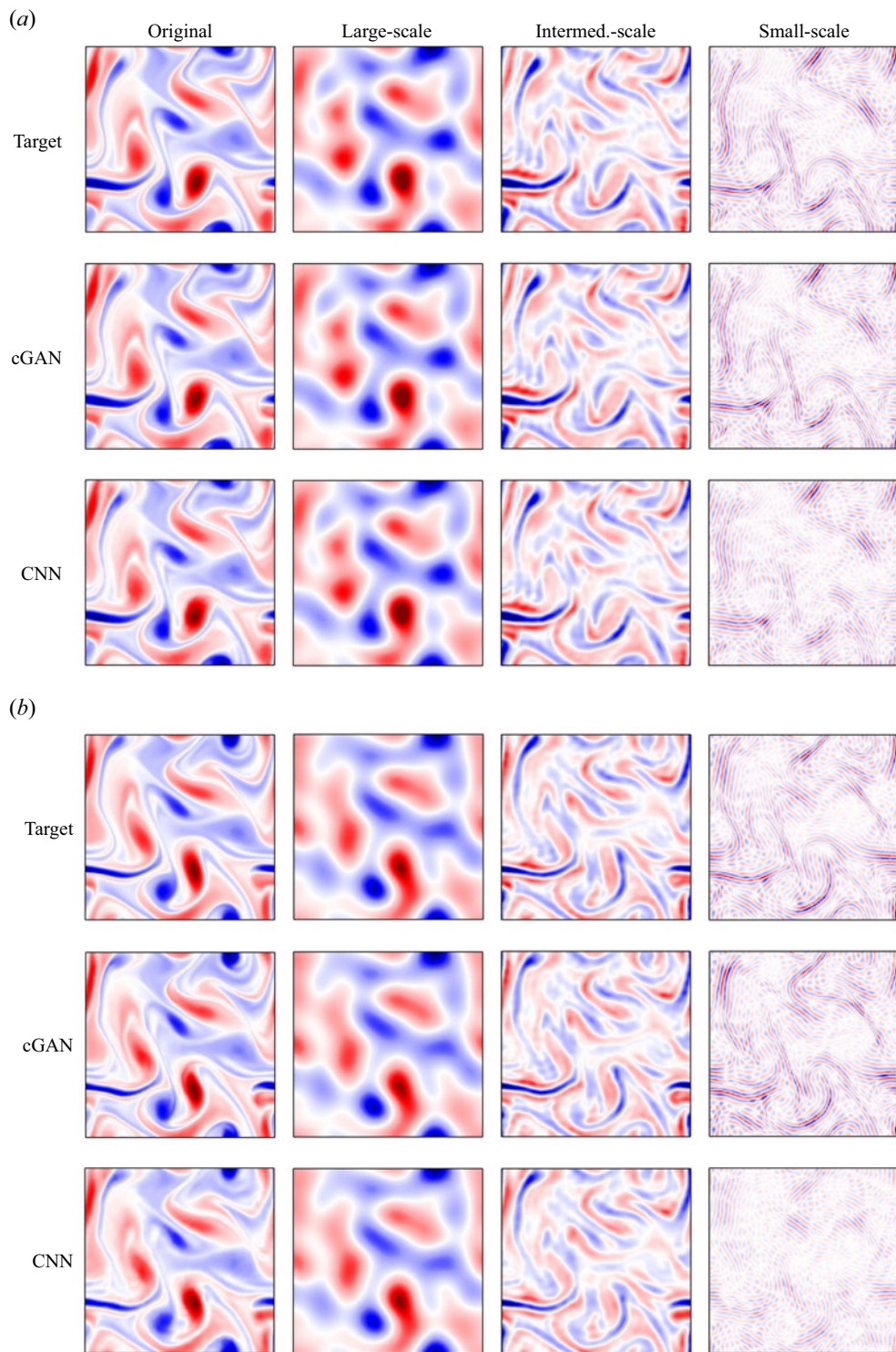


Figure 21. For the caption, see the next page.

