

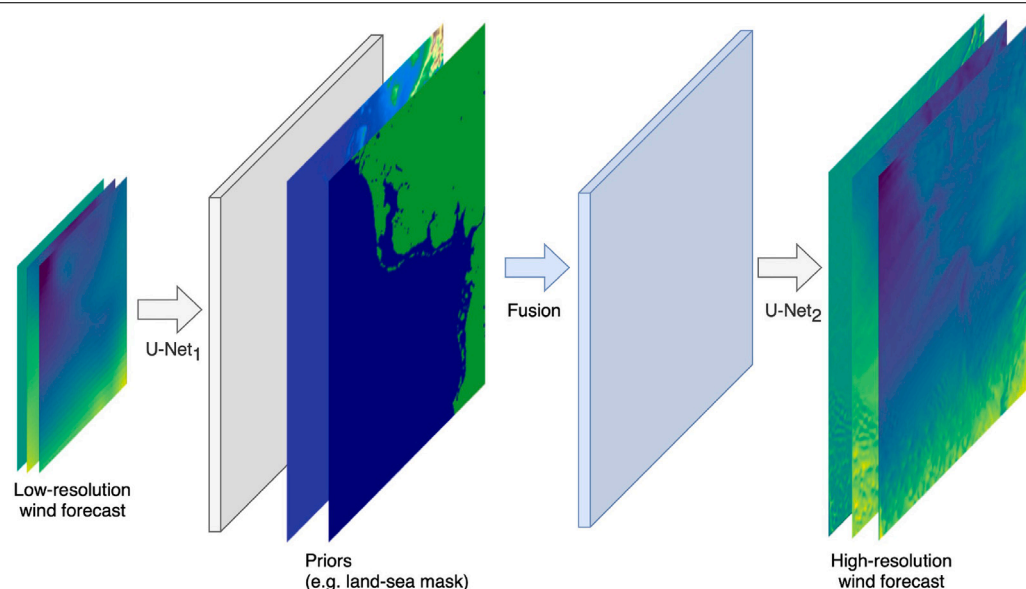
# Domain-informed CNN architectures for downscaling regional wind forecasts

Alexander M. Campbell <sup>a</sup>,<sup>\*</sup>, Simon C. Warder <sup>a</sup>, B. Bhaskaran <sup>b</sup>, Matthew D. Piggott <sup>a</sup>

<sup>a</sup> Department of Earth Science and Engineering, Imperial College London, London, SW7 2AZ, UK

<sup>b</sup> Shell Information Technology International Ltd, Shell Centre, London, SE1 7NA, UK

## GRAPHICAL ABSTRACT



## HIGHLIGHTS

- WRF simulations were generated to explore AI-based downscaling of wind forecasts.
- A sequential U-Net-based downscaling model leverages static fields (e.g. topography).
- A composite Wiener-L1 loss function improves high-frequency signal reconstruction.

## ARTICLE INFO

**Keywords:**  
Weather  
Wind energy  
Downscaling  
CNN

## ABSTRACT

High-resolution wind speed forecasts are of great importance to the wind energy industry, from short-term energy forecasting and trading to longer-term resource assessment and planning. Generating high-resolution regional wind forecasts currently requires compute-intensive numerical models to downscale from a global forecast. Black-box AI models, once trained, can produce results in a fraction of the time and cost; however, they tend to produce smoothed outputs, are not interpretable and generalise poorly. The domain-informed AI architecture presented in this work seeks to address these problems by incorporating prior static fields directly

<sup>\*</sup> Corresponding author.

E-mail address: [alexander.campbell19@imperial.ac.uk](mailto:alexander.campbell19@imperial.ac.uk) (A.M. Campbell).

<https://doi.org/10.1016/j.egyai.2025.100485>

Received 29 September 2024; Received in revised form 4 February 2025; Accepted 5 February 2025

Available online 13 February 2025

2666-5468/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

into the model architecture. Specifically, the proposed approach combines two sequential U-Nets – the first upsamples the input wind fields and expands the number of feature maps, a fusion layer then injects prior static data such as topography, and a second U-Net generates the final output wind field. This approach improves all performance metrics versus a baseline U-Net model and generalises better to out-of-sample scenarios. In addition, this study compares the performance of several loss functions, including standard pixel-wise measures such as mean-squared error, structural similarity and frequency-focused functions, and a function based on Wiener filter theory. All loss functions, with the exception of the Wiener loss, perform comparably and tend to attenuate higher-frequency detail. Although the Wiener loss encourages higher frequencies, it over-estimates amplitudes. A composite Wiener-L1 loss function balances generating high-frequency detail and correctly predicting amplitudes.

## 1. Introduction

### 1.1. Motivation

Numerical weather prediction (NWP) provides a wide variety of important benefits to society, from predicting natural disasters to forecasting solar energy production. NWP also plays a number of crucial roles in the wind energy industry, from short-term energy forecasting to longer-term farm maintenance, planning and array design [1–3]. NWP is an initial value problem: given the current state of the atmosphere, the equations governing atmospheric motion and evolution must be integrated forward in time to predict the future state [4]. This requires state-of-the-art numerical models and supercomputing resources. The spatial and temporal resolution of global NWP models is therefore limited by the computing resources available, with some of the world's largest supercomputers required to generate forecasts. As a result, many key atmospheric features cannot be directly represented by the model and must instead be parameterised; and finer details of the weather field are not resolved, for example, the flow around a wind farm. Increasing the resolution of forecasts is therefore a key goal of the weather and climate community. To generate higher resolution forecasts, NWP models can be run at higher resolutions over limited areas, an approach commonly known as nesting or dynamical downscaling [5]. Whilst this has proven effective, it is also computationally intensive [6].

The exponential growth in the field of Artificial Intelligence (AI) presents an opportunity to develop alternative methods. Once trained, AI-based downscaling “super-resolution” models have the potential to be significantly faster than an equivalent numerical downscaling model. This could allow for faster forecasts and increased resolution operationally versus traditional numerical approaches. Despite this, there are significant challenges in applying AI to emulate dynamical downscaling models. Super-resolution is a challenging and generally ill-posed problem, and standard AI-based approaches tend to over-smooth output predictions and generalise poorly [7,8]. This is of particular concern when predicting high-resolution wind fields over potentially complex terrain. This work addresses this problem by incorporating domain-specific knowledge and data into the modelling framework, and investigating alternative loss functions that have the potential to encourage high-frequency detail reconstruction.

### 1.2. Related work

AI has been successfully applied across many areas within weather forecasting – including data pre-processing and data assimilation, emulating parameterisation schemes, post-processing NWP outputs, and fully AI-based forecast models such as FourCastNet and Pangu-Weather [6,9–11]. AI methods have also been widely used across the wind energy industry, including in energy forecasting [12], turbine condition monitoring and fault detection [13,14], and wake modelling and farm layout optimisation [3,15].

Dynamical downscaling in the field of NWP is the process of increasing the resolution of a forecast using numerical models. A lower-resolution model provides the initial and boundary conditions; then

either a single higher-resolution domain or multiple nested domains at progressively higher grid resolutions, can be used to smoothly transition to a significantly greater resolution than the initial model through integrating the governing differential equations forward in time. This is required when the resolution of a global NWP model is insufficient for the target application, e.g. resolving the finer-scale weather at a wind farm. Due to the large computational resources needed to run such downscaling models, they are generally only run over small regions of the Earth and at limited spatial and temporal resolutions. A number of works have shown AI can be used to downscale a wide variety of weather and climate data [7,16–20]. [16] applied a stacked convolutional neural network (CNN) model to downscale daily precipitation from 100 km to 12.5 km, and also included high-resolution static data. [21] applied CNNs to super-resolution of Landsat data. They compared the Super-resolution CNN (SRCNN) architecture of [22] against a deeper CNN that used residual learning, deep connectivity, and residual connections, finding the latter model performed better. [23] applied a generative model to downscaling WRF wind speed data. They generated the coarse inputs by downsampling the high-resolution target wind speeds by a factor of four. They modified a Generative adversarial Network (GAN) architecture, the Enhanced Super-resolution GAN (ESRGAN) model [24], replacing the adversarial loss with a power spectral density (PSD) loss that penalises differences between the energy spectrum of the prediction and ground truth data. [25] showed the effectiveness of CNNs in generating high-resolution (4 km) near-surface temperature and precipitation data from coarse ERA-interim (approx. 81 km) input data at a daily scale between the years 1981 and 2010. [26] applied an adversarial approach to super-resolving wind and solar fields from global climate models by up to a factor of 50, using a modified version of the super-resolution GAN (SRGAN) model presented by [27]. [28] investigated several CNN architectures of varying complexity for spatially downscaling near-surface wind fields from 27 km to 9 km. They concluded that a U-Net-based model performed optimally and found that including static high-resolution data, such as the land-sea mask and topography, further improved performance. [7] applied CNNs to spatio-temporal downscaling of temperature and precipitation fields. They considered both predicting the high-resolution fields directly, and an error-predicting approach; and apply an image gradient term to the loss function to combat over-smoothing. [29] applied a U-Net-based architecture to spatially reconstruct flow data from satellite observations. They considered a number of additional terms in the loss function, including the velocity Jacobian, a vorticity term, and a zero-divergence term. [30] downscaled from a NWP model to radar precipitation observations, using the complete set of available NWP variables as input. They compared several deep-learning architectures, including U-Nets and Conditional GANs (CGANs), and found CGANs performed optimally. Although generative adversarial approaches have shown promise, such as in [26,30], they are also known to hallucinate detail and tend to be unstable during training [31,32]. Recent work by [33] employed a two-step approach to atmospheric downscaling of several variables alongside new channel synthesis, using a U-Net to predict the mean and a corrector diffusion model to predict the residual.

Physics-informed neural networks (PINNs) have gained substantial attention for solving PDE-based problems, including those in fluid

dynamics [34]. By embedding governing physical laws directly into the loss function, PINNs can effectively use prior knowledge of the system and perform well in limited-data scenarios. Furthermore, in contrast to traditional numerical methods, PINNs are mesh-free and can easily integrate sparse or noisy measurements [35]. Despite these advantages, training PINNs poses a number of challenges. The PDE constraints can lead to a complex optimisation landscape that is highly sensitive to the choice of hyperparameters [36–38], and networks can struggle to propagate information from the initial or boundary conditions to unseen regions or future timesteps [38]. PINNs have been empirically shown to often fail where PDE solutions exhibit high-frequency or multi-scale structure [38–42]. Scaling PINNs to tackle large-scale or high-dimensional problems can also present significant challenges [43]. [38] address some of the limitations of PINNs with the introduction of the physics-informed neural operator (PINO), where they combine PDE constraints in the loss function with neural operator models.

Much work on physics-informed approaches thus far has focused on comparatively simple systems. The NWP downscaling problem examined here involves a complex set of fully compressible, non-hydrostatic atmospheric PDEs, with various additional parameterisation schemes to consider (e.g., microphysics, radiation, boundary-layer physics) [44]. Additionally, the present study focuses on emulating the spatial aspect of downscaling only, at the hub-height of wind turbines, with the numerical model providing a low-resolution physically-consistent spatial grid at each required timestep. Thus, the benefits of using physics-informed approaches here become less obvious. This study therefore concentrates on U-Net-based architectures and loss functions without formal PDE constraints, that could be of immediate practical use in the wind energy industry and quickly re-applied to other NWP setups.

### 1.3. Problem setup

The generic super-resolution problem is outlined here, where  $X \in \mathbb{R}^l$  is a low-resolution input and  $Y \in \mathbb{R}^h$  is a high-resolution output with  $h > l$ . The forward map  $f$  and inverse map  $f^{-1}$  are then given by:

$$\underbrace{f : \mathbb{R}^h \rightarrow \mathbb{R}^l}_{\text{coarsening}}, \quad \underbrace{f^{-1} : \mathbb{R}^l \rightarrow \mathbb{R}^h}_{\text{downscaling}}. \quad (1)$$

The forward map  $f$  provides a unique mapping, but the inverse problem  $f^{-1}$  is generally ill-posed, meaning there are multiple possible high-resolution outputs for a given low-resolution input [7]. In the field of AI, the inverse map  $f^{-1}$  is parameterised by a data-driven model such as a neural network:

$$f_{\theta}^{-1} : \mathbb{R}^l \rightarrow \mathbb{R}^h, \quad (2)$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta), \quad (3)$$

where  $\theta$  is a set of learned network parameters and  $\mathcal{L}$  represents a loss function to be minimised. To limit complexity and computational requirements, the problem is constrained to the 2-D horizontal domain, i.e. downscaling in space only, at a single altitude. There are therefore  $(x_i, y_i)^N$  pairs of input and output data, where  $x_i$  is a low-resolution image patch with  $C \times P \times Q$  grid points and  $y_i$  is the corresponding high-resolution patch with  $\bar{C} \times \bar{P} \times \bar{Q}$  grid points,  $k_x$  and  $k_y$  are the downscaling factors such that  $\bar{P} = k_x P$  and  $\bar{Q} = k_y Q$ , and  $N$  is the number of samples. Each pair  $(x_i, y_i)$  is defined at a single point in time  $t$  and height  $z$ ;  $C$  corresponds to the channels, here the wind vector components with  $\bar{C} = C$ .

## 2. Methodology

### 2.1. Data generation

The Weather Research and Forecasting (WRF) model was used in this study to generate high-resolution simulations centred over the

**Table 1**

Summary of the final WRF setup used in this work.

| WRF                                       |                                          |
|-------------------------------------------|------------------------------------------|
| Version                                   | 4.4.1                                    |
| Initialisation data                       | ERA5                                     |
| Projection                                | Lambert                                  |
| Domains                                   | 3: D01, D02, D03 (Fig. 1(a))             |
| Horizontal grid spacing                   | 27 km, 9 km, 1.8 km                      |
| Timestep <sup>a</sup>                     | 90 s, 40 s, 8 s                          |
| Horizontal grid size                      | 129 × 129, 163 × 163, 291 × 291          |
| Vertical levels                           | 62, with top at 5000 Pa                  |
| Simulation length                         | 24 h, plus 12 h spin-up (total 36 h)     |
| Simulation period                         | 01/01/2008–12/31/2008                    |
| Physics and dynamics schemes <sup>b</sup> |                                          |
| Microphysics                              | Goddard 4-ice                            |
| Cumulus                                   | Multi-scale Kain-Fritsch (D01, D02 only) |
| Surface Layer                             | MM5 surface layer                        |
| Land Surface                              | Noah-MP                                  |
| Planetary boundary layer                  | Yonsei University (YSU)                  |
| Long-wave radiation                       | Goddard LW                               |
| Short-wave radiation                      | Goddard SW                               |
| MYNN TKE advection <sup>c</sup>           | False                                    |

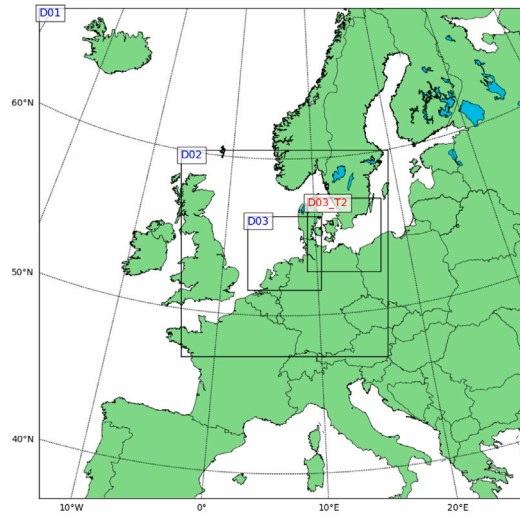
<sup>a</sup> Starting timestep in adaptive scheme.

<sup>b</sup> Unless specified, the scheme is the same for all domains.