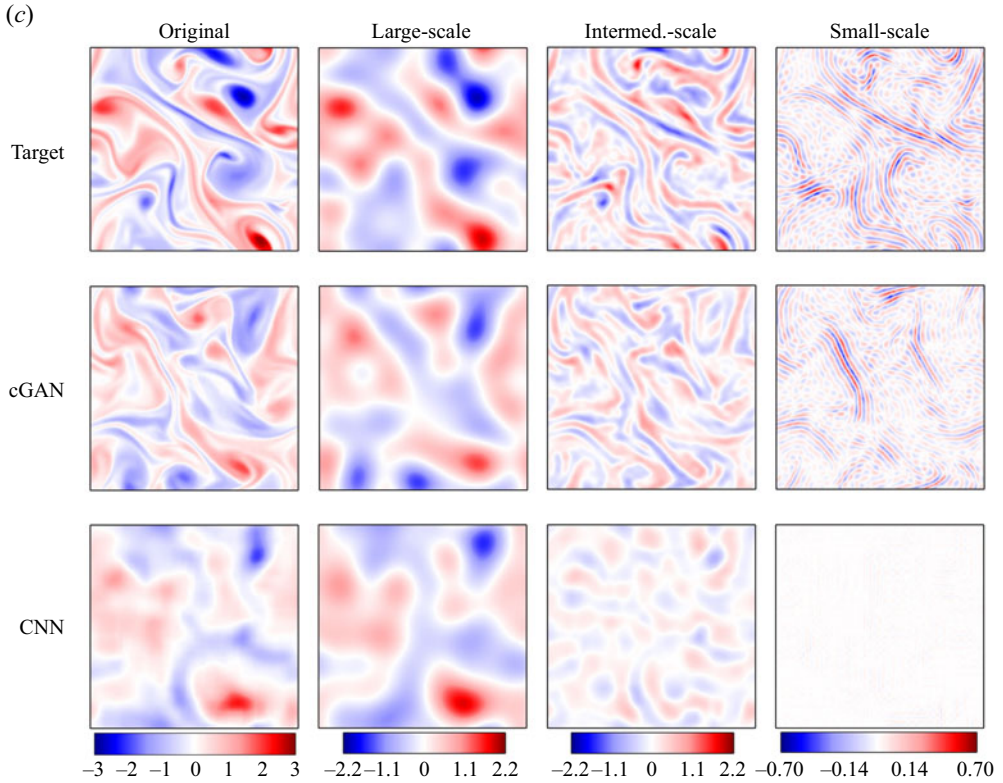


Figure 21. Scale decompositions of other values of T . (a) $0.25T_L$, (b) $0.5T_L$ and (c) $2T_L$.

where $\hat{\omega}$ is the Fourier coefficient of vorticity, and κ_x and κ_y are wavenumbers in x - and y -directions, respectively. Pseudo-spectral approximation to (A1) yields

$$\frac{\partial \hat{\omega}}{\partial t} + i\kappa_x \hat{u} \hat{\omega} + i\kappa_y \hat{v} \hat{\omega} = -\nu \kappa^2 \hat{\omega} \quad (\text{A5})$$

with

$$-\kappa^2 \hat{\psi} = -\hat{\omega}, \quad \hat{u} = i\kappa_y \hat{\psi}, \quad \hat{v} = -i\kappa_x \hat{\psi}, \quad (\text{A6a-c})$$

where $\hat{\psi}$, \hat{u} and \hat{v} correspond to the Fourier coefficients of ψ , u and v , respectively, and $\kappa^2 = \kappa_x^2 + \kappa_y^2$. Here $\hat{u} \hat{\omega}$ and $\hat{v} \hat{\omega}$ are the Fourier coefficients of $u\omega$ and $v\omega$ which are pointwisely calculated in the physical space. From (A6a-c),

$$\hat{u} = \frac{i\kappa_y}{\kappa^2} \hat{\omega}, \quad \hat{v} = -\frac{i\kappa_x}{\kappa^2} \hat{\omega}, \quad (\text{A7a,b})$$

yielding with (A5)

$$\frac{\partial \hat{u}}{\partial t} = \frac{i\kappa_y}{\kappa^2} \frac{\partial \hat{\omega}}{\partial t} \quad (\text{A8})$$

$$= \frac{i\kappa_y}{\kappa^2} \left(-i\kappa_x \hat{u} \hat{\omega} - i\kappa_y \hat{v} \hat{\omega} - \nu \kappa^2 \hat{\omega} \right) \quad (\text{A9})$$

$$= \hat{v} \hat{\omega} - i\kappa_x \left(-\frac{i\kappa_x}{\kappa^2} \hat{v} \hat{\omega} + \frac{i\kappa_y}{\kappa^2} \hat{u} \hat{\omega} \right) - \nu \kappa^2 \hat{u}. \quad (\text{A10})$$

Lead time	0.25T _L		0.5T _L		T _L		2T _L	
	cGAN	CNN	cGAN	CNN	cGAN	CNN	cGAN	CNN
Original	0.9940	0.9898	0.9708	0.9699	0.8547	0.8870	0.4530	0.5227
Large scale	0.9992	0.9983	0.9964	0.9952	0.9707	0.9750	0.6505	0.6542
Intermediate scale	0.9920	0.9869	0.9470	0.9449	0.6662	0.7166	0.0415	0.0292
Small scale	0.8956	0.8131	0.6348	0.5727	0.1148	0.0909	-0.0008	0.0004

Table 4. Correlation coefficient between the target and prediction of the scale-decomposed fields depending on the lead time.

Similarly,

$$\frac{\partial \hat{v}}{\partial t} = -\frac{i\kappa_x}{\kappa^2} \frac{\partial \hat{\omega}}{\partial t}, \tag{A11}$$

$$= -\widehat{u\omega} - i\kappa_y \left(-\frac{i\kappa_x}{\kappa^2} \widehat{v\omega} + \frac{i\kappa_y}{\kappa^2} \widehat{u\omega} \right) - \nu \kappa^2 \hat{v}. \tag{A12}$$

Noting that the quantity in brackets in ((A10) and (A12)) is identical, taking the inverse Fourier transform of both equations leads to

$$\frac{\partial u}{\partial t} - v\omega = -\frac{\partial P}{\partial x} + \nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right), \tag{A13}$$

$$\frac{\partial v}{\partial t} + u\omega = -\frac{\partial P}{\partial y} + \nu \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right), \tag{A14}$$

with

$$\hat{P} = -\frac{i\kappa_x}{\kappa^2} \widehat{v\omega} + \frac{i\kappa_y}{\kappa^2} \widehat{u\omega}, \tag{A15}$$

guaranteeing that the velocity field satisfies the divergence-free condition. Equations ((A13) and (A14)) are the rotational form of the 2-D Navier–Stokes equation. This proves that the pseudo-spectral approximation to the vorticity-streamfunction formulation is equivalent to pseudo-spectral approximation to the rotational form of the 2-D Navier–Stokes equation as first shown by Orszag (1971) and Fox & Orszag (1973).

Appendix B. Scale decomposition results of lead times other than T_L

This appendix presents the scale-decomposition results of both models and visualisations for other lead times. The CC of the decomposed fields between the target and predictions by the cGAN and CNN for all lead times is listed in table 4. The visualisations for $T = 0.25T_L$, $0.5T_L$ and $2T_L$ are shown in figures 21(a), 21(b) and 21(c), respectively. For $T = 0.25T_L$, as shown in figure 21(a), there appears to be no significant difference in the performances of the cGAN and CNN for all scales. This is owing to the high correlation for the short lead time (CC = 0.8131 in table 4), which enables the CNN model to generate accurate small-scale vorticity structures. However, the energy contained in the small-scale field is expected to be significantly underpredicted by the CNN from the enstrophy spectra in figure 7(a). For $T = 0.5T_L$, the underprediction of the small-scale field by the CNN worsens, compared with the target and cGAN predictions. However, cGAN shows high accuracy both in figure 21(a,b) in the prediction of the small-scale structures unlike $T = T_L$

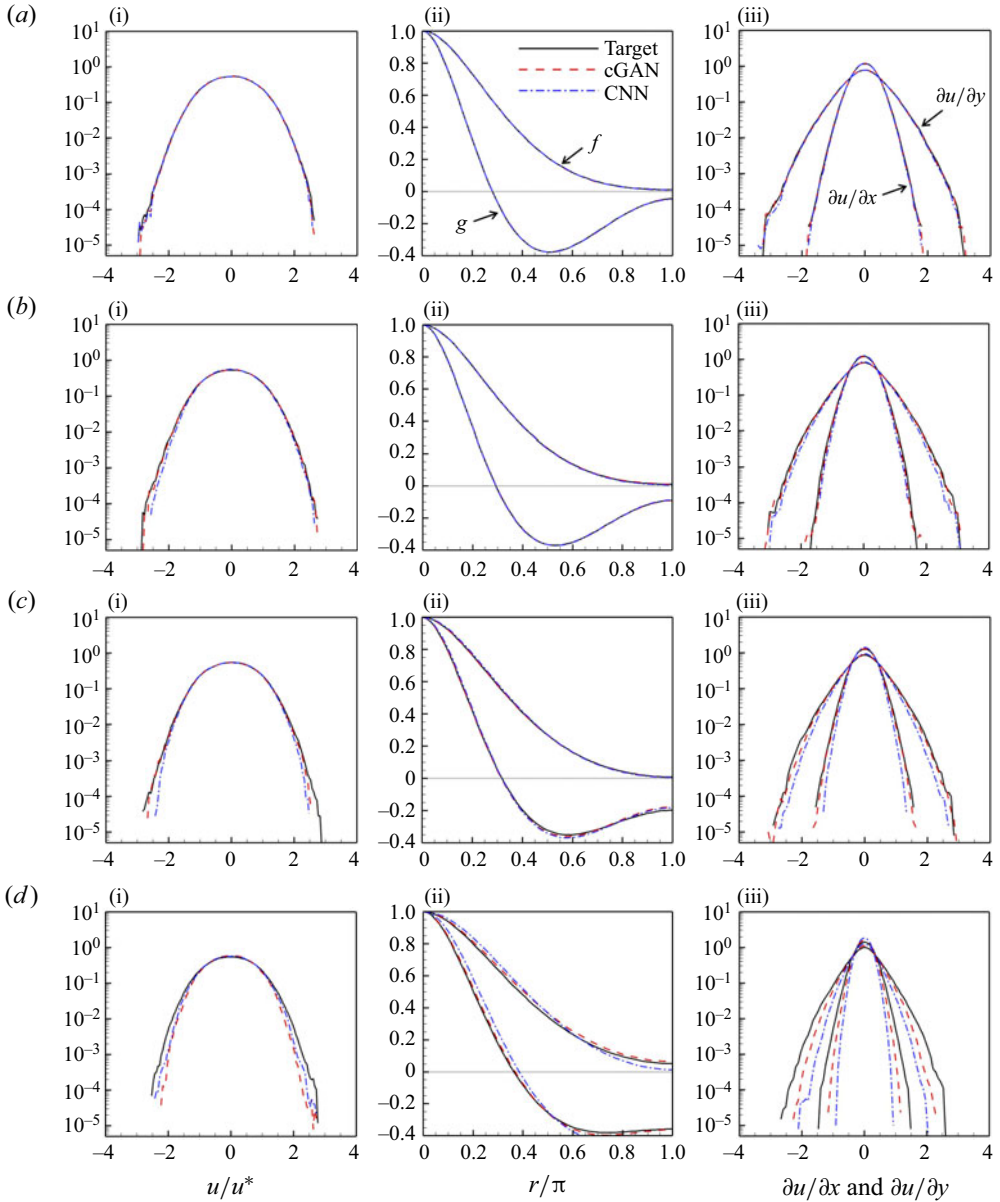


Figure 22. Comparison between the velocity statistics of the target and prediction results depending on T : (a) $T = 0.25T_L$, (b) $T = 0.5T_L$, (c) $T = T_L$ and (d) $T = 2T_L$ for the same input time point of t_0 . Panels (a i, b i, c i, d i), (a ii, b ii, c ii, d ii), and (a iii, b iii, c iii, d iii) display velocity p.d.f.s, longitudinal and transverse correlation functions and velocity gradient p.d.f.s, respectively.

in figure 9 of § 4.1. Finally, for $T = 2T_L$ in figure 21(c), the CNN fails to generate a proper prediction of the large-scale field, as well as the intermediate- and small-scale fields. Although the cGAN predictions were much better than those by CNN predictions, the overall predictions were not sufficiently good. However, in a statistical sense the cGAN appeared to generate reasonable predictions.