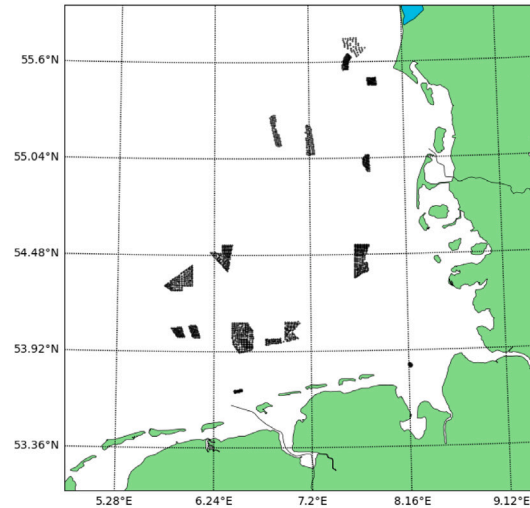
<sup>c</sup> Tested during sensitivity study, requires MYNN PBL scheme.

German Bight region of the North Sea (Fig. 1(a)). This location is a major hub for wind farm development, so is an appropriate choice for investigating downscaling NWP models for wind energy applications (Fig. 1(b)). WRF is a fully compressible non-hydrostatic atmospheric model, and supports massively-parallel computation across many platforms [44,45]. The WRF setup used in this paper is based on published best-practices and previous work (e.g. [46–50]), and was established following a sensitivity study of the key physical parameterisation schemes (not shown). The final chosen setup is summarised in Table 1. WRF simulations were run from 01/01/2008 to 31/12/2008. ECMWF's ERA5 reanalysis dataset was used to generate initial and boundary conditions for the WRF simulations [51]. A simulation was initialised every 24 h starting at UTC 00:00 and lasting for a total of 36 h. The first 12 h were discarded as spin-up time, resulting in consecutive 24-hour simulations for the year. Three nested domains were used, at horizontal grid resolutions of 27 km, 9 km and 1.8 km resolution (D01, D02 and D03, Fig. 1(a)). To evaluate the AI models and assess their generalisation ability across different locations, an additional WRF dataset configured identically but centred over a different area was generated for the period 01/12/2008–31/12/2008 (D03\_T2, Fig. 1(a)).

To assess AI model performance across several scales and complexities, processed datasets are generated from the WRF data at multiple downscaling factors. In each case, the inner domain D03 is patched to arrays of  $256 \times 256$  grid points, and average pooling is applied to reduce the patch sizes by factors of 2, 4, 8 and 16 in each dimension. These patches are extracted at single slices in time and at 100 m height above sea/ground, as this is a standard wind turbine hub-height. The resultant datasets are summarised in Table 2. Every increase in the downsampling factor, from 2 to 16, represents more data loss and therefore a more challenging downscaling problem. Each resultant dataset is split into training (01/01/2008–30/04/2008, 01/06/2008–30/11/2008), validation (01/05/2008–31/05/2008) and two testing sets (01/12/2008–31/12/2008), one at the same location as during training (D03) and the other centred over a different area (D03\_T2). The input and target variables are the three components of the wind vector -  $u$ ,  $v$ , and  $w$ ; the high-resolution land-sea mask and topography variables are used as additional inputs to the domain-informed models. Channel-wise min-max normalisation is applied to the datasets, based on the training data statistics.



(a) WRF domain configuration consisting of three nested domains. The primary WRF setup is labelled in blue - D01, D02 and D03. The second testing domain D03\_T2 is labelled in red.



(b) Wind turbine arrays within the study domain D03. Wind turbines are shown as black dots; turbine locations were obtained from [52].

**Fig. 1.** WRF domain configuration and relevant wind farms located in the inner WRF domain (see [52]).

**Table 2**

Summary of the different datasets generated. In each case, the number of channels (variables) is not changed, being the three components of the wind vector for the input and target, plus the high-resolution land-sea mask and topography.

| Input grid<br>(Width × Height) | Target grid<br>(Width × Height) | Downsampling<br>ratio | Total samples<br>(train./val./test.) |
|--------------------------------|---------------------------------|-----------------------|--------------------------------------|
| 128 × 128                      | 256 × 256                       | 1:2                   | 57624/5684/6076                      |
| 64 × 64                        | 256 × 256                       | 1:4                   | 57624/5684/6076                      |
| 32 × 32                        | 256 × 256                       | 1:8                   | 57624/5684/6076                      |
| 16 × 16                        | 256 × 256                       | 1:16                  | 57624/5684/6076                      |

## 2.2. Models

### 2.2.1. Bilinear interpolation

Bilinear interpolation of the input grid is included as a naive baseline approach for super-resolution. Bilinearly interpolating an array  $\mathbf{X}$  to a given coordinate  $(a, b)$  can be written as a weighted sum of the four discrete neighbouring points  $[a, b]$  [53]:

$$\mathbf{X}(a, b) = \sum_{(x, y) \in [a, b]} w_{xy} \cdot \mathbf{X}[x, y], \quad (4)$$

where  $w_{xy}$  are the weighting coefficients based on the relative distances between coordinates  $(a, b)$  and  $(x, y)$ . This simple approach preserves spatial structure but attenuates higher frequency features, resulting in smoothed output predictions [53].

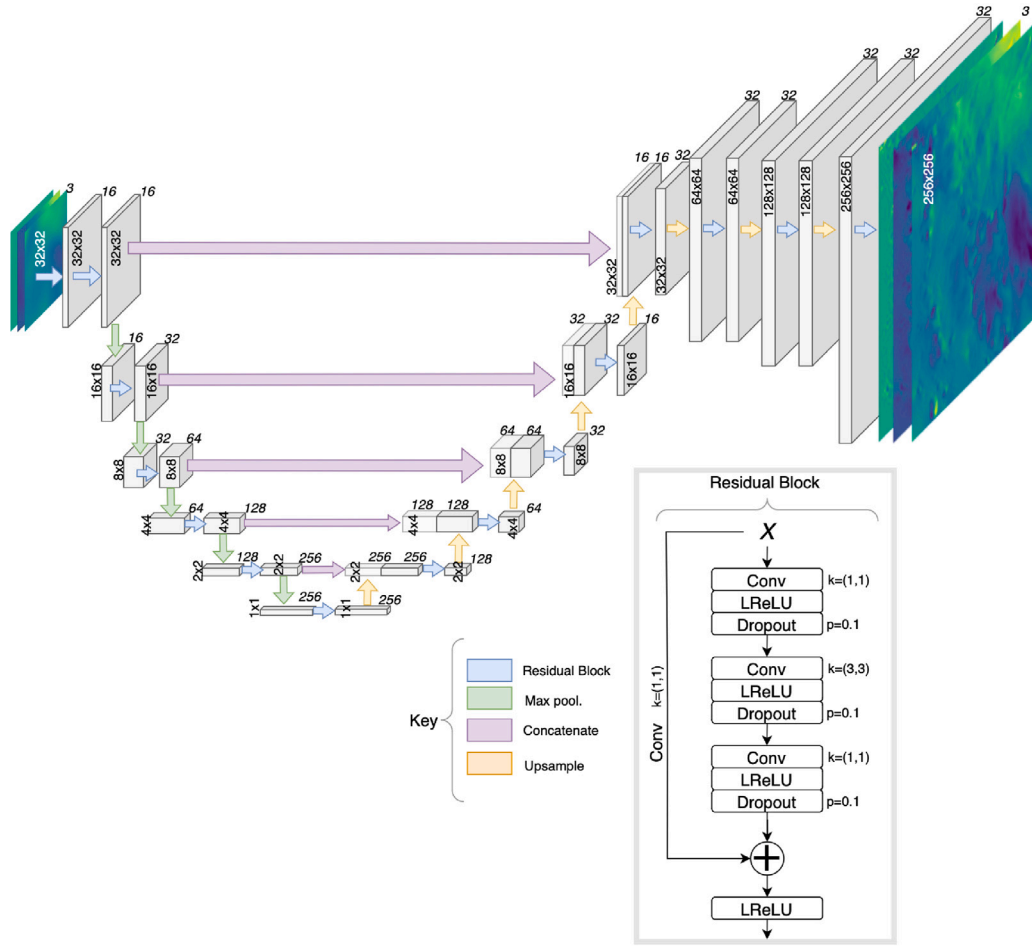


Fig. 2. Example U-Net and upsampling structure in the case of an input of size  $32 \times 32$ , with the maximum channels capped at 256. The U-Net takes an input of size  $32 \times 32$  with 3 channels and outputs an image of the same size, but with 32 channels. This is followed by an expansion block containing several inter-leaved residual blocks (blue) and upsampling layers (yellow) to increase the array size and reduce the number of channels, giving an output of  $256 \times 256$  and 3 channels.

### 2.2.2. U-Net

The U-Net model is a well-known encoder–decoder CNN architecture originally developed for segmenting structures in medical images [54]. The U-Net architecture used in this study is similar to that used by [55], consisting of a contractive part with multiple stages of convolutional residual blocks and max-pooling layers, followed by an expansive part where max-pooling is replaced with bilinear upsampling. The model architecture is summarised in Fig. 2. Each residual block is composed of 3 convolutional layers ( $1 \times 1$ ,  $3 \times 3$ ,  $1 \times 1$ ), and Leaky Rectified Linear Unit (LReLU) activation functions between each convolution. In addition, channel-wise dropout is incorporated within the residual blocks, i.e. where entire channels are switched off randomly with probability  $p$  sampled from a Bernoulli distribution. Dropout is not applied in the first and last blocks of the model, to avoid overly-strong regularisation. Skip connections (Fig. 2, purple arrows) are used to avoid loss of information during the encoding process by concatenating features from the corresponding layer in the encoder with the upsampled features in the decoder. This procedure allows the model to access information at different scales.

### 2.2.3. W-Net

The topography and the land–sea mask are static, and known in advance of a forecast at a high-resolution. Wind speed is strongly correlated with topography near the surface, and wind fields display different characteristics over sea versus land; therefore it follows that injecting these fields has the potential to improve model predictions

and generalisability to different areas. To incorporate these fields into the standard “black-box” U-Net architecture, the model is expanded to use two sequential U-Nets and a fusion layer is developed to add the priors, i.e. the topography and land–sea mask, in an explainable way. This approach, termed a “W-Net” here, is summarised in Fig. 3a. The first U-Net receives the low-resolution array as input and generates an output array at the target grid size, as shown in Fig. 2, whilst also increasing the number of channels. A fusion layer then combines this with the priors (Fig. 3b), before passing the output onto the second U-Net which generates the final prediction. The fusion layer consists of a residual block that expands the priors to the same number of channels as the output of the first U-Net. These are then concatenated together and passed through a channel-wise attention mechanism (CAM), followed by a second residual block to reduce the number of channels.

The attention mechanism is based on the channel attention module outlined in [56]. However, here a softmax function is instead used during attention, such that each learned channel weight can be interpreted as a relative importance and explicitly indicates the value added by incorporating the priors versus the low-resolution input wind fields. In the CAM, the input is reduced to a single pixel but retains the same number of channels, via max-pooling and average pooling, to give 1D arrays which are passed to a shared multi-layer perceptron (MLP). The outputs are added together and passed through a softmax activation function, and at this point can be considered learnt channel weights which are used to weight each channel of the input  $X_{in}$  (Fig. 3b).

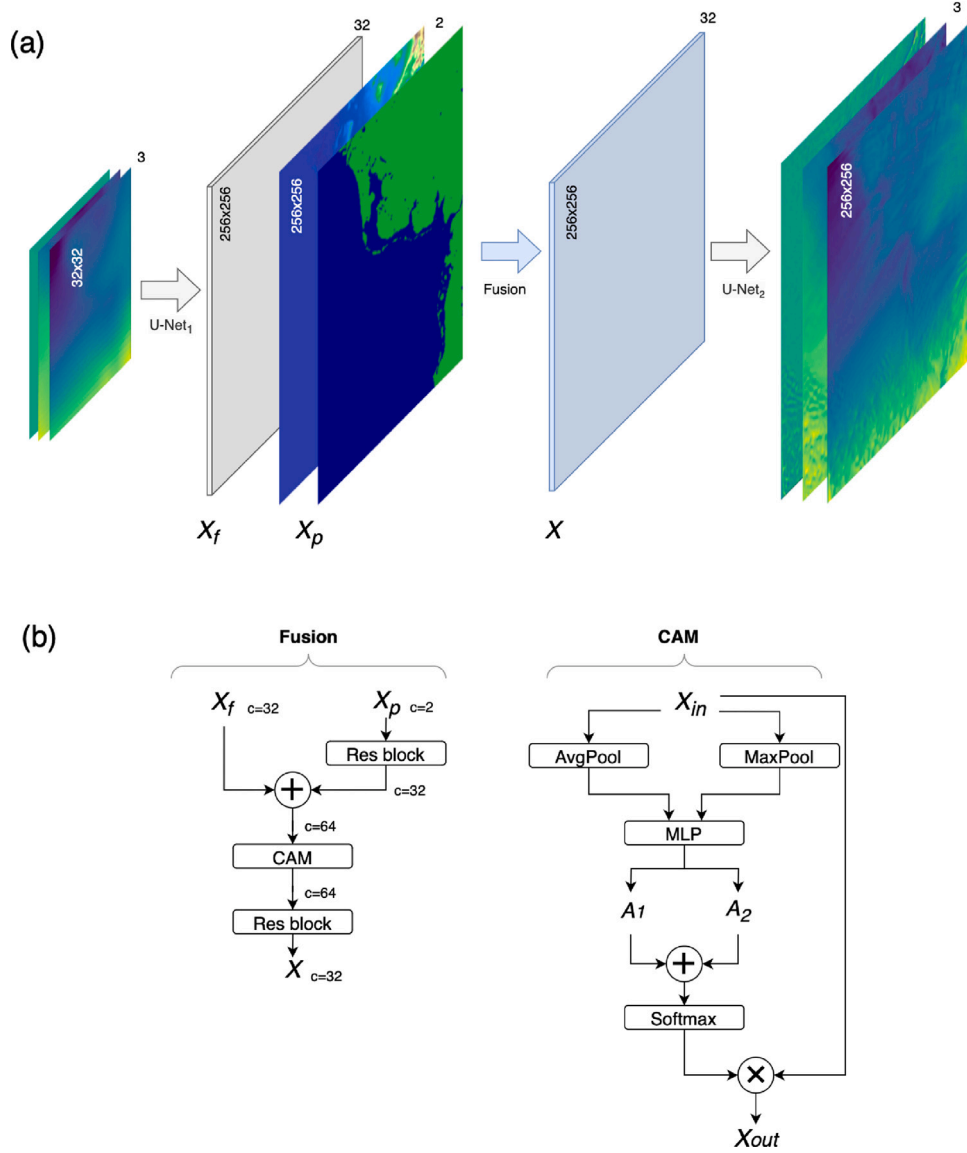


Fig. 3. The W-Net architecture proposed in this work. The first U-Net upsamples the low-resolution wind variables to a larger number of feature maps  $X_f$ ; this is then fused with the prior static variables  $X_p$ , and finally  $X$  is passed through a second U-Net to generate the high-resolution wind prediction. The number of channels is denoted by  $c$  in part (b).

### 2.3. Loss functions & evaluation metrics

#### 2.3.1. Mean absolute error & mean squared error

Two standard loss functions for super-resolution problems in AI are the Mean Absolute Error (MAE)  $\mathcal{L}_{\ell_1}$  and Mean Squared Error (MSE)  $\mathcal{L}_{\ell_2}$ , which for a single image pair are defined as:

$$\mathcal{L}_{\ell_1}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|, \quad (5)$$

$$\mathcal{L}_{\ell_2}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2, \quad (6)$$

where  $i$  is the index of the pixel,  $\hat{y}$  is the predicted image and  $y$  is the ground-truth. However, these loss functions assume pixel-wise independence and that all data points are of equal importance [57,58]. As a result, these functions tend to be insensitive to global structure and texture, and instead focus on properties of individual pixels without considering neighbouring pixels [59]. This also tends to result in blurring of the output predictions of the model.