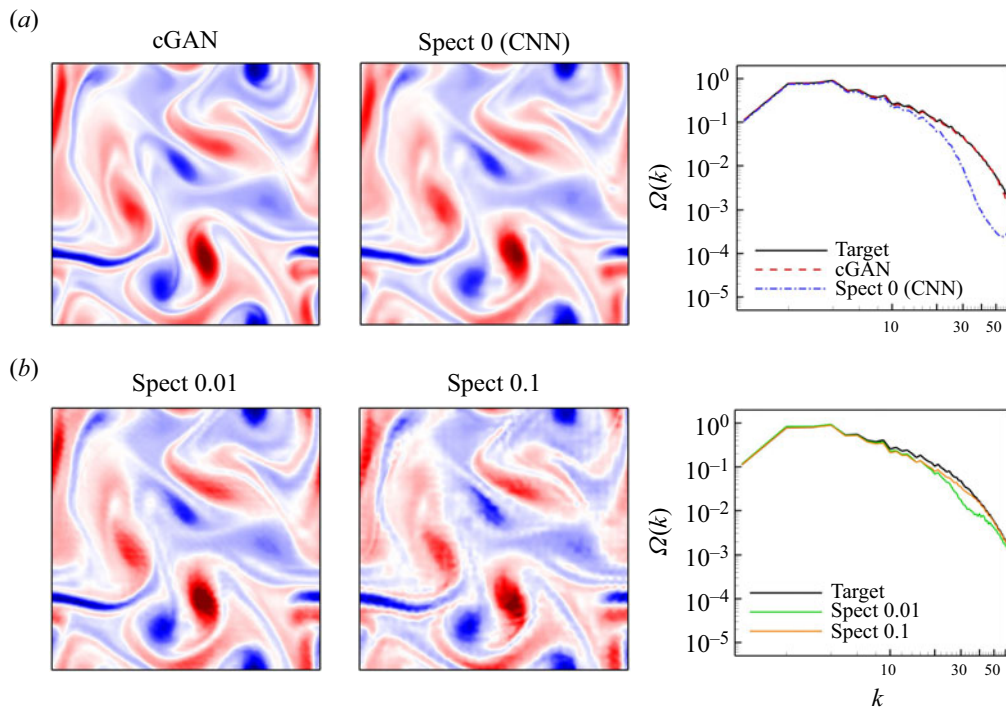


Figure 23. Visualised examples of prediction result and enstrophy spectra for $0.5T_L$ prediction using (a) cGAN and Spect 0 (baseline CNN) and (b) spectrum loss added CNNs.

Appendix C. PredictionNet: predictive accuracy on velocity statistics

The performance of PredictionNet was investigated in the vorticity field, because 2-D turbulence can be fully simulated using vorticity alone. In this appendix, the investigation of the performance of the developed network in predicting the velocity vectors is briefly discussed. In figure 22, the results of the velocity p.d.f.s, longitudinal and transverse correlation functions (f and g), and velocity gradient p.d.f.s are compared between the cGAN and CNN for various lead times. The overall superiority of cGAN over the baseline CNN is observed again. In particular, for the velocity gradient, a performance difference similar to that of the vorticity was observed. Notably, neither the velocity p.d.f. nor the two-point correlations show significant differences between the two models when compared to figure 8 and the velocity gradient statistics. Evidently, the small-scale features in the velocity fields are less dominant than those in the vorticity fields; thus, the velocity field is easier to predict than the vorticity field. However, the cGAN model outperformed the CNN in terms of velocity prediction.

Appendix D. Effect of adding an explicit loss to CNN

In this appendix, we present the performance of a CNN model with an explicit loss reflecting the turbulence characteristics. Within the GAN scheme, the generator learns the system characteristics embedded in the data by maximising an implicit loss based on the latent variable in the discriminator. This significantly enhanced PredictionNet compared with the baseline CNN model. However, even the CNN trained solely using MSE loss demonstrated reasonable prediction for relatively short lead times; the issue

of poor performance across all lead times was in the small-scale structures. Hence, we investigate possibility of performance enhancement by explicitly enforcing statistics on the model, without introducing an additional network for adversarial learning. To achieve this, a spectrum loss term has been incorporated into the existing CNN, employing the following formula:

$$\left. \begin{aligned} L_{CNN} &= \sigma_1 \mathbb{E}_{X \sim p(X)} \left[\|Y^* - Y\|_2^2 \right] + \sigma_2 R(w_p) + \sigma_3 L_{spectrum}, \\ L_{spectrum} &= \mathbb{E}_{X \sim p(X)} \left[\left\| \log(\Omega_Y(k)) - \log(\Omega_{Y^*}(k)) \right\|_1 \right], \end{aligned} \right\} \quad (D1)$$

where $\Omega_Y(k)$ and $\Omega_{Y^*}(k)$ represent enstrophy spectra of the target and prediction, respectively. Here σ_3 modulates the strength of spectrum loss and two different values of σ_3 , $0.01/K$ (Spect 0.01) and $0.1/K$ (Spect 0.1), were tested, where the maximum valid wavenumber $K = 64$.

In training models for $T = 0.5T_L$, both cases exhibited convergence in the data loss. However, it is worth noting that during the training process, these cases displayed more fluctuations in the data loss compared with what was observed in figure 6. Figure 23 shows comparison of predicted fields and the corresponding enstrophy spectra that was explicitly given as a loss term in the objective function. In figure 23(a), for a direct comparison, results by cGAN and original CNN for $0.5T_L$ have been extracted from figures 7 and 8 in the main text. Figure 23(b) displays the outcomes for the additional cases. Two noteworthy observations emerge from these figures. First, both Spect 0.01 and Spect 0.1 significantly improved the spectrum when compared with Spect 0. Second, despite the enhancement in spectrum, the visualisations show poor performance compared to Spect 0. When comparing other statistical metrics to those presented in table 2, Spect 0.01 showed slightly inferior performance compared with Spect 0, with CC of 0.9644, MSE of 0.0477 and r.m.s. of 0.9384. Spect 0.1 performed even worse, with CC of 0.9441, MSE of 0.0736 and r.m.s. of 0.9179. It is important to highlight that although the r.m.s. of Spect 0.01 seems closer to the target even than cGAN, it was caused by underpredictions in spectrum at intermediate and small scales and the slight overpredictions at large scale.

We observed that a consideration of an explicit spectrum loss in the training of CNN did not lead to better performance. We investigate the phase information, an essential information along with the spectrum. Figure 24 provides the phase error in which Spect 0.01 exhibits a slightly increased error compared with Spect 0, and Spect 0.1 reveals a significant degradation in the phase error. It can be inferred that explicitly introducing specific statistical characteristics of the system as loss terms in the model might lead to degradation in other statistics. One can potentially enhance performance by adding more explicit losses while conducting finer optimisation. However, this approach requires an extensive parameter optimisation. On the other hand, GANs, by implicitly learning system characteristics through competition with the discriminator, seem to effectively learn the statistical properties of turbulence.

Appendix E. Effect of the disturbance strength on the pre-trained surrogate model

For the training of the control network, PredictionNet (cGAN- $0.5T_L$), was used as a surrogate model. As mentioned in § 4.4, if the strength of the disturbance ΔX is significantly large, then $X + \Delta X$ will deviate from the dynamics of the pre-trained PredictionNet, and the surrogate model will not work properly. Thus, the control effect cannot be properly evaluated as disturbance-added predictions, $\text{Pred}(X + \Delta X) = \tilde{Y}$, are vastly different from the results of the actual simulation, $\mathcal{N}(X + \Delta X)$, with $X + \Delta X$ as the

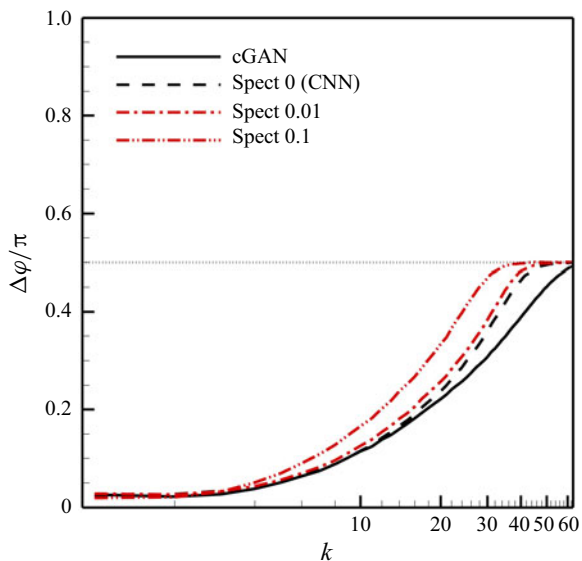


Figure 24. Prediction results of the phase error for $0.5T_L$ comparing cGAN, Spect 0 and spectrum loss added CNNs.

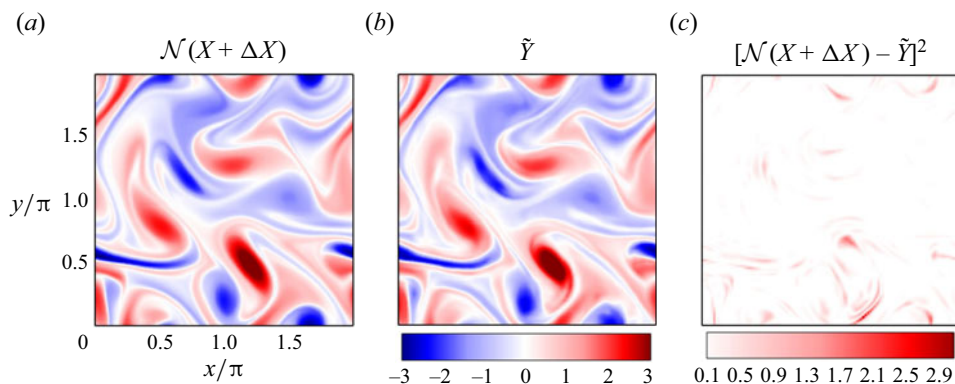


Figure 25. Example of visualising the effect of the disturbance strength on a pre-trained surrogate model using one test dataset with cGAN- $0.5T_L$ and $\Delta X_{rms}=0.2X_{rms}$: (a) disturbance-added simulation ($\mathcal{N}(X + \Delta X)$), (b) disturbance-added prediction ($\text{Pred}(X + \Delta X) = \tilde{Y}$) and (c) squared difference of (a) and (b).

initial condition. Based on the r.m.s. of the input field, \tilde{Y} was compared with $\mathcal{N}(X + \Delta X)$ using disturbances of various strengths ($\Delta X_{rms} = 0.1, 0.5, 1, 5, 10$ and 20% of X_{rms}). We could confirm that the surrogate model works properly for all testing cases. Therefore, an example using one of the test data is presented in [figure 25](#) for the 20% case, which most likely deviates from the dynamics of PredictionNet. The two fields in [figure 25\(a,b\)](#) are qualitatively similar, and most of the area in [figure 25\(c\)](#), which shows the square of the difference between the two, is close to zero. Moreover, the values are not large even at positions where a relatively large error is observed. The MSE in [figure 25\(c\)](#) is 0.0582 , which is reasonably small. In addition, there was little change in the structure of the disturbance field as the disturbance strength changed; there was only a difference in the degree of change in the field over time. Therefore, the disturbance-added solution for a

specific target does not respond sensitively to the disturbance strength within the operating range of the surrogate model.