#### 2.3.2. Structural similarity index measure

One approach to combat this is to consider measures that penalise differences in image structure, such as the Structural Similarity Index Measure (SSIM). Specifically, the SSIM consists of three components – a luminance (mean) term  $l$ , a contrast (variance) term  $c$ , and a correlation term  $s$  [60]:

$$l(x, y) = \frac{2\bar{x}\bar{y} + c_1}{\bar{x}^2 + \bar{y}^2 + c_1}, \quad (7)$$

$$c(x, y) = \frac{2s_x s_y + c_2}{s_x^2 + s_y^2 + c_2}, \quad (8)$$

$$s(x, y) = \frac{s_{x,y} + c_3}{s_x s_y + c_3}, \quad (9)$$

with the Structural Similarity Index Measure defined as:

$$\text{SSIM}(x, y) = l(x, y)c(x, y)s(x, y). \quad (10)$$

$\bar{x}$ ,  $\bar{y}$ ,  $s_x^2$ ,  $s_y^2$ , and  $s_{x,y}$  represent, respectively, the local mean of  $x$  and  $y$ , the local variance of  $x$  and  $y$ , and the local covariance between  $x$  and  $y$ . Constants  $c_1$ ,  $c_2$ , and  $c_3$  characterise the saturation effects of the visual system at low luminance and contrast regions and maintain numerical

stability when the denominators approach zero [60]. Choosing  $c_3 = c_2/2$ , the SSIM is as follows [60]:

$$\text{SSIM}(x, y) = \frac{(2\bar{x}\bar{y} + c_1)(2s_{x,y} + c_2)}{(\bar{x}^2 + \bar{y}^2 + c_1)(s_x^2 + s_y^2 + c_2)}, \quad (11)$$

where  $-1 \leq \text{SSIM}(x, y) \leq 1$  and  $\text{SSIM}(x, y) = 1 \iff x = y$ , i.e. the more similar  $x$  and  $y$  are to each other, the closer  $\text{SSIM}(x, y)$  is to 1. The SSIM is computed over local patches of the images, and then averaged to produce an overall SSIM. Means and standard deviations are computed using a Gaussian filter  $G_{\sigma_G}$  with standard deviation  $\sigma_G$  [61]. Here, a kernel size of 11 is used, with  $\sigma_G = 1.5$ ,  $c_1 = (K_1 L)^2$  and  $c_2 = (K_2 L)^2$ , where  $L$  is the dynamic range of the pixel values,  $K_1 = 0.01$  and  $K_2 = 0.03$  [62]. The overall Mean SSIM (MSSIM) across the whole image is then:

$$\text{MSSIM}(x, y) = \frac{1}{M} \sum_{i=1}^M \text{SSIM}(x_i, y_i), \quad (12)$$

where  $x_i$  and  $y_i$  are the image contents at the  $i$ th local window and  $M$  is the number of local windows covering the full image. Whilst the SSIM is generally used as a metric, it is differentiable and therefore can be used within a loss function, which is defined here as:

$$\mathcal{L}_{\text{SSIM}}(y, \hat{y}) = \frac{1 - \text{MSSIM}(y, \hat{y})}{2}. \quad (13)$$

### 2.3.3. Focal frequency loss

An alternative approach is to consider differences in the frequency domain. [23] penalise differences in power spectral density (PSD), yet this only accounts for errors in amplitudes. Instead, motivated by the idea that penalising differences in the frequency domain could improve image quality and that more attention should be paid to hard-to-synthesise frequency components, [63] propose the Focal Frequency loss (FFL). The FFL function considers both amplitude and phase, and adaptively focuses on hard-to-synthesise frequency components by down-weighting easy ones. Their derivation of the FFL is summarised below; for additional details and a PyTorch implementation the reader is referred to [63]. [63] first propose a distance metric to quantify the difference between predicted and target images in the frequency domain, based on the 2D discrete Fourier transform. Defining the 2D discrete Fourier transform as follows [63]:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cdot \left[ \cos 2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right) - i \sin 2\pi \left( \frac{ux}{M} + \frac{vy}{N} \right) \right], \quad (14)$$

where the image size is  $M \times N$ ;  $(x, y)$  denotes the coordinate of an image pixel in the spatial domain;  $f(x, y)$  is the pixel value;  $(u, v)$  represents the coordinate of a spatial frequency on the frequency spectrum;  $F(u, v)$  is the complex frequency value. Letting  $R(u, v) = a$  and  $I(u, v) = b$  be the real and the imaginary part of  $F(u, v)$  respectively,  $F(u, v)$  can then be written as:

$$F(u, v) = R(u, v) + I(u, v)i = a + bi. \quad (15)$$

[63] then map each frequency value to a Euclidean vector in two-dimensional space, where the real and imaginary components correspond to the  $x$ -axis and  $y$ -axis, respectively. Given the frequency value at the spectrum coordinate  $(u, v)$  of a real image  $F_r(u, v)$  and corresponding predicted image  $F_f(u, v)$  (Eq. (15)), their respective vectors are then given as  $\mathbf{r}_r$  and  $\mathbf{r}_f$  (Fig. 4). Considering the definitions of amplitude and phase:

$$|F(u, v)| = \sqrt{R(u, v)^2 + I(u, v)^2} = \sqrt{a^2 + b^2}, \quad (16)$$

$$\angle F(u, v) = \arctan \left( \frac{I(u, v)}{R(u, v)} \right) = \arctan \left( \frac{b}{a} \right), \quad (17)$$

it follows that amplitudes are given by the vector magnitudes  $|\mathbf{r}_r|$  and  $|\mathbf{r}_f|$ , and phases by the angles  $\theta_r$  and  $\theta_f$  (Fig. 4). The distance for a single frequency is then:

$$d(\mathbf{r}_r, \mathbf{r}_f) = \|\mathbf{r}_r - \mathbf{r}_f\|_2^2 = |F_r(u, v) - F_f(u, v)|^2. \quad (18)$$

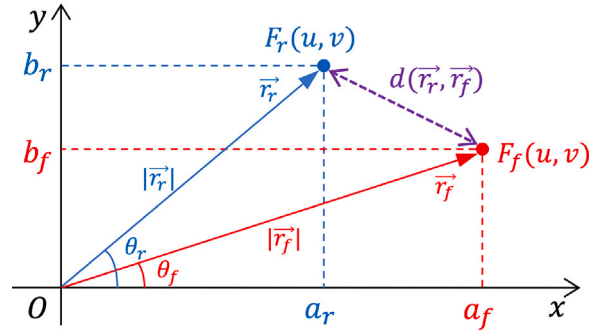


Fig. 4. Frequency distance (purple line, Eq. (18)) between the vectors  $\mathbf{r}_r$  and  $\mathbf{r}_f$  mapped from corresponding real and predicted frequency values  $F_r(u, v)$  and  $F_f(u, v)$ . Figure is from [63].

[63] then define the overall frequency distance between the real and predicted images as:

$$d(F_r, F_f) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} |F_r(u, v) - F_f(u, v)|^2. \quad (19)$$

Next, [63] introduce a spectrum weight matrix to down-weight frequencies that are easier to predict. This matrix is dynamically determined by a non-uniform distribution on the current loss of each frequency during the training process, with the matrix weight element for the spatial frequency at  $(u, v)$  defined as [63]:

$$w(u, v) = |F_r(u, v) - F_f(u, v)|^\alpha, \quad (20)$$

where  $\alpha$  is a scaling factor ( $\alpha = 1$  here) and matrix values are normalised into the range  $[0, 1]$ . The final FLL loss function is then given by [63] as:

$$\mathcal{L}_{\text{FFL}} = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} w(u, v) |F_r(u, v) - F_f(u, v)|^2. \quad (21)$$

### 2.3.4. Wiener loss

[59] present an alternative novel loss function utilising Wiener filter theory. Commonly used in image processing for denoising and deblurring applications, [59] instead apply Wiener filters to compare data samples in a globally correlated manner. The intuition behind their approach is briefly outlined here, and the reader is referred to [59] for a full theoretical derivation and to their publicly available PyTorch implementation for practical details. Defining a Wiener filter  $\mathbf{v}$  as the convolutional filter providing the best least-squares match between two signals  $\mathbf{x}$  and  $\mathbf{y}$ , expressed as column vectors, calculating this filter  $\mathbf{v}$  corresponds to minimising the following expression with respect to  $\mathbf{v}$ :

$$\min_{\mathbf{v}} \frac{1}{2} \|\mathbf{Y}\mathbf{v} - \mathbf{x}\|^2, \quad (22)$$

where  $\mathbf{Y}$  is a Toeplitz matrix that represents a convolution with signal  $\mathbf{y}$ . Eq. (22) can be solved efficiently in the Fourier-domain as follows:

$$\mathbf{v}(\mathbf{x}, \mathbf{y}) = \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}\{\mathbf{x}\} \otimes \mathcal{F}\{\mathbf{y}\} + \lambda}{\mathcal{F}\{\mathbf{x}\} \otimes \mathcal{F}\{\mathbf{x}\} + \lambda} \right\}, \quad (23)$$

where  $\mathcal{F}$  represents the fast Fourier transform (FFT),  $\otimes$  represents the Hadamard product, the vinculum (horizontal line) represents Hadamard division, and  $\lambda$  is a pre-whitening scalar used to stabilise the deconvolution. If  $\mathbf{x}$  and  $\mathbf{y}$  are identical, then the Wiener filter best matching them is given by the Dirac delta function  $\delta$  (the identity function of the convolution operation). This is illustrated in Fig. 5 where in the bottom row  $\mathbf{x}$  and  $\mathbf{y}$  are equal and thus  $\mathbf{v}$  is equal to  $\delta$ .

To move predicted model samples towards the target samples in a global sense, [59] seek to define a loss that maximises the likelihood of  $\delta$  under a multivariate Gaussian distribution with a mean defined as data-matching full-lag Wiener filters, this likelihood is defined as

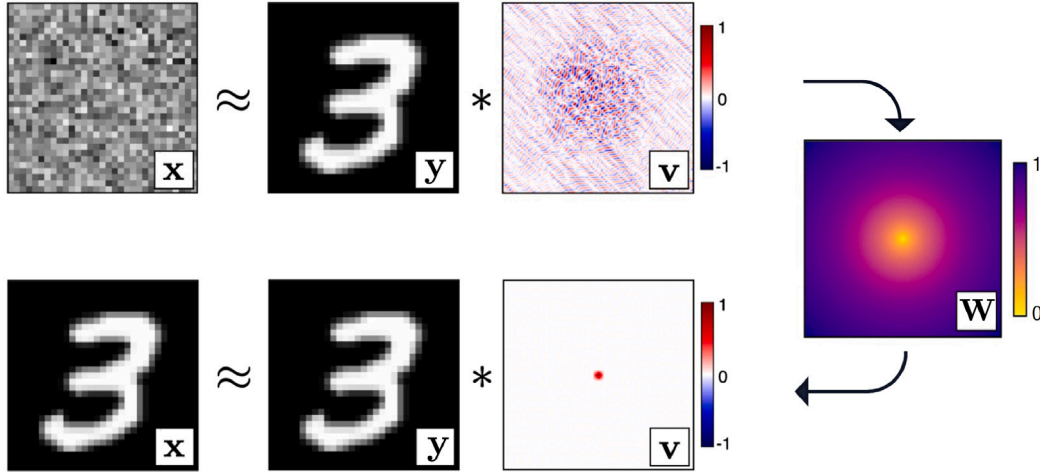


Fig. 5. An illustration of the Wiener loss approach (Eq. (26)) applied to the MNIST dataset, modified from [59]. Top: matching sample  $y$  to random noise  $x$ . Bottom: matching  $y$  to  $x$  where  $x = y$ , showing that the Wiener filter  $\mathbf{v}$  is the Dirac delta function when  $x = y$ . The matrix  $\mathbf{W}$  shown in Eq. (26) is given as a Laplace density function here.

follows:

$$p(\delta | \mathbf{v}) = \prod_{i=1} \mathcal{N}(\delta | \mathbf{v}(\mathbf{x}_{\theta}^{(i)}, \mathbf{y}^{(i)}), \Sigma), \quad (24)$$

where  $\mathbf{x}_{\theta}^{(i)}$  is the  $i$ th predicted sample from a model with parameters  $\theta$ ,  $\mathbf{y}^{(i)}$  is the  $i$ th target sample, and  $\Sigma$  is the covariance matrix. Taking the negative log-likelihood and neglecting terms that are not dependent on the Wiener filters yields:

$$-\log p(\delta | \mathbf{v}) \propto \sum_{i=1} \mathcal{L}_w(\mathbf{x}_{\theta}^{(i)}, \mathbf{y}^{(i)}), \quad (25)$$

where the Wiener loss  $\mathcal{L}_w$  is defined as:

$$\mathcal{L}_w(\mathbf{x}_{\theta}, \mathbf{y}) = \frac{1}{2} \|\mathbf{W} \{\mathbf{v}(\mathbf{x}_{\theta}, \mathbf{y}) - \delta\}\|^2, \quad (26)$$

with the inverse covariance matrix decomposing as  $\Sigma^{-1} = \mathbf{W}^T \mathbf{W}$ . The matrix  $\mathbf{W}$  can be considered a whitening matrix that is tuneable; in this work a two-dimensional Laplace density function is used after [59] (shown in Fig. 5). The resultant Wiener loss function retains a number of desirable optimisation qualities, including smoothness, convexity and differentiability [59].

### 2.3.5. Composite loss

Any of these individual loss functions (Eqs. (5), (6), (13), (21) and (26)) may complement each other and be combined into a composite loss function:

$$\mathcal{L} = (1 - \alpha)\mathcal{L}_A + \alpha\mathcal{L}_B, \quad (27)$$

where  $\alpha$  is a tuneable hyperparameter representing the relative weight between two loss functions  $\mathcal{L}_A$  and  $\mathcal{L}_B$ .

### 2.3.6. Evaluation metrics

For performance evaluation, in addition to reporting the MAE, MSE and SSIM, the peak signal-to-noise ratio (PSNR) is also provided. For a given image  $y$ , the PSNR is defined as:

$$\text{PSNR}(y) = 10 \cdot \log_{10} \left( \frac{\text{MAX}_I^2(y)}{\mathcal{L}_{\ell_2}(\hat{y}, y)} \right), \quad (28)$$

where  $\text{MAX}_I$  is the maximum possible pixel value of the image and  $\mathcal{L}_{\ell_2}$  is the loss defined by Eq. (6).

## 2.4. Training procedure & hyperparameters

Pytorch's DistributedDataParallel framework was used to distribute model training and hyperparameter optimisation across multiple Nvidia

Table 3

A summary of the final tuned hyperparameters used for the U-Net and W-Net architectures, where  $N_f$  is a cap for the maximum number of channels in a given convolutional layer,  $\alpha$  is the weight for the Wiener-L1 composite function, and Max Norm refers to the maximum gradient norm value used in gradient clipping. Note that  $N_f$  was capped to limit the number of model parameters, and  $\alpha$  was chosen heuristically.

| Hyperparameter       | Value              |
|----------------------|--------------------|
| Batch size           | 32                 |
| Learning rate        | $5 \times 10^{-4}$ |
| Dropout probability  | 0.1                |
| $N_f$ (max channels) | 256                |
| Activation function  | LReLU              |
| Optimiser            | Adam               |
| Epochs               | 200                |
| $\alpha$             | 0.3                |
| Max Norm (clipping)  | 10                 |

A100 40 GB GPUs on a DGX cluster. Just-in-time compilation and mixed precision training were used to reduce training time and memory requirements. A random search approach was applied to tune hyperparameters in the *Weights & Biases* (wandb) framework, the final set of tuned hyperparameters for the U-Net and W-Net models is presented in Table 3. Due to computational limitations, hyperparameters were tuned for models downscaling by a factor of eight, and assumed to be reasonable for all factors. A learning rate of  $5 \times 10^{-4}$  was used with the Adam optimiser, and models were trained for a total of 200 epochs. Unless specified otherwise, a  $\mathcal{L}_{\ell_1}$  loss function (Eq. (5)) was used. The maximum number of channels for each U-Net/W-Net was capped at 256 and the first residual block expanded all inputs to 16 channels. A channel-dropout probability of  $p = 0.1$  was identified, and applied within all residual blocks, except the first and last blocks in each U-Net, and the first block in the fusion layer. For the composite Wiener-L1 function (Section 3.2), the relative weight of each term  $\alpha = 0.3$  (Eq. (27)) was chosen heuristically, giving a weighting of 0.3 and 0.7 for the Wiener and L1 terms respectively.

## 3. Results & discussion

### 3.1. Model variants

Summary metrics of model performance on each unseen test dataset, derived from WRF domains D03 and D03\_T2 as shown in Fig. 1(a), are shown in Tables 4 and 5 respectively, with the best metrics for each

**Table 4**

Summary of the results for each downscaling factor for test set D03. Each model was trained using a L1 loss function. For each factor, the best result is highlighted in bold.

| Model    | Factor | MSE                                     | MAE                                     | PSNR        | SSIM         |
|----------|--------|-----------------------------------------|-----------------------------------------|-------------|--------------|
| U-Net    | 2      | $4.75 \times 10^{-6}$                   | $1.08 \times 10^{-3}$                   | 53.5        | 0.998        |
| W-Net    | 2      | <b><math>1.60 \times 10^{-6}</math></b> | <b><math>7.72 \times 10^{-4}</math></b> | <b>58.2</b> | <b>0.999</b> |
| Bilinear | 2      | $3.57 \times 10^{-5}$                   | $3.07 \times 10^{-3}$                   | 44.9        | 0.981        |
| U-Net    | 4      | $4.55 \times 10^{-5}$                   | $3.81 \times 10^{-3}$                   | 43.7        | 0.973        |
| W-Net    | 4      | <b><math>2.59 \times 10^{-5}</math></b> | <b><math>2.98 \times 10^{-3}</math></b> | <b>46.5</b> | <b>0.983</b> |
| Bilinear | 4      | $1.05 \times 10^{-4}$                   | $5.74 \times 10^{-3}$                   | 40.2        | 0.943        |
| U-Net    | 8      | $1.14 \times 10^{-4}$                   | $6.68 \times 10^{-3}$                   | 39.9        | 0.934        |
| W-Net    | 8      | <b><math>9.38 \times 10^{-5}</math></b> | <b><math>5.96 \times 10^{-3}</math></b> | <b>40.9</b> | <b>0.944</b> |
| Bilinear | 8      | $1.66 \times 10^{-4}$                   | $7.75 \times 10^{-3}$                   | 38.3        | 0.921        |
| U-Net    | 16     | $1.73 \times 10^{-4}$                   | $8.37 \times 10^{-3}$                   | 38.2        | 0.920        |
| W-Net    | 16     | <b><math>1.53 \times 10^{-4}</math></b> | <b><math>7.83 \times 10^{-3}</math></b> | <b>38.9</b> | <b>0.929</b> |
| Bilinear | 16     | $2.29 \times 10^{-4}$                   | $9.50 \times 10^{-3}$                   | 37.0        | 0.914        |

downscaling factor highlighted in bold. Metrics are reported based on the normalised data. Example predictions are shown for each downscaling factor in Figs. 6–9 for test set D03, and Figs. 11–14 for test set D03\_T2.

### 3.1.1. Test set D03

The first test set is derived from WRF domain D03, which is the same domain as used for training the downscaling models. In every case, the W-Net model outperforms all other methods across all metrics, whilst the U-Net model also significantly improves upon baseline bilinear interpolation. Performance drops for all models as the downscaling factor is increased. At lower downscaling factors ( $\leq 4\times$ , Figs. 6–7), more detail is retained in the input and simpler models can still perform well, with bilinear interpolation achieving a SSIM value of 0.981 at a  $2\times$  factor, and the U-Net and W-Net values of 0.998 and 0.999 respectively. Differences are also observed in other metrics, e.g. MSE values of  $3.57 \times 10^{-5}$ ,  $4.75 \times 10^{-6}$  and  $1.60 \times 10^{-6}$  for bilinear, U-Net and W-Net models respectively (Table 4). At higher downscaling factors ( $\geq 8\times$ , Figs. 8–9), the U-Net model tends to produce overly smooth predictions. However, the W-Net predictions feature a greater level of high-frequency detail, which is attributed to the use of the channel-wise attention mechanism allowing high-resolution topography and land-sea mask data to be incorporated within the model.

Examining the kinetic energy spectrum allows us to identify differences in the distribution of energy across a range of wavenumbers (Fig. 10). A  $k^{-5/3}$  dependence is anticipated in the mesoscale; the effective resolution is defined as the point at which the model spectrum decays relative to this expected spectrum [64]. The spectral tail past this point is attributed to aliasing, thus interpretation should be limited to before this point, which occurs at a wavenumber of  $\sim 10^{-4} \text{ m}^{-1}$  in Fig. 10. All approaches, including bilinear interpolation, closely match the target spectrum at a downscaling factor of  $2\times$ . Differences between spectra increase at higher factors and, in all cases, the W-Net spectrum is closer to the target, although at factors  $> 4\times$  it also struggles to predict the kinetic energy at higher wavenumbers.

### 3.1.2. Test set D03\_T2

The second test set is derived from WRF domain D03\_T2, which covers a different location to domain D03 with only a small overlap. While data from domain D03 was used for model training, data from domain D03\_T2 is from a previously unseen region. In general, the U-Net and W-Net models perform worse on test set D03\_T2 than test set D03. This is expected since models were trained on data generated over the same geographic area as test set D03. Inference on D03\_T2 data is a more challenging problem, as the topography and land-sea area differ from the training set, as might the wind regime.

At a lower resolution ( $\leq 4\times$ , Figs. 11, 12), both the U-Net and W-Net models still perform well, better than bilinear interpolation. At higher downscaling factors ( $\geq 8\times$ , Figs. 13, 14), the U-Net model

**Table 5**

Summary of the results for each downscaling factor for test set D03\_T2. Each model was trained using a L1 loss function. For each factor, the best result is highlighted in bold.

| Model    | Factor | MSE                                     | MAE                                     | PSNR        | SSIM         |
|----------|--------|-----------------------------------------|-----------------------------------------|-------------|--------------|
| U-Net    | 2      | $1.09 \times 10^{-5}$                   | $1.61 \times 10^{-3}$                   | 50.4        | 0.994        |
| W-Net    | 2      | <b><math>4.88 \times 10^{-6}</math></b> | <b><math>1.24 \times 10^{-3}</math></b> | <b>53.7</b> | <b>0.998</b> |
| Bilinear | 2      | $3.02 \times 10^{-5}$                   | $3.09 \times 10^{-3}$                   | 45.6        | 0.98         |
| U-Net    | 4      | $7.47 \times 10^{-5}$                   | $4.95 \times 10^{-3}$                   | 41.9        | 0.954        |
| W-Net    | 4      | <b><math>2.94 \times 10^{-5}</math></b> | <b><math>3.44 \times 10^{-3}</math></b> | <b>45.6</b> | <b>0.982</b> |
| Bilinear | 4      | $8.43 \times 10^{-5}$                   | $5.61 \times 10^{-3}$                   | 41.2        | 0.945        |
| U-Net    | 8      | $1.44 \times 10^{-4}$                   | $7.65 \times 10^{-3}$                   | 39.0        | 0.923        |
| W-Net    | 8      | <b><math>7.76 \times 10^{-5}</math></b> | <b><math>5.92 \times 10^{-3}</math></b> | <b>41.4</b> | <b>0.953</b> |
| Bilinear | 8      | $1.34 \times 10^{-4}$                   | $7.63 \times 10^{-3}$                   | 39.1        | 0.926        |
| U-Net    | 16     | $1.95 \times 10^{-4}$                   | $9.44 \times 10^{-3}$                   | 37.6        | 0.917        |
| W-Net    | 16     | <b><math>1.32 \times 10^{-4}</math></b> | <b><math>8.06 \times 10^{-3}</math></b> | <b>39.1</b> | <b>0.942</b> |
| Bilinear | 16     | $1.92 \times 10^{-4}$                   | $9.57 \times 10^{-3}$                   | 37.6        | 0.919        |

generalises poorly, with metrics comparable to bilinear interpolation (Table 5). Some artefacts are visible within Figs. 12–14 for the U-Net model, and topography-related features from the original D03 domain are also visible (e.g. bottom-right corner). In contrast, the corresponding W-Net model appears to generalise better, displaying no visible artefacts and producing coherent high-frequency structures related to the local topography. This improved generalisation capability and model stability is attributed to the CAM architecture within the W-Net (Section 2.2.3) allowing high-resolution prior data fields (topography and land-sea mask) from the previously unseen region to be directly incorporated during inference. Overall, the W-Net maintains a significant improvement over other methods at all downscaling factors, indicating improved generalisation ability. However, at  $16\times$ , the W-Net fails to reconstruct much of the finer flow characteristics (Fig. 14), but is still able to make corrections in areas of complex topography. When analysing the kinetic energy spectra (Fig. 15) similar trends to test set D03 are noted. In general, the W-Net spectra continue to match the target spectra better than other approaches, although as the resolution factor is increased, the spectra of all models diverge from the target.

### 3.2. Loss function variants

The performance of a number of loss functions is evaluated for downscaling by a factor of  $8\times$  using the W-Net model. The results for test sets D03 and D03\_T2 are summarised in Tables 6 and 7, with the best metrics highlighted in bold. Example predictions for each loss function variant are shown in Figs. 16 and 17, and KE spectra are shown in Figs. 18 and 19.

In this work, it is found that the L1, L2, SSIM and FFL functions all perform comparably. This is reflected both in the metrics and in the energy spectra. Although the SSIM is often cited as a perceptual measure (e.g. [61]) capable of emulating the human visual system, several works have shown this not to be the case and that it is closely related to the MSE and PSNR [65]. To our knowledge, this is the first study, after its introduction by [59], that applies and compares the Wiener loss to other functions. The performance of the Wiener loss differs significantly from the L1, L2, SSIM and FFL loss functions. It tends to generate higher-frequency detail, which is reflected in the improved KE spectrum (Figs. 18–19). However, amplitudes tend to be over-estimated, resulting in significantly poorer metrics in comparison to other loss functions (Tables 6–7). To combat this, a composite Wiener-L1 function ( $\alpha = 0.3$ ) is used that yields both improved metrics versus the standard Wiener loss, as well as improved higher-frequency detail reconstruction and a closer KE spectrum in comparison to other loss functions.

The optimal choice of loss function may vary with the use-case. If only the large-scale structure of the wind field is required then L1 or L2 may suffice. However, if high-frequency detail is a priority, as may be the case for wind energy applications, then a Wiener-based function

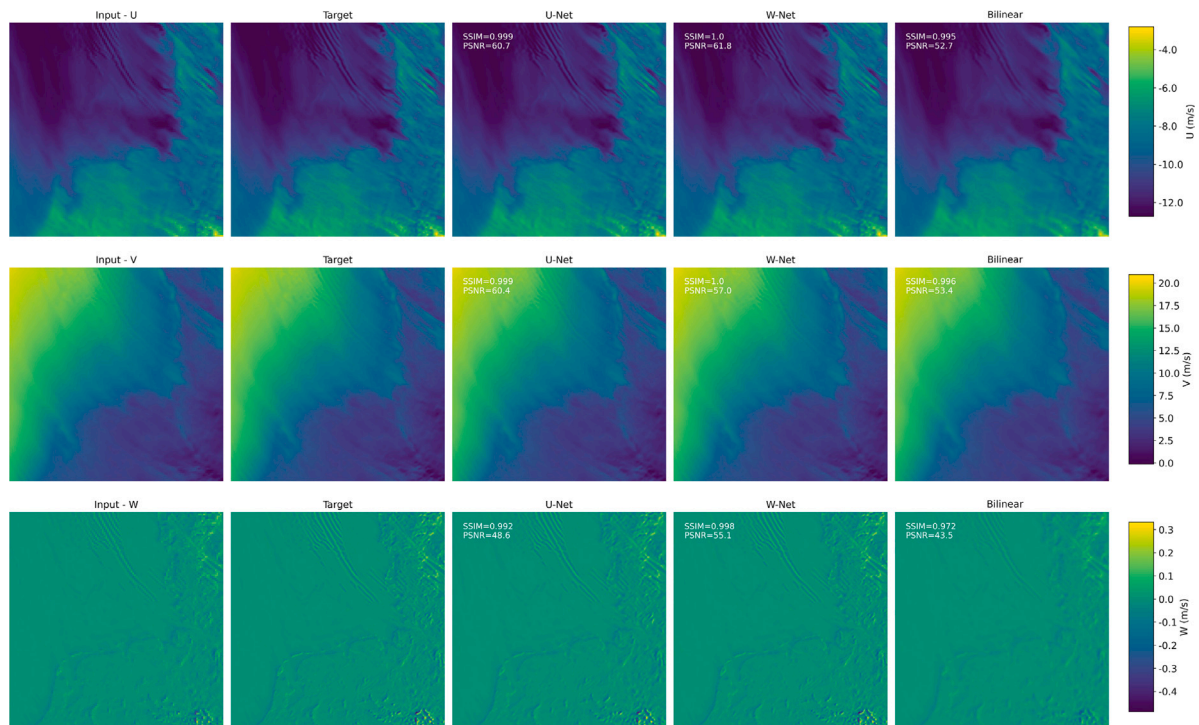


**Fig. 6.** Example predictions for test set D03, for a downscaling factor of 2. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

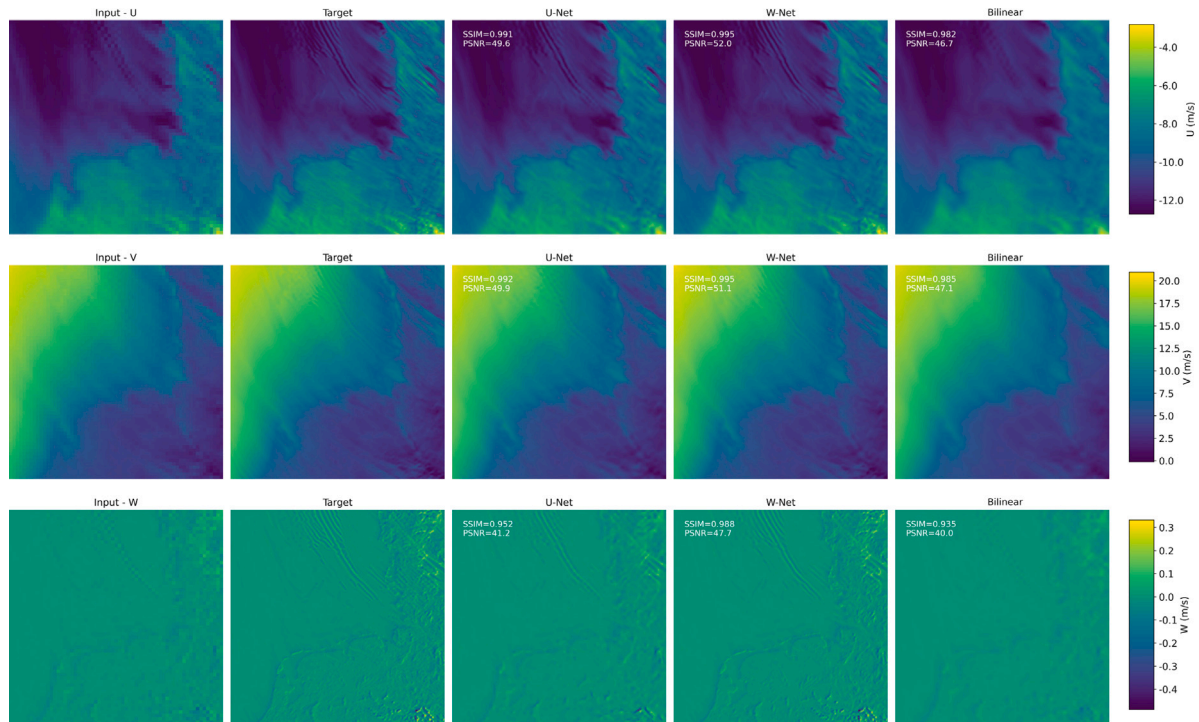


**Fig. 7.** Example predictions for test set D03, for a downscaling factor of 4. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