REFERENCES

- ALEXAKIS, A. & DOERING, C.R. 2006 Energy and enstrophy dissipation in steady state 2D turbulence. *Phys. Lett. A* **359** (6), 652–657.
- ARJOVSKY, M., CHINTALA, S. & BOTTOU, L. 2017 Wasserstein generative adversarial networks. In *34th ICML* (ed. D. Precup & Y.W. Teh), pp. 214–223. PMLR.
- BAE, H.J. & KOUMOUTSAKOS, P. 2022 Scientific multi-agent reinforcement learning for wall-models of turbulent flows. *Nat. Commun.* **13** (1).
- BECK, A. & KURZ, M. 2021 A perspective on machine learning methods in turbulence modeling. *GAMM Mitt.* **44** (1), e202100002.
- BECK, A.D., FLAD, D.G. & MUNZ, C.D. 2018 Deep neural networks for data-driven turbulence models. Preprint, [arXiv:1806.04482](https://arxiv.org/abs/1806.04482).
- BI, K., XIE, L., ZHANG, H., CHEN, X., GU, X. & TIAN, Q. 2023 Pangu-weather: a 3D high-resolution model for fast and accurate global weather forecast. *Nature* **619**, 533.
- BRACHET, M.E., MENEGUZZI, M., POLITANO, H. & SULEM, P.L. 1988 The dynamics of freely decaying two-dimensional turbulence. *J. Fluid Mech.* **194**, 333.
- BRENNER, M.P., ELDREDGE, J.D. & FREUND, J.B. 2019 Perspective on machine learning for advancing fluid mechanics. *Phys. Rev. Fluids* **4**, 100501.
- BRUNTON, S.L., NOACK, B.R. & KOUMOUTSAKOS, P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508.
- COLABRESE, S., GUSTAVSSON, K., CELANI, A. & BIFERALE, L. 2017 Flow navigation by smart microswimmers via reinforcement learning. *Phys. Rev. Lett.* **118** (15), 158004.
- CRESWELL, A., WHITE, T., DUMOULIN, V., ARULKUMARAN, K., SENGUPTA, B. & BHARATH, A.A. 2018 Generative adversarial networks: an overview. *IEEE Signal Process. Mag.* **35** (1), 53–65.
- DAOUD, W.Z., KAHL, J.D. & GHORAI, J.K. 2003 On the synoptic-scale Lagrangian autocorrelation function. *J. Appl. Meteorol.* **42** (2), 318–324.
- DENG, Z., HE, C., LIU, Y. & KIM, K.C. 2019 Super-resolution reconstruction of turbulent velocity fields using a generative adversarial network-based artificial intelligence framework. *Phys. Fluids* **31** (12), 125111.
- DUPUY, D., ODIER, N. & LAPEYRE, C. 2023 Data-driven wall modeling for turbulent separated flows. *J. Comput. Phys.* **487**, 112173.
- DURAISAMY, K., IACCARINO, G. & XIAO, H. 2019 Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51** (1), 357–377.
- DURAISAMY, K., ZHANG, Z.J. & SINGH, A.P. 2015 New approaches in turbulence and transition modeling using data-driven techniques. In *AIAA Paper*, p. 1284. AIAA.
- FOX, D.G. & ORSZAG, S.A. 1973 Pseudospectral approximation to two-dimensional turbulence. *J. Comput. Phys.* **11** (4), 612–619.
- GAMAHARA, M. & HATTORI, Y. 2017 Searching for turbulence models by artificial neural network. *Phys. Rev. Fluids* **2** (5), 054604.
- GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. & BENGIO, Y. 2014 Generative adversarial nets. In *NIPS* (ed. Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence & K.Q. Weinberger), pp. 2672–2680. Curran Associates.
- GOODFELLOW, I., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A. & BENGIO, Y. 2020 Generative adversarial networks. *Commun. ACM* **63** (11), 139–144.
- GUAN, Y., CHATTOPADHYAY, A., SUBEL, A. & HASSANZADEH, P. 2021 Stable a posteriori LES of 2D turbulence using convolutional neural networks: backscattering analysis and generalization to higher Re via transfer learning. Preprint, [arXiv:2102.11400](https://arxiv.org/abs/2102.11400).
- GULRAJANI, I., AHMED, F., ARJOVSKY, M., DUMOULIN, V. & COURVILLE, A. 2017 Improved training of Wasserstein GANs. Preprint, [arXiv:1704.00028](https://arxiv.org/abs/1704.00028).
- HAN, B.Z. & HUANG, W.X. 2020 Active control for drag reduction of turbulent channel flow based on convolutional neural networks. *Phys. Fluids* **32** (9), 095108.
- HENNIGH, O. 2017 Lat-Net: compressing Lattice Boltzmann flow simulations using deep neural networks. Preprint, [arXiv:1705.09036](https://arxiv.org/abs/1705.09036).
- HO, J., JAIN, A. & ABBEEL, P. 2020 Denoising diffusion probabilistic models. *Adv. Neural Inform. Proc. Syst.* **33**, 6840–6851.
- HUSSAIN, A.K.M.F. 1986 Coherent structures and turbulence. *J. Fluid Mech.* **173**, 303–356.
- JIMÉNEZ, J. 2018 Machine-aided turbulence theory. *J. Fluid Mech.* **854**, R1.