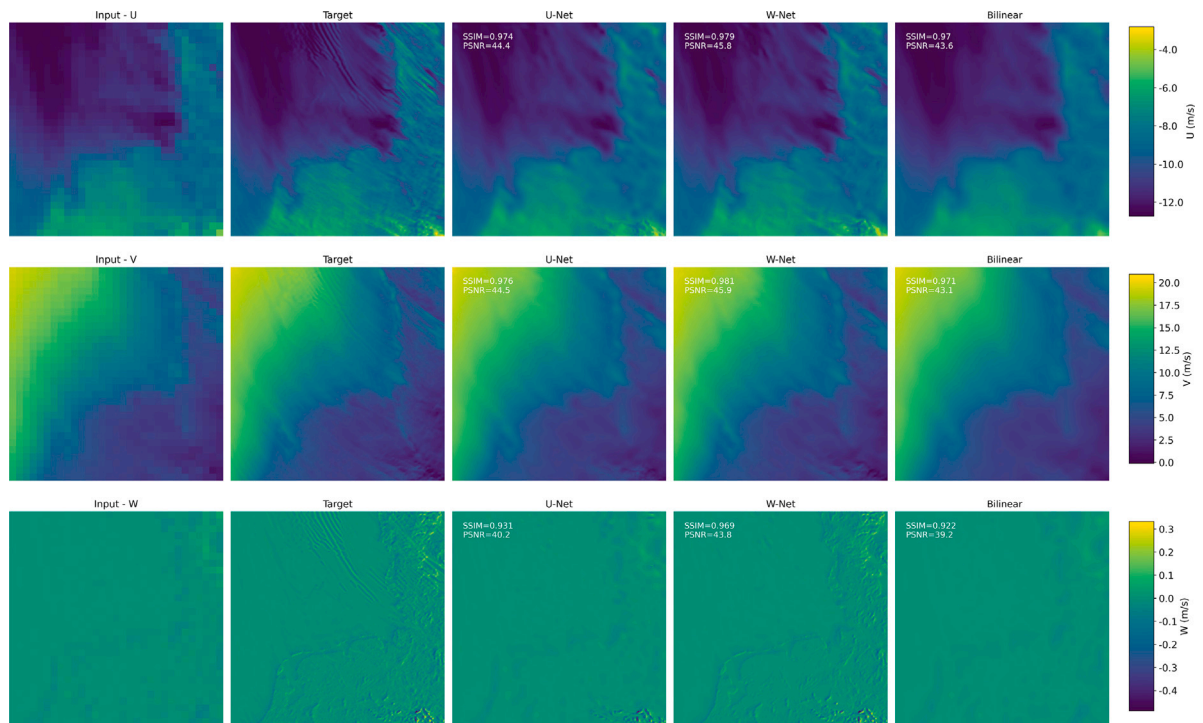


**Fig. 8.** Example predictions for test set D03, for a downscaling factor of 8. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

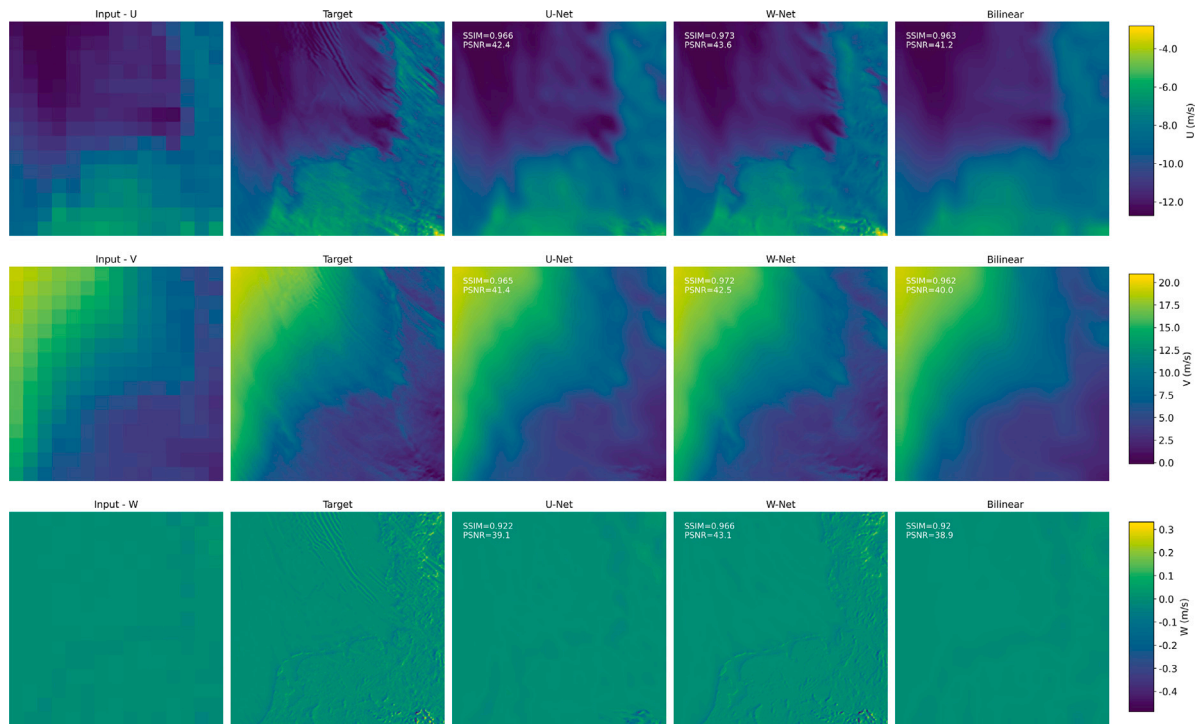


**Fig. 9.** Example predictions for test set D03, for a downscaling factor of 16. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

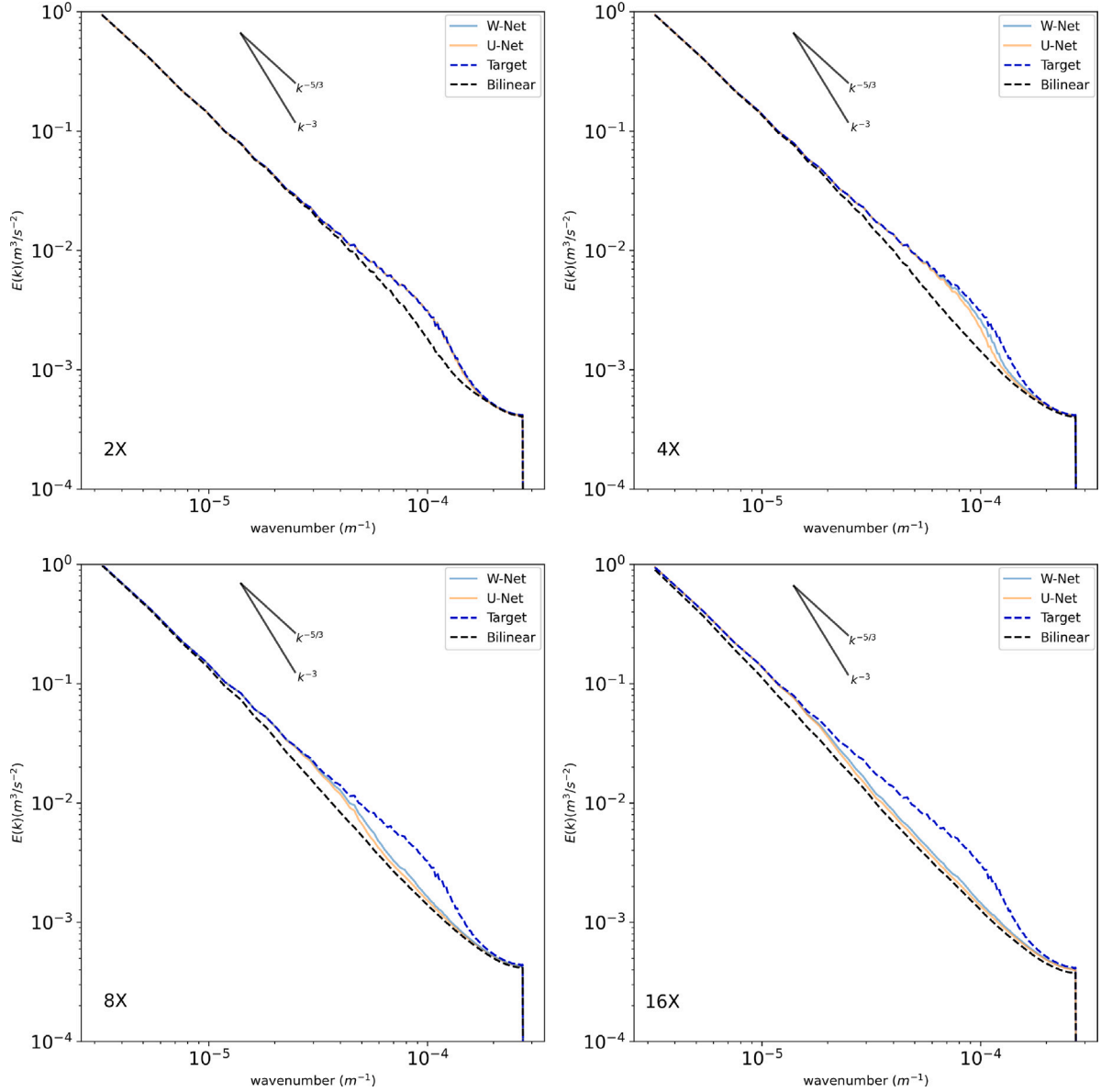


Fig. 10. Kinetic energy spectrum comparison over all downscaling factors (2x top-left to 16x bottom-right) for test set D03.

Table 6

Summary of the results on test set D03 for a W-Net model trained using each loss function and downscaling by a factor of 8x. The best result is highlighted in bold.

| Model     | MSE                                     | MAE                                     | PSNR        | SSIM         |
|-----------|-----------------------------------------|-----------------------------------------|-------------|--------------|
| L1        | <b><math>9.38 \times 10^{-5}</math></b> | <b><math>5.96 \times 10^{-3}</math></b> | <b>40.9</b> | <b>0.944</b> |
| L2        | $9.44 \times 10^{-5}$                   | $6.01 \times 10^{-3}$                   | <b>40.9</b> | <b>0.944</b> |
| SSIM      | $1.11 \times 10^{-4}$                   | $6.91 \times 10^{-3}$                   | 40.1        | <b>0.944</b> |
| FFL       | $9.60 \times 10^{-5}$                   | $6.07 \times 10^{-3}$                   | 40.8        | 0.943        |
| Wiener    | $6.28 \times 10^{-4}$                   | $1.85 \times 10^{-2}$                   | 33.1        | 0.918        |
| Wiener-L1 | $1.26 \times 10^{-4}$                   | $6.60 \times 10^{-3}$                   | 39.7        | 0.937        |
| Bilinear  | $1.66 \times 10^{-4}$                   | $7.75 \times 10^{-3}$                   | 38.3        | 0.921        |

Table 7

Summary of the results on test set D03.T2 for a W-Net model trained using each loss function and downscaling by a factor of 8x. The best result is highlighted in bold.

| Model     | MSE                                     | MAE                                     | PSNR        | SSIM         |
|-----------|-----------------------------------------|-----------------------------------------|-------------|--------------|
| L1        | $7.76 \times 10^{-5}$                   | <b><math>5.92 \times 10^{-3}</math></b> | 41.4        | 0.953        |
| L2        | <b><math>7.69 \times 10^{-5}</math></b> | $5.93 \times 10^{-3}$                   | <b>41.5</b> | <b>0.954</b> |
| SSIM      | $1.26 \times 10^{-4}$                   | $8.03 \times 10^{-3}$                   | 39.6        | 0.952        |
| FFL       | $7.75 \times 10^{-5}$                   | $6.01 \times 10^{-3}$                   | 41.4        | 0.952        |
| Wiener    | $5.34 \times 10^{-4}$                   | $1.67 \times 10^{-2}$                   | 34.4        | 0.913        |
| Wiener-L1 | $1.14 \times 10^{-4}$                   | $6.97 \times 10^{-3}$                   | 39.7        | 0.941        |
| Bilinear  | $1.34 \times 10^{-4}$                   | $7.63 \times 10^{-3}$                   | 39.1        | 0.926        |

should be considered. It should also be noted that there are other possible permutations of loss functions, and although any tuneable parameters specific to the function were investigated, such as the kernel size in the SSIM function or the filter size in the Wiener function, this was not fully exhaustive and it is possible that other combinations will yield different results. In particular, the Wiener loss has a number of

tuneable parameters that may result in further improvements, such as optimising the design of its penalty function. It is also possible that results will vary for different downscaling factors and for different models. Further investigation of these aspects is beyond the scope of this study but will be considered in future work.

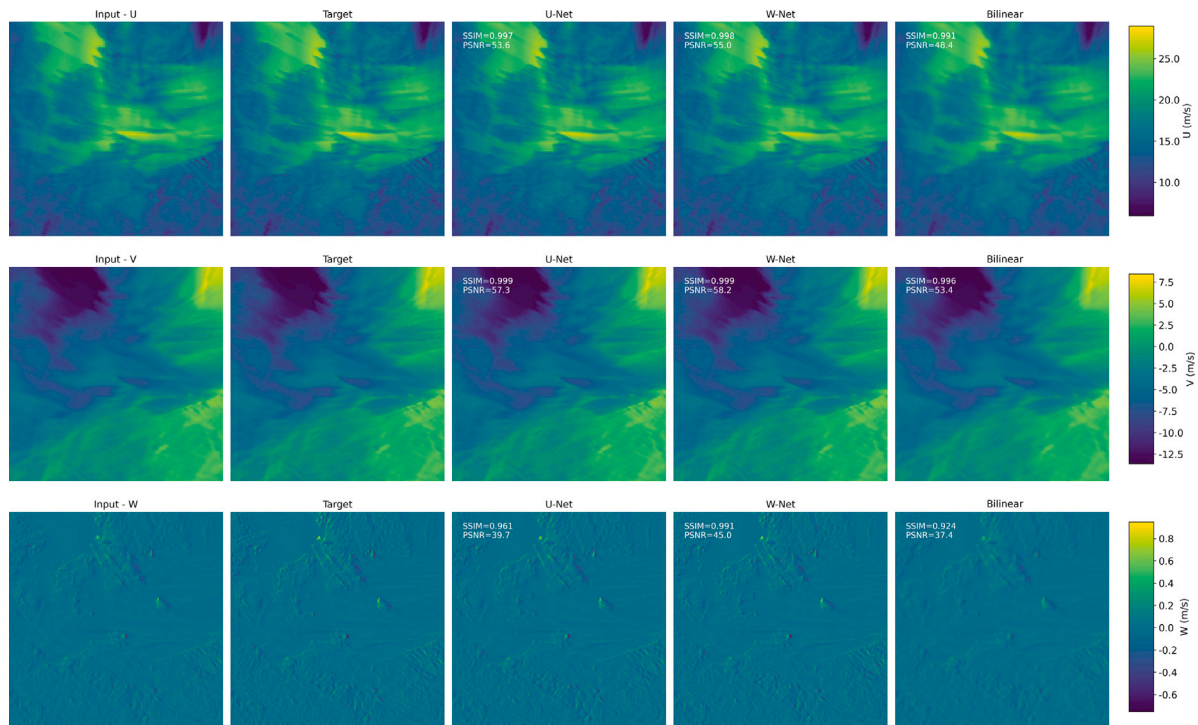


Fig. 11. Example predictions for test set D03\_T2, for a downscaling factor of 2. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

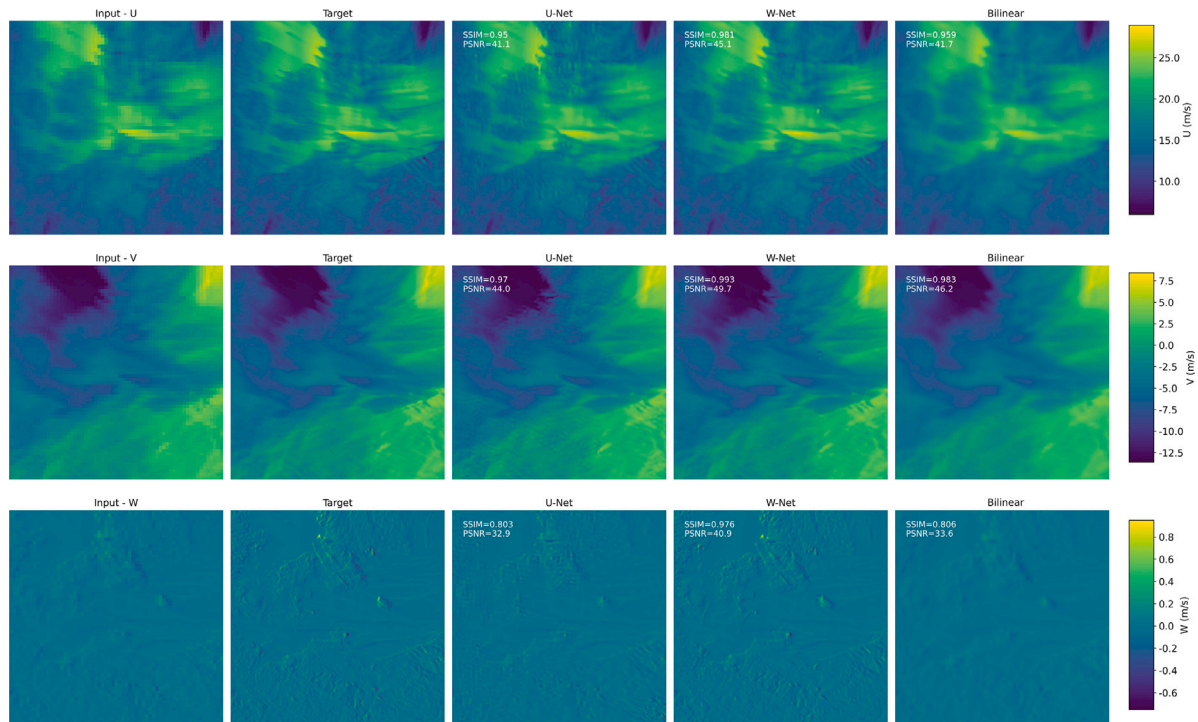


Fig. 12. Example predictions for test set D03\_T2, for a downscaling factor of 4. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

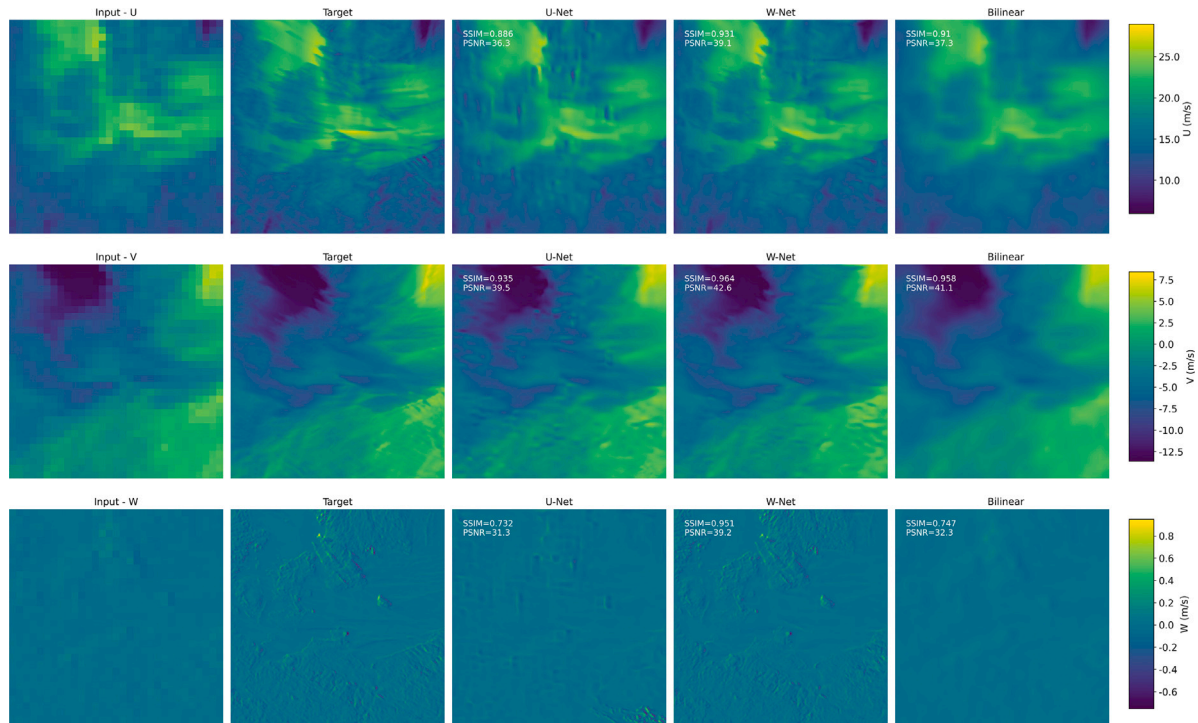


Fig. 13. Example predictions for test set D03\_T2, for a downscaling factor of 8. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

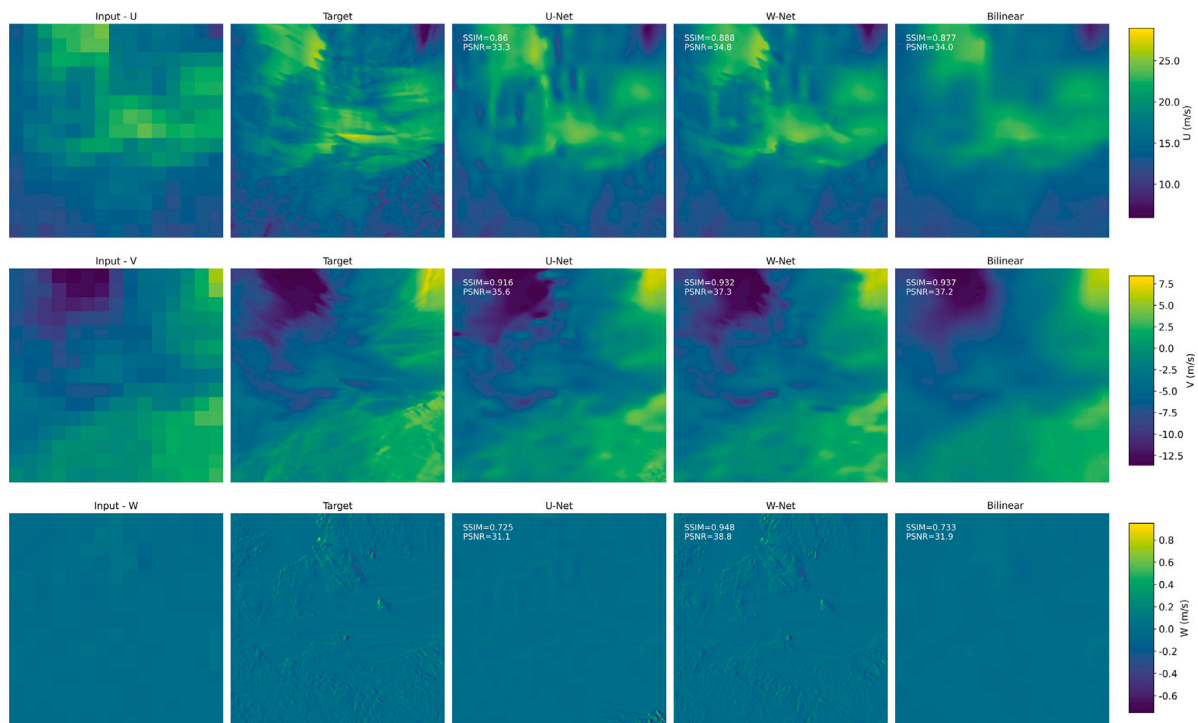


Fig. 14. Example predictions for test set D03\_T2, for a downscaling factor of 16. Top to bottom show the  $u$ ,  $v$ , and  $w$  components of the wind field. Left to right show input, target, U-Net, W-Net, and bilinear interpolation.

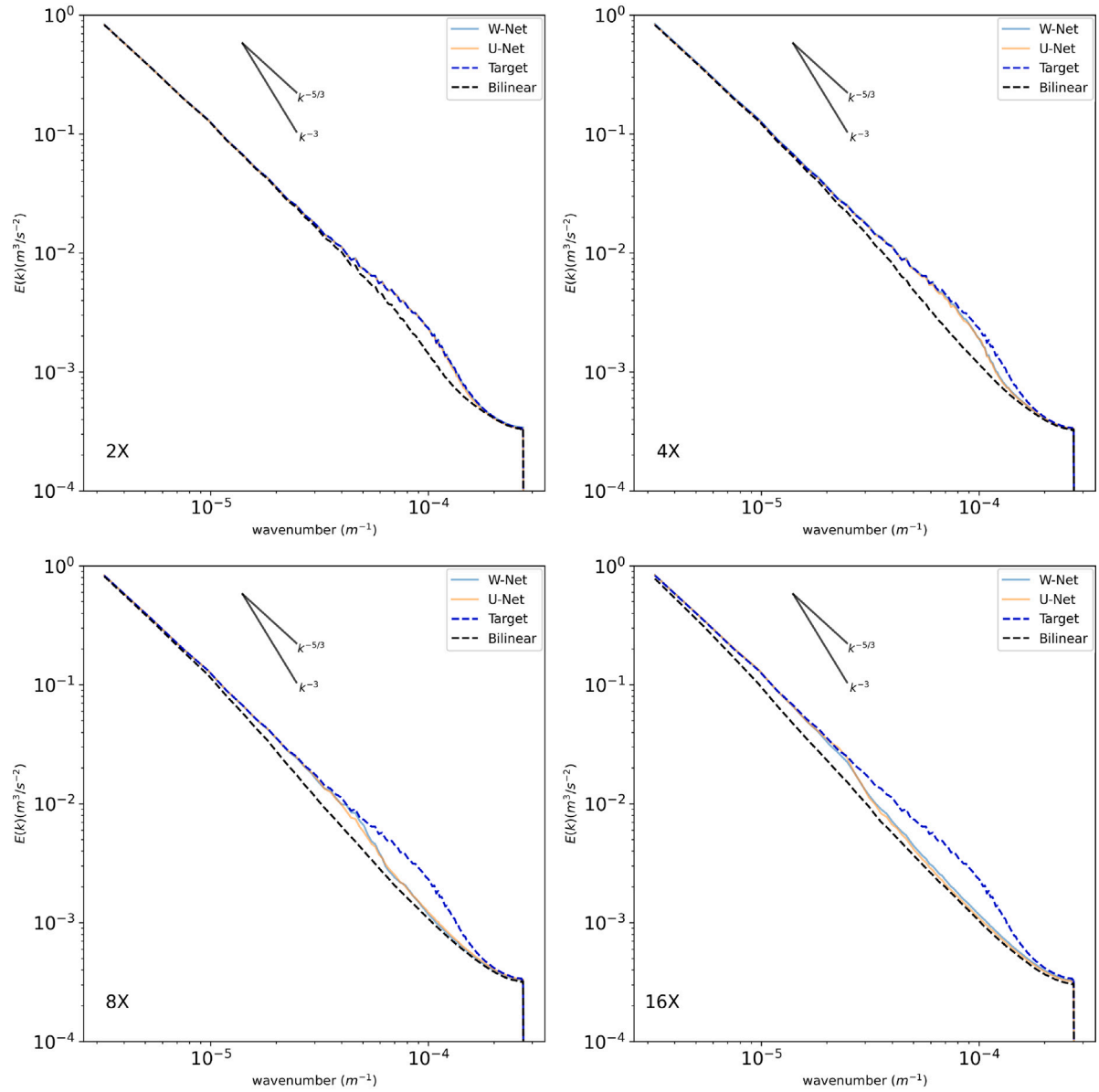


Fig. 15. Kinetic energy spectrum comparison over all downscaling factors (2× top-left to 16× bottom-right) for test set D03\_T2.

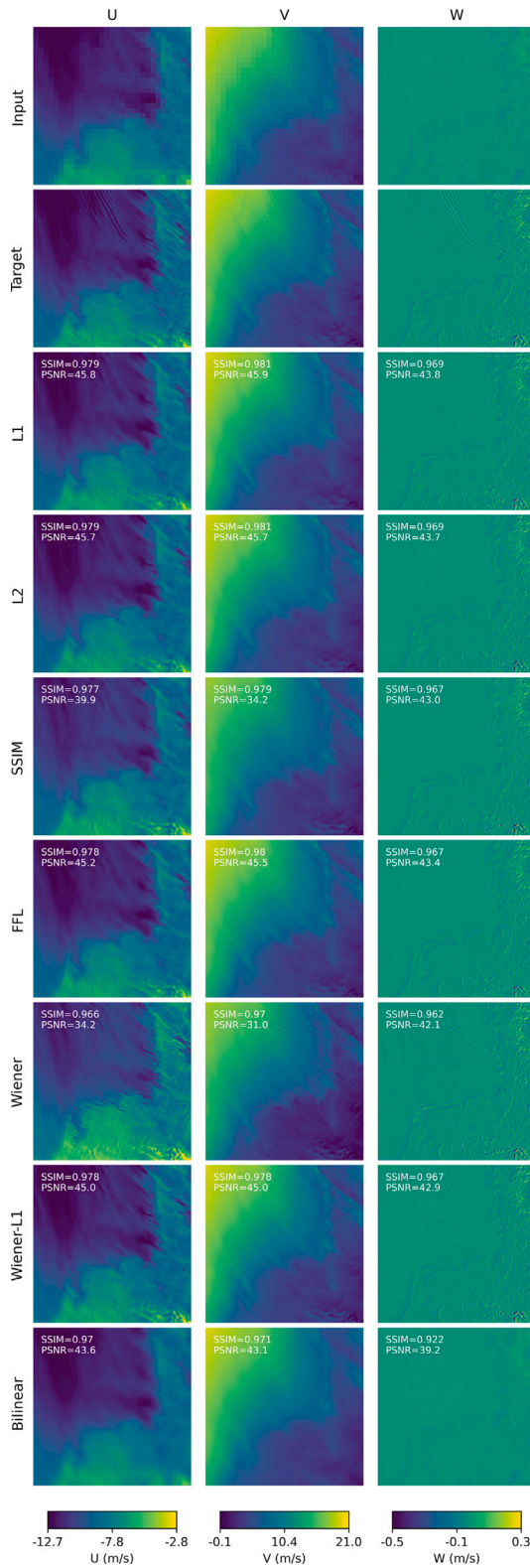


Fig. 16. Example W-Net predictions for each loss function for test set D03, for a downscaling factor of 8x. From left to right -  $u$ ,  $v$ , and  $w$  components of the wind field; from top to bottom - input, target, L1, L2, SSIM, FFL, Wiener, Wiener-L1, bilinear.