Prediction and control of 2-D decaying turbulence using GANs

- JIMÉNEZ, J., MOFFATT, H.K. & VASCO, C. 1996 The structure of the vortices in freely decaying two-dimensional turbulence. *J. Fluid Mech.* **313**, 209–222.
- KARRAS, T., AILA, T., LAINE, S. & LEHTINEN, J. 2017 Progressive growing of GANs for improved quality, stability, and variation. Preprint, [arXiv:1710.10196](https://arxiv.org/abs/1710.10196).
- KATZ, L. 1953 A new status index derived from sociometric analysis. *Psychometrika* **18** (1), 39–43.
- KIM, H., KIM, J. & LEE, C. 2023 Interpretable deep learning for prediction of Prandtl number effect in turbulent heat transfer. *J. Fluid Mech.* **955**, A14.
- KIM, H., KIM, J., WON, S. & LEE, C. 2021 Unsupervised deep learning for super-resolution reconstruction of turbulence. *J. Fluid Mech.* **910**, A29.
- KIM, J., KIM, H., KIM, J. & LEE, C. 2022 Deep reinforcement learning for large-eddy simulation modeling in wall-bounded turbulence. *Phys. Fluids* **34** (10), 105132.
- KIM, J. & LEE, C. 2020a Deep unsupervised learning of turbulence for inflow generation at various Reynolds numbers. *J. Comput. Phys.* **406**, 109216.
- KIM, J. & LEE, C. 2020b Prediction of turbulent heat transfer using convolutional neural networks. *J. Fluid Mech.* **882**, A18.
- KING, R., HENNIGH, O., MOHAN, A. & CHERTKOV, M. 2018 From deep to physics-informed learning of turbulence: diagnostics. Preprint, [arXiv:1810.07785](https://arxiv.org/abs/1810.07785).
- LAM, R., *et al.* 2023 Learning skillful medium-range global weather forecasting. *Science* **382** (6677), 1416–1421.
- LEE, C., KIM, J., BABCOCK, D. & GOODMAN, R. 1997 Application of neural networks to turbulence control for drag reduction. *Phys. Fluids* **9** (6), 1740–1747.
- LEE, S. & YOU, D. 2019 Data-driven prediction of unsteady flow over a circular cylinder using deep learning. *J. Fluid Mech.* **879**, 217–254.
- LEE, T., KIM, J. & LEE, C. 2023 Turbulence control for drag reduction through deep reinforcement learning. *Phys. Rev. Fluids* **8**, 024604.
- LING, J., KURZAWSKI, A. & TEMPLETON, J. 2016 Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166.
- LIU, H.B. & LEE, I. 2020 MPL-GAN: toward realistic meteorological predictive learning using conditional GAN. *IEEE Access* **8**, 93179–93186.
- LOZANO-DURÁN, A. & BAE, H.J. 2023 Machine learning building-block-flow wall model for large-eddy simulation. *J. Fluid Mech.* **963**, A35.
- MAULIK, R., SAN, O., RASHEED, A. & VEDULA, P. 2018 Subgrid modelling for two-dimensional turbulence using neural networks. *J. Fluid Mech.* **858**, 122–144.
- MCWILLIAMS, J.C. 1990 The vortices of two-dimensional turbulence. *J. Fluid Mech.* **219**, 361.
- MCWILLIAMS, J.C., WEISS, J.B. & YAVNEH, I. 1994 Anisotropy and coherent vortex structures in planetary turbulence. *Science* **264** (5157), 410–413.
- MESCHEDER, L., GEIGER, A. & NOWOZIN, S. 2018 Which training methods for GANs do actually converge? In *35th ICML* (ed. J. Dy & A. Krause), pp. 3481–3490. PMLR.
- MEZIĆ, I. 2005 Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dyn.* **41** (1–3), 309–325.
- MEZIĆ, I. 2013 Analysis of fluid flows via spectral properties of the Koopman operator. *Annu. Rev. Fluid Mech.* **45** (1), 357–378.
- MIRZA, M. & OSINDERO, S. 2014 Conditional generative adversarial nets. Preprint, [arXiv:1411.1784](https://arxiv.org/abs/1411.1784).
- MOHAN, A., DANIEL, D., CHERTKOV, M. & LIVESCU, D. 2019 Compressed convolutional LSTM: an efficient deep learning framework to model high fidelity 3D turbulence. Preprint, [arXiv:1903.00033](https://arxiv.org/abs/1903.00033).
- MOHAN, A.T. & GAITONDE, D.V. 2018 A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks. Preprint, [arXiv:1804.09269](https://arxiv.org/abs/1804.09269).
- NAKAMURA, T., FUKAMI, K., HASEGAWA, K., NABAE, Y. & FUKAGATA, K. 2021 Convolutional neural network and long short-term memory based reduced order surrogate for minimal turbulent channel flow. *Phys. Fluids* **33** (2), 025116.
- ORSZAG, S.A. 1971 Numerical simulation of incompressible flows within simple boundaries: accuracy. *J. Fluid Mech.* **49**, 75–112.
- PARISH, E.J. & DURAISAMY, K. 2016 A paradigm for data-driven predictive modeling using field inversion and machine learning. *J. Comput. Phys.* **305**, 758–774.
- PARK, J. & CHOI, H. 2020 Machine-learning-based feedback control for drag reduction in a turbulent channel flow. *J. Fluid Mech.* **904**, A24.
- PARK, J. & LEE, C. 2023 TPTNet: a data-driven temperature prediction model based on turbulent potential temperature. Preprint, [arXiv:2312.14980](https://arxiv.org/abs/2312.14980).

- PARK, T., LIU, M.Y., WANG, T.C. & ZHU, J.Y. 2019 Semantic image synthesis with spatially-adaptive normalization. In *IEEE/CVF Conference on CVPR*.
- RABAULT, J., KUCHTA, M., JENSEN, A., RÉGLADE, U. & CERARDI, N. 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control. *J. Fluid Mech.* **865**, 281–302.
- RABAULT, J. & KUHNLE, A. 2019 Accelerating deep reinforcement learning strategies of flow control through a multi-environment approach. *Phys. Fluids* **31** (9), 094105.
- RAVURI, S., *et al.* 2021 Skilful precipitation nowcasting using deep generative models of radar. *Nature* **597** (7878), 672–677.
- RÜTTGERS, M., LEE, S., JEON, S. & YOU, D. 2019 Prediction of a typhoon track using a generative adversarial network and satellite images. *Sci. Rep.* **9** (1), 1–15.
- SCHMID, P.J. 2010 Dynamic mode decomposition of numerical and experimental data. *J. Fluid Mech.* **656**, 5–28.
- SHI, X., CHEN, Z., WANG, H., YEUNG, D.Y., WONG, W.K. & WOO, W.C. 2015 Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In *NIPS* (ed. C. Cortes, N. Lawrence, D. Lee, M. Sugiyama & R. Garnett), pp. 802–810. Curran Associates.
- SHU, D., LI, Z. & FARIMANI, A.B. 2023 A physics-informed diffusion model for high-fidelity flow field reconstruction. *J. Comput. Phys.* **478**, 111972.
- SINGH, A.P., MEDIDA, S. & DURAISAMY, K. 2017 Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils. *AIAA J.* **55** (7), 2215–2227.
- SIROVICH, L. 1987 Turbulence and the dynamics of coherent structures. Part I. Coherent structures. *Q. Appl. Maths* **45** (3), 561–571.
- SOHL-DICKSTEIN, J., WEISS, E., MAHESWARANATHAN, N. & GANGULI, S. 2015 Deep unsupervised learning using nonequilibrium thermodynamics. In *32nd ICML* (ed. F. Bach & D. Blei), pp. 2256–2265. PMLR.
- SOMMERIA, J. 1986 Experimental study of the two-dimensional inverse energy cascade in a square box. *J. Fluid Mech.* **170**, 139–168.
- SRINIVASAN, P.A., GUASTONI, L., AZIZPOUR, H., SCHLATTER, P. & VINUESA, R. 2019 Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* **4** (5), 054603.
- TANG, H., RABAULT, J., KUHNLE, A., WANG, Y. & WANG, T. 2020 Robust active flow control over a range of Reynolds numbers using an artificial neural network trained through deep reinforcement learning. *Phys. Fluids* **32** (5), 053605.
- VADROT, A., YANG, X.I., BAE, H.J. & ABKAR, M. 2023 Log-law recovery through reinforcement-learning wall model for large eddy simulation. *Phys. Fluids* **35** (5), 055122.
- VERMA, S., NOVATI, G. & KOUMOUTSAKOS, P. 2018 Efficient collective swimming by harnessing vortices through deep reinforcement learning. *Proc. Natl Acad. Sci.* **115** (23), 5849–5854.
- VILLANI, C. 2009 *Optimal Transport*. Springer.
- WANG, Z., XIAO, D., FANG, F., GOVINDAN, R., PAIN, C.C. & GUO, Y. 2018 Model identification of reduced order fluid dynamics systems using deep learning. *Intl J. Numer. Meth. Fluids* **86** (4), 255–268.
- WU, J.L., XIAO, H. & PATERSON, E. 2018 Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. *Phys. Rev. Fluids* **3** (7), 074602.
- YANG, X.I.A., ZAFAR, S., WANG, J.X. & XIAO, H. 2019 Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Phys. Rev. Fluids* **4** (3), 034602.
- YEH, C.A., MEENA, M.G. & TAIRA, K. 2021 Network broadcast analysis and control of turbulent flows. *J. Fluid Mech.* **910**, A15.
- YUAN, X., HE, P., ZHU, Q. & LI, X. 2019 Adversarial examples: attacks and defenses for deep learning. *IEEE Trans. Neural Netw. Learn. Syst.* **30** (9), 2805–2824.
- ZHOU, Z., HE, G. & YANG, X. 2021 Wall model based on neural networks for LES of turbulent flows over periodic hills. *Phys. Rev. Fluids* **6** (5), 054610.
- ZHU, L., ZHANG, W., KOU, J. & LIU, Y. 2019 Machine learning methods for turbulence modeling in subsonic flows around airfoils. *Phys. Fluids* **31** (1), 015105.
- ZHU, P., ABDAL, R., QIN, Y. & WONKA, P. 2020 SEAN: image synthesis with semantic region-adaptive normalization. In *IEEE/CVF Conference on CVPR*.