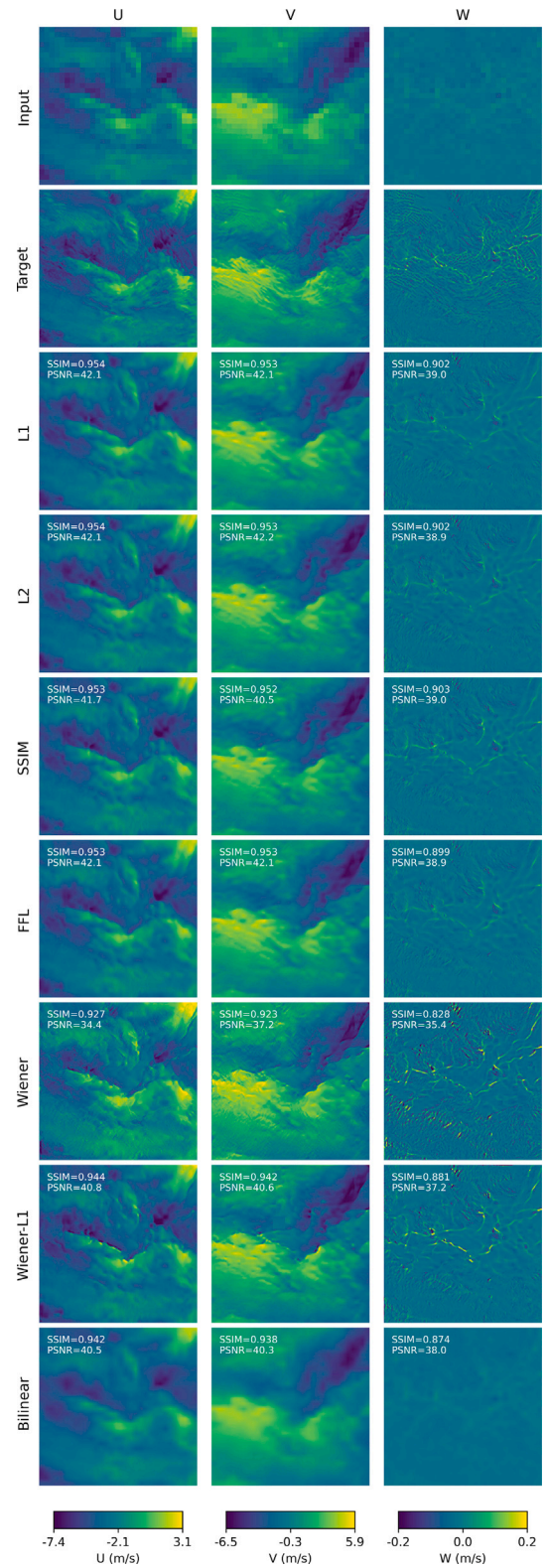


Fig. 17. Example W-Net predictions for each loss function for test set D03\_T2, for a downscaling factor of 8x. From left to right -  $u$ ,  $v$ , and  $w$  components of the wind field; from top to bottom - input, target, L1, L2, SSIM, FFL, Wiener, Wiener-L1, bilinear.

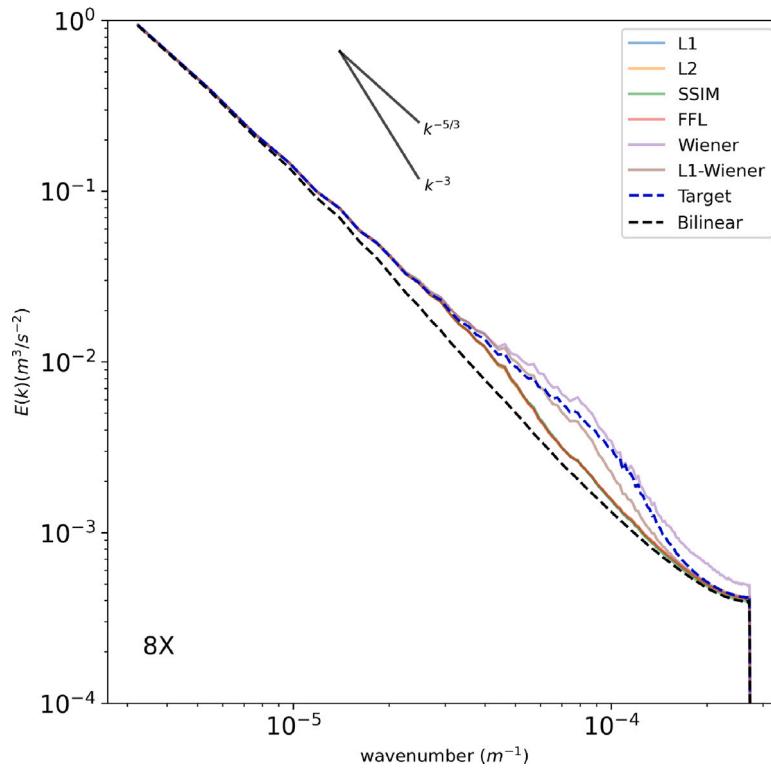


Fig. 18. Comparison of kinetic energy spectra for each loss function, for the W-Net at an 8x downscaling factor for test set D03.

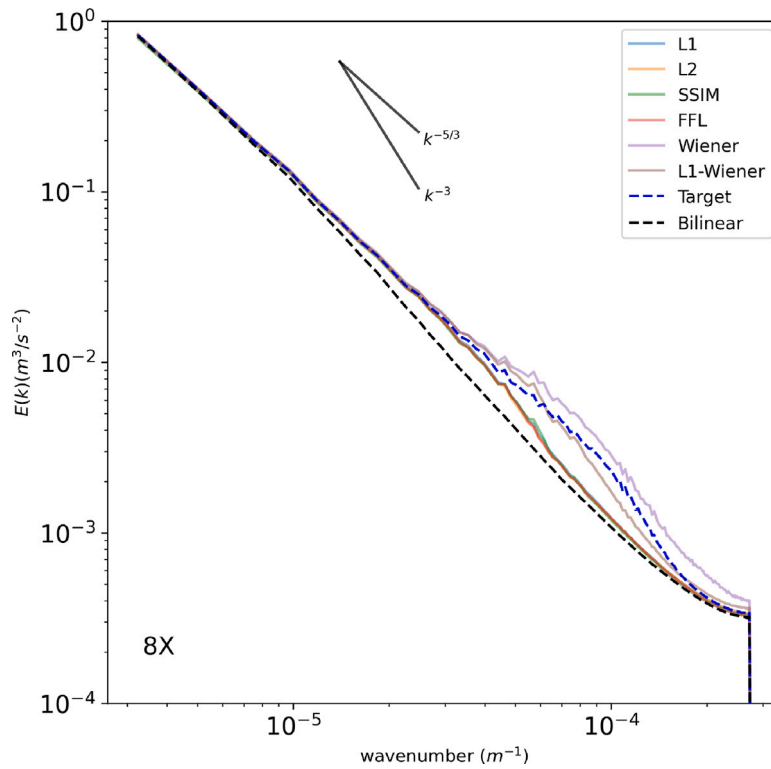


Fig. 19. Comparison of kinetic energy spectra for each loss function, for the W-Net at an 8x downscaling factor for test set D03\_T2.

#### 4. Conclusion

This work demonstrates how U-Nets can be applied to spatially downscale WRF simulations in the North Sea at several different resolutions, up to a factor of 16. A W-Net architecture is presented that incorporates prior fields through the use of two sequential U-Nets and a fusion layer which utilises a channel-wise attention mechanism. The advantages of this approach over a standard U-Net architecture are demonstrated through improvements in all performance metrics and enhanced generalisation capability across different geographical domains, a benefit that becomes increasingly apparent at higher downscaling factors.

No significant difference in performance is identified between W-Net models trained using standard pixel-wise loss functions (L1 and L2), structural similarity or frequency-focused functions. However, a Wiener loss function promotes the generation of higher-frequency detail, which is reflected in the KE spectra, but also introduces a tendency in the model to over-estimate amplitudes. A Wiener-L1 composite loss provides a balance between generating high-frequency detail and correctly predicting amplitudes.

Further work could include applying the model to different geographical regions, exploring the addition of physics-based constraints within the loss function, extending downscaling to the 3D spatio-temporal domain, and comparison against other cutting-edge architectures such as vision transformers and neural operators. This work also raises the question of what is the “best” metric to assess performance in downscaling wind fields, which may vary depending on the application. Standard metrics such as the MAE and MSE may not be sufficient on their own. KE spectra should also be considered in this domain.

#### CRedit authorship contribution statement

**Alexander M. Campbell:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Simon C. Warder:** Writing – review & editing. **B. Bhaskaran:** Writing – review & editing, Supervision. **Matthew D. Piggott:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

#### Declaration of competing interest

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

#### Acknowledgements

The authors would like to acknowledge funding from Shell Information Technology International Ltd, and the UK Engineering and Physical Sciences Research Council under the Industrial CASE programme (Project EP/V519534/1).

#### Data availability

Data will be made available on request.

#### References

- [1] Piggott MD, Kramer SC, Funke SW, Culley DM, Angeloudis A. 8.10 - optimization of marine renewable energy systems. In: Letcher TM, editor. *Comprehensive renewable energy*. 2nd ed.. Oxford: Elsevier; 2022, p. 176–220. <http://dx.doi.org/10.1016/B978-0-12-819727-1.00179-5>, URL: <https://www.sciencedirect.com/science/article/pii/B9780128197271001795>.
- [2] Alkhatay G, Mehmood R. A review and taxonomy of wind and solar energy forecasting methods based on deep learning. *Energy AI* 2021;4:100060. <http://dx.doi.org/10.1016/j.egyai.2021.100060>, URL: <https://www.sciencedirect.com/science/article/pii/S2666546821000148>.
- [3] Li S, Robert A, Faisal AA, Piggott MD. Learning to optimise wind farms with graph transformers. *Appl Energy* 2024;359:122758. <http://dx.doi.org/10.1016/j.apenergy.2024.122758>, URL: <https://www.sciencedirect.com/science/article/pii/S0306261924001417>.
- [4] Pu Z, Kalnay E. *Numerical weather prediction basics: Models, numerical methods, and data assimilation*. Springer; 2018.
- [5] Srivastava AK, Ullrich PA, Rastogi D, Vahmani P, Jones A, Grotjahn R. Assessment of wrf (v 4.2.1) dynamically downscaled precipitation on subdaily and daily timescales over conus. *Geosci Model Dev* 2023;16:3699–722. <http://dx.doi.org/10.5194/gmd-16-3699-2023>, URL: <https://gmd.copernicus.org/articles/16/3699/2023/>.
- [6] Kashinath K, Mustafa M, Albert A, Wu J-L, Jiang C, Esmaeilzadeh S, Azizzadehsheli K, Wang R, Chattopadhyay A, Singh A, Manepalli A, Chirila D, Yu R, Walters R, White B, Xiao H, Tchelepi HA, Marcus P, Anandkumar A, Hassanzadeh P, Prabhat N. Physics-informed machine learning: case studies for weather and climate modelling. *Philos Trans R Soc A* 2021;379:20200093. <http://dx.doi.org/10.1098/rsta.2020.0093>, URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0093>, arXiv:<https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2020.0093>.
- [7] Serifi A, Günther T, Ban N. Spatio-temporal downscaling of climate data using convolutional and error-predicting neural networks. *Front Clim* 2021;3. <http://dx.doi.org/10.3389/fclim.2021.656479>, URL: <https://www.frontiersin.org/articles/10.3389/fclim.2021.656479>.
- [8] Gopalakrishnan S, Choudhury A. A deep review of video super-resolution. *Signal Process, Image Commun* 2024;117:175. <http://dx.doi.org/10.1016/j.image.2024.117175>, URL: <https://www.sciencedirect.com/science/article/pii/S0923596524000766>.
- [9] Chantry M, Christensen H, Dueben P, Palmer T. Opportunities and challenges for machine learning in weather and climate modelling: hard, medium and soft ai. *Philos Trans R Soc Lond Ser A* 2021;379:20200083. <http://dx.doi.org/10.1098/rsta.2020.0083>, URL: <https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2020.0083>, arXiv:<https://royalsocietypublishing.org/doi/pdf/10.1098/rsta.2020.0083>.
- [10] Keller JD, Pothast R. Ai-based data assimilation: Learning the functional of analysis estimation. 2024, URL: <https://arxiv.org/abs/2406.00390>, arXiv:2406.00390.
- [11] Bi K, Xie L, Zhang H, Chen X, Gu X, Tian Q. Accurate medium-range global weather forecasting with 3d neural networks. *Nat* 2023;619:533–8, URL: <https://www.nature.com/articles/s41586-023-06185-3>.
- [12] Zhao E, Sun S, Wang S. New developments in wind energy forecasting with artificial intelligence and big data: a scientometric insight. *Data Sci Manag* 2022;5:84–95. <http://dx.doi.org/10.1016/j.dsm.2022.05.002>, URL: <https://www.sciencedirect.com/science/article/pii/S2666764922000212>.
- [13] Liang J, Zhang K, Al-Durra A, Mueyen S, Zhou D. A state-of-the-art review on wind power converter fault diagnosis. *Energy Rep* 2022;8:5341–69. <http://dx.doi.org/10.1016/j.egyri.2022.03.178>, URL: <https://www.sciencedirect.com/science/article/pii/S2352484722007338>.
- [14] Abirami Sasanthiran SG, Ragala R. A review of artificial intelligence applications in wind turbine health monitoring. *Int J Sustain Energy* 2024;43:2326296. <http://dx.doi.org/10.1080/14786451.2024.2326296>.
- [15] Clare MCA, Warder SC, Neal R, Bhaskaran B, Piggott MD. An unsupervised learning approach for predicting wind farm power and downstream wakes using weather patterns. *J Adv Model Earth Syst* 2024;16:e2023MS003947. <http://dx.doi.org/10.1029/2023MS003947>, URL: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2023MS003947>, arXiv:<https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2023MS003947>, e2023MS003947 2023MS003947.
- [16] Vandal T, Kodra E, Ganguly S, Michaelis A, Nemani R, Ganguly A. DeepSD: Generating high resolution climate change projections through single image super-resolution. In: *Proceedings of the 23rd acm sigkdd international conference on knowledge discovery and data mining*. 2017, p. 1663–72, URL: <https://dl.acm.org/doi/10.1145/3097983.3098004>.
- [17] Rodrigues R, Eduardo, Oliveira, Igor, Cunha, Renato, Netto, Marco. Deepdownscale: A deep learning strategy for high-resolution weather forecast. In: *2018 IEEE 14th international conference on e-science (e-science)*. 2018, p. 415–22. <http://dx.doi.org/10.1109/eScience.2018.00130>, URL: <https://ieeexplore.ieee.org/document/8588749>.
- [18] Vandal T, Kodra E, Ganguly A. Intercomparison of machine learning methods for statistical downscaling: The case of daily and extreme precipitation. *Theor Appl Climatol* 2019;137:557–70, URL: <https://link.springer.com/article/10.1007/s00704-018-2613-3>.
- [19] Pan B, Hsu K, Aghakouchak A, Sorooshian S. Improving precipitation estimation using convolutional neural network. *Water Resour Res* 2019;55:2301–21. <http://dx.doi.org/10.1029/2018wr024090>, URL: <https://agupubs.onlinelibrary.wiley.com/doi/full/10.1029/2018WR024090>.

- [20] Cheng J, Kuang Q, Shen C, Liu J, Tan X, Liu W. Reslap: Generating high-resolution climate prediction through image super-resolution. *IEEE Access* 2020;8:39623–34, URL: <https://api.semanticscholar.org/CorpusID:212646477>.
- [21] Pouliot D, Latifovic R, Pasher J, Duffe J. Landsat super-resolution enhancement using convolution neural networks and sentinel-2 for training. *Remote Sens* 2018;10. <http://dx.doi.org/10.3390/rs10030394>, URL: <https://www.mdpi.com/2072-4292/10/3/394>.
- [22] Dong C, Loy CC, He K, Tang X. Learning a deep convolutional network for image super-resolution. In: *Computer vision—ECCV 2014: 13th European conference, Zurich, Switzerland, September 6–12, 2014, proceedings, part IV* 13. Springer; 2014, p. 184–99, URL: [https://link.springer.com/chapter/10.1007/978-3-319-10593-2\\_13](https://link.springer.com/chapter/10.1007/978-3-319-10593-2_13).
- [23] Singh A, Albert A, White B. Numerical weather model super-resolution. In: *33rd conference on neural information processing systems*. 2019, p. 00, URL: [https://ml4physicsciences.github.io/2019/files/NeurIPS\\_ML4PS\\_2019\\_75.pdf](https://ml4physicsciences.github.io/2019/files/NeurIPS_ML4PS_2019_75.pdf).
- [24] Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Qiao Y, Change Loy C. Esrgan: Enhanced super-resolution generative adversarial networks. In: *Proceedings of the European conference on computer vision (ECCV) workshops*. 2018, URL: [https://link.springer.com/chapter/10.1007/978-3-030-11021-5\\_5](https://link.springer.com/chapter/10.1007/978-3-030-11021-5_5), pp. 0–0.
- [25] Huang X. Deep-learning based climate downscaling using the super-resolution method: a case study over the western us. *Geosci Model Dev Discuss* 2020. <http://dx.doi.org/10.5194/gmd-2020-214>, URL: <https://gmd.copernicus.org/preprints/gmd-2020-214/>.
- [26] Stengel K, Glaws A, Hettinger D, King RN. Adversarial super-resolution of climatological wind and solar data. *Proc Natl Acad Sci - PNAS* 2020;117:16805–15. <http://dx.doi.org/10.1073/pnas.1918964117>, URL: <https://www.ncbi.nlm.nih.gov/pubmed/32631993>.
- [27] Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, et al. Photo-realistic single image super-resolution using a generative adversarial network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 4681–90, URL: [https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Ledig\\_Photo-Realistic\\_Single\\_Image\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.pdf).
- [28] Höhle K, Kern M, Hewson T, Westermann R. A comparative study of convolutional neural network models for wind field downscaling. *Meteorol Appl* 2020;27:e1961. <http://dx.doi.org/10.1002/met.1961>, URL: <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/met.1961>.
- [29] Schweri L, Foucher S, Tang J, Azevedo VC, Günther T, Solenthaler B. A physics-aware neural network approach for flow data reconstruction from satellite observations. *Front Clim* 2021;3:656505, URL: <https://www.frontiersin.org/journals/climate/articles/10.3389/fclim.2021.656505/full>.
- [30] Rojas-Campos A, Langguth M, Wittenbrink M, Pipa G. Deep learning models for generation of precipitation maps based on numerical weather prediction. *Geosci Model Dev* 2023;16:1467–80. <http://dx.doi.org/10.5194/gmd-16-1467-2023>, URL: <https://gmd.copernicus.org/articles/16/1467/2023/>.
- [31] Jo Y, Kang J, Oh SW, Nam S, Vajda P, Kim SJ. Learning the loss functions in a discriminative space for video restoration. 2020, arXiv preprint arXiv:2003.09124. URL: <https://arxiv.org/abs/2003.09124>.
- [32] Chen H. Challenges and corresponding solutions of generative adversarial networks (gans): a survey study. In: *Journal of physics: Conference series*. vol. 1827, IOP Publishing; 2021, 012066, URL: <https://iopscience.iop.org/article/10.1088/1742-6596/1827/1/012066/pdf>.
- [33] Mardani M, Brenowitz N, Cohen Y, Pathak J, Chen C-Y, Liu C-C, Vahdat A, Nabian MA, Ge T, Subramaniam A, Kashinath K, Kautz J, Pritchard M. Residual corrective diffusion modeling for km-scale atmospheric downscaling. 2024, URL: <https://arxiv.org/abs/2309.15214>. arXiv:2309.15214.
- [34] Raissi M, Perdikaris P, Karniadakis G. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 2019;378:686–707. <http://dx.doi.org/10.1016/j.jcp.2018.10.045>, URL: <https://www.sciencedirect.com/science/article/pii/S0021999118307125>.
- [35] Toscano JD, Oommen V, Varghese AJ, Zou Z, Daryakenari NA, Wu C, Karniadakis GE. From pinns to pikans: Recent advances in physics-informed machine learning. 2024, URL: <https://arxiv.org/abs/2410.13228>. arXiv:2410.13228.
- [36] Wang S, Teng Y, Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM J Sci Comput* 2021;43:A3055–81. <http://dx.doi.org/10.1137/20M1318043>, arXiv:https://doi.org/10.1137/20M1318043.
- [37] Sun L, Gao H, Pan S, Wang J-X. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput Methods Appl Mech Engrg* 2020;361:112732. <http://dx.doi.org/10.1016/j.cma.2019.112732>, URL: <https://www.sciencedirect.com/science/article/pii/S004578251930622X>.
- [38] Li Z, Zheng H, Kovachki N, Jin D, Chen H, Liu B, Azizzadenesheli K, Anandkumar A. Physics-informed neural operator for learning partial differential equations. *ACM/JMS J Data Sci* 2024;1:1–27. <http://dx.doi.org/10.1145/3648506>.
- [39] Wang S, Yu X, Perdikaris P. When and why pinns fail to train: A neural tangent kernel perspective. *J Comput Phys* 2022;449:110768. <http://dx.doi.org/10.1016/j.jcp.2021.110768>, URL: <https://www.sciencedirect.com/science/article/pii/S002199912100663X>.
- [40] Fuks O, Tchelepi HA. Limitations of physics informed machine learning for nonlinear two-phase transport in porous media. *J Mach Learn Model Comput* 2020;1:19–37. <http://dx.doi.org/10.1615/JMachLearnModelComput.2020033905>, URL: <https://www.dl.begellhouse.com/journals/558048804a15188a,583c4e56625ba94e,415f83b5707fde65.html>.
- [41] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Sci* 2020;367:1026–30. <http://dx.doi.org/10.1126/science.aaw4741>, URL: <https://www.science.org/doi/abs/10.1126/science.aaw4741>. arXiv:https://www.science.org/doi/pdf/10.1126/science.aaw4741.
- [42] Huang AJ, Agarwal S. On the limitations of physics-informed deep learning: Illustrations using first-order hyperbolic conservation law-based traffic flow models. *IEEE Open J Intell Transp Syst* 2023;4:279–93. <http://dx.doi.org/10.1109/OJITS.2023.3268026>, URL: <https://ieeexplore.ieee.org/document/10105558>.
- [43] Urbán JF, Stefanou P, Pons JA. Unveiling the optimization process of physics informed neural networks: How accurate and competitive can pinns be? *J Comput Phys* 2025;523:113656. <http://dx.doi.org/10.1016/j.jcp.2024.113656>, URL: <https://www.sciencedirect.com/science/article/pii/S0021999124009045>.
- [44] Skamarock WC, Klemp JB, Dudhia J, Gill DO, Liu Z. A description of the advanced research wrf model version 4. NCAR technical notes, 2019, <http://dx.doi.org/10.5065/1dfh-6p97>, URL: <http://n2t.net/ark:/85065/d72r3vrp>.
- [45] Powers JG, Klemp JB, Skamarock WC, Davis CA, Dudhia J, Gill DO, Coen JL, Gochis DJ, Ahmadov R, PeckCKham SE, Grell GA, Michalakos J, Trahan S, Benjamin SG, Alexander CR, Dimego GJ, Wang W, Schwartz CS, Romine GS, Liu Z, Snyder C, Chen F, Barlage MJ, Yu W, Duda MG. The weather research and forecasting model: Overview, system efforts, and future directions. *Bull Am Meteorol Soc* 2017;98. <http://dx.doi.org/10.1175/BAMS-D-15-00308.1>.
- [46] Witha B, Hahmann A, Sile T, Dörenkämper M, Ezber Y, García-Bustamante E, González Rouco JF, Leroy G, Navarro J. Report on wrf model sensitivity studies and specifications for the mesoscale wind atlas production runs (newa deliverable 4.3). 2019, <http://dx.doi.org/10.5281/zenodo.2682603>, URL: [https://backend.orbit.dtu.dk/ws/portalfiles/portal/180023688/2019\\_05\\_09\\_NEWA\\_D4\\_3\\_final.pdf](https://backend.orbit.dtu.dk/ws/portalfiles/portal/180023688/2019_05_09_NEWA_D4_3_final.pdf).
- [47] Hahmann AN, Sile T, Witha B, Davis NN, Dörenkämper M, Ezber Y, García-Bustamante E, González-Rouco JF, Navarro J, Olsen BT, Söderberg S. The making of the new european wind atlas – part 1: Model sensitivity. *Geosci Model Dev* 2020;13:5053–78. <http://dx.doi.org/10.5194/gmd-13-5053-2020>, URL: <https://gmd.copernicus.org/articles/13/5053/2020/>.
- [48] Coimbra I, Laginha Palma J. Synoptic and mesoscale winds in the complex terrain of perdão. *J Phys: Conf Ser* 2022;2265:022010. <http://dx.doi.org/10.1088/1742-6596/2265/2/022010>, URL: <https://iopscience.iop.org/article/10.1088/1742-6596/2265/2/022010>.
- [49] Yu E, Bai R, Chen X, Shao L. Impact of physical parameterizations on wind simulation with wrf v3.9.1.1 under stable conditions at planetary boundary layer gray-zone resolution: a case study over the coastal regions of north china. *Geosci Model Dev* 2022;15:8111–34. <http://dx.doi.org/10.5194/gmd-15-8111-2022>, URL: <https://gmd.copernicus.org/articles/15/8111/2022/>.
- [50] Cañadillas B, Wang S, Ahlert Y, Djath B, Berekzai M, Foreman R, Lampert A. Coastal horizontal wind speed gradients in the north sea based on observations and era5 reanalysis data. *Meteorol Z* 2023;32:207–28. <http://dx.doi.org/10.1127/metz/2022/1166>.
- [51] Hersbach H, Bell B, Berrisford P, Hirahara S, Horányi A, Muñoz-Sabater J, Nicolas J, Peubey C, Radu R, Schepers D, et al. The era5 global reanalysis. *Q J R Meteorol Soc* 2020;146:1999–2049. <http://dx.doi.org/10.1002/qj.3803>, URL: <https://rmets.onlinelibrary.wiley.com/doi/10.1002/qj.3803>.
- [52] Zhang T, Tian B, Sengupta D, Zhang L, Si Y. Global offshore wind turbine dataset. *Sci Data* 2021;8:191. <http://dx.doi.org/10.1038/s41597-021-00982-z>, URL: <https://www.nature.com/articles/s41597-021-00982-z>.
- [53] Xu K, Yu Z, Wang X, Mi MB, Yao A. Enhancing video super-resolution via implicit resampling-based alignment. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2024, p. 2546–55, URL: [https://openaccess.thecvf.com/content/CVPR2024/papers/Xu\\_Enhancing\\_Video\\_Super-Resolution\\_via\\_Implicit\\_Resampling-based\\_Alignment\\_CVPR\\_2024\\_paper.pdf](https://openaccess.thecvf.com/content/CVPR2024/papers/Xu_Enhancing_Video_Super-Resolution_via_Implicit_Resampling-based_Alignment_CVPR_2024_paper.pdf).
- [54] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. In: *Medical image computing and computer-assisted intervention—MICCAI 2015*. Springer; 2015, p. 234–41. [http://dx.doi.org/10.1007/978-3-319-24574-4\\_28](http://dx.doi.org/10.1007/978-3-319-24574-4_28), URL: [https://link.springer.com/chapter/10.1007/978-3-319-24574-4\\_28#citeas](https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28#citeas).
- [55] Jiang CM, Esmaeilzadeh S, Azizzadenesheli K, Kashinath K, Mustafa M, Tchelepi HA, Marcus P, Prabhath, Anandkumar A. Meshfreeflownet: A physics-constrained deep continuous space–time super-resolution framework. 2020, CoRR abs/2005.01463 URL: <https://arxiv.org/abs/2005.01463>. arXiv:2005.01463.
- [56] Woo S, Park J, Lee J-Y, Kweon IS. Cbam: Convolutional block attention module. In: *Computer vision – ECCV 2018: 15th European conference, Munich, Germany, September 8–14, 2018, proceedings, part VII*. Berlin, Heidelberg: Springer-Verlag; 2018, p. 3–19. [http://dx.doi.org/10.1007/978-3-030-01234-2\\_1](http://dx.doi.org/10.1007/978-3-030-01234-2_1).
- [57] Mathieu M, Couprie C, LeCun Y. Deep multi-scale video prediction beyond mean square error. 2015, arXiv preprint arXiv:1511.05440 URL: <https://arxiv.org/abs/1511.05440>.

- [58] Zhang R, Isola P, Efros AA, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2018, p. 586–95. <http://dx.doi.org/10.1109/CVPR.2018.00068>, URL: <https://ieeexplore.ieee.org/document/8578166>.
- [59] Cruz DP, Strong G, Bates O, Cueto C, Yao J, Guasch L. Convolve and conquer: Data comparison with wiener filters. 2024, URL: <https://arxiv.org/abs/2311.06558>.
- [60] Brunet D, Vrsay ER, Wang Z. On the mathematical properties of the structural similarity index. IEEE Trans Image Process 2012;21:1488–99. <http://dx.doi.org/10.1109/TIP.2011.2173206>, URL: <https://ieeexplore.ieee.org/document/6059504>.
- [61] Zhao H, Gallo O, Frosio I, Kautz J. Loss functions for image restoration with neural networks. IEEE Trans Comput Imaging 2017;3:47–57. <http://dx.doi.org/10.1109/TCI.2016.2644865>, URL: <https://ieeexplore.ieee.org/document/7797130>.
- [62] Wang Z, Bovik A, Sheikh H, Simoncelli E. Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 2004;13:600–12. <http://dx.doi.org/10.1109/TIP.2003.819861>, URL: <https://ieeexplore.ieee.org/document/1284395>.
- [63] Jiang L, Dai B, Wu W, Loy CC. Focal frequency loss for image reconstruction and synthesis. In: 2021 IEEE/CVF international conference on computer vision. ICCV, 2021, p. 13899–909. <http://dx.doi.org/10.1109/ICCV48922.2021.01366>, URL: <https://ieeexplore.ieee.org/document/9710849>.
- [64] Skamarock WC. Evaluating mesoscale nwp models using kinetic energy spectra. Mon Weather Rev 2004;132:3019–32. <http://dx.doi.org/10.1175/MWR2830.1>, URL: <https://journals.ametsoc.org/view/journals/mwre/132/12/mwr2830.1.xml>.
- [65] Nilsson J, Akenine-Möller T. Understanding ssim. NVIDIA Research; 2020, [https://research.nvidia.com/publication/2020-07\\_Understanding-SSIM](https://research.nvidia.com/publication/2020-07_Understanding-SSIM).