Proceedings of the 1st international Symposium on AI and Fluid Mechanics
Paper No S11 P11

Physics-informed Transformer-based Neural Operator for Parametric Super Resolution of Turbulent Channel Flows from Spatially Sparse Data

Yuhang Wang [1*], Sergiy Shelyag [2] and Jorg Schluter [1]

[1] School of Engineering, Deakin University, Waurn Ponds VIC 3216, Australia
[2] College of Science and Engineering, Flinders University, Tonsley, SA 5042, Australia

Keywords: Super resolution, Turbulent flows, Machine learning, Neural operator, Galerkin-type attention, Physics-informed Neural Networks.

Abstract. Over recent years, machine learning (ML)-based super resolution (SR) has been a hot interest in fluid dynamics, with its aim of reconstructing high-resolution (HR) flow fields based on their low-resolution (LR) counterparts to enhance both computational efficiency and solution accuracy. However, previous ML-based SR methods are mostly built upon neural network (NN) architecture, thus suffering from the NN limitations, including reliance on a fixed discretisation and a fixed SR scaling factor, and lack of generalisability to varied flow parameters. Therefore, we propose a physics-informed transformer-based neural operator (NO) to perform parametric super resolution of turbulent flows from spatially sparse data. The neural operator architecture comprises three components: a CNN-based encoder, an interpolation-based embedder, and an attention-type decoder. Aside from data constraint via supervised learning, a physics-informed loss function was developed to enforce physical laws of fluid flows via unsupervised learning, thus enabling the physical realism of the reconstructed HR solutions. The proposed model was then tested on direct numerical simulations (DNSs) of a 3D fully developed turbulent channel flow across varied Reynolds numbers. The results demonstrated the model's superior capabilities of reconstructing HR flow fields while transferring solutions between varied mesh resolutions and Reynolds numbers. Besides, as trained with spatially sparse data and an additional grid context, the model can achieve zero-shot parametric super resolution, thereby significantly reducing training time with improved capability of generalizing to different flow conditions.

1. Introduction

Super resolution (SR) covers a broad range of fluidic applications, including high-resolution reconstruction from spatially sparse point measurements, denoising of noisy experimental data, missing data recovery, turbulence modelling, and subgrid-scale enrichment for LESs. However, the traditional SR techniques like interpolation, sparse encoding are incapable of tackling turbulent flow problems due to the highly chaotic, multi-scale nature of turbulence. Consequently, reconstructions from those techniques tend to be over-smoothing and missing all the high-frequency detailed features within. On the other hand, machine learning (ML) has been a hot research field for super-resolution reconstruction, so-called ML-based SR methods, since they are data-driven, super computationally efficient, and can directly learn intricate latent flow patterns from existing flow databases. One typical example of those SR models is neural network (NN)-based ML models, including convolutional neural networks (CNNs) and generative adversarial network (GANs). Some previous studies [1-5] have clearly demonstrated its excellent ability to extract detailed flow features across various spatiotemporal scales, predict temporal evolution of turbulent flows, and enrich turbulence at unseen scales across various flow conditions. Nevertheless, since the NN-based SR methods are all built on NN architectures where flow data only operates on evenly spaced meshes, their predictions are mesh dependent and limited to finite-dimensional space at a fixed SR scaling factor, thereby cannot query new points beyond the meshes used for model training. Physics-informed neural networks (PINNs) [6] are proposed to overcome the mesh dependence issues by incorporating physics-informed losses to directly map the coordinates of arbitrarily sampled points to the flow quantity values at the corresponding points [7]. Despite the advantage, the PINNs also suffer from certain drawbacks. First, incomplete incorporation of physical

losses remains a concern, as their predictions heavily rely on physical laws that governs fluid flows. Second, the PINNs are constrained by a limited model architecture model architecture, solely relying on fully-connected neural networks (FCNNs) due to the use of automatic differentiation (AD) [8]. Consequently, their predictions are pointwise and cannot capture non-local flow interactions. Third, the PINNs have insufficient scalability and high computational cost for large-scale, high-dimensional data. More importantly, one PINN can only be trained to learn one instance of parametric partial differential equations (PDFs). It needs to be retrained when adjusting flow conditions, so its predictions cannot generalise to various flow conditions [9].

Therefore, we propose a novel physics-informed transformer-based neural operator to learn parametric super-resolution reconstruction of full 3D turbulent flows from spatially sparse data across a family of flow conditions. Neural operators (NOs), as a generalised version of NNs, can learning mappings between infinite dimensional function spaces, such that they can offer mesh-invariant predictions and perform super resolution across arbitrary SR scaling factors. Furthermore, they can learn the solution operator for an entire family of parametric PDEs after a single training process, thereby their predictions can generalise well across a range of flow conditions.

2. Methodology

2.1. Problem Setting

Under the continuum, incompressibility assumptions, the governing equations for turbulent flows are the 3D incompressible Navier-Stokes equations for mass and momentum, formulated as:

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \mathbf{u} \tag{2}$$

where $\mathbf{u} = [u, v, w]^T$ is the velocity vector, ∇ is the del operator, and ∇^2 is Laplacian operator, ρ, p and ν are the density, static pressure, and kinematic viscosity of the fluid, respectively.

Our goal is to reconstruct high-resolution continuum velocity and pressure fields from sparsely distributed data. Assume that a sparse data set $\{\mathbf{u}_{LR}(\mathbf{x}_i), p_{LR}(\mathbf{x}_i)\}$, where $\mathbf{x}_i \in \Omega_{LR} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, is known, the HR solution fields must satisfy the data constraints, physical constraints and boundary constraints simultaneously to achieve an accurate SR reconstruction. Such a super-resolution operator $\mathcal{S}$ can be presented as:

$$\big(\mathbf{u}(\mathbf{x}), p(\mathbf{x})\big) = \mathcal{S}(\{\mathbf{x}_i, \mathbf{u}_{LR}(\mathbf{x}_i), p_{LR}(\mathbf{x}_i)\}), \quad \forall \mathbf{x} \in \Omega \tag{3}$$

$$\text{subject to} \quad \mathbf{u}(\mathbf{x}_i) = \mathbf{u}_{LR}(\mathbf{x}_i), p(\mathbf{x}_i) = p_{LR}(\mathbf{x}_i), \quad \forall \mathbf{x}_i \in \Omega_{LR} \subset \Omega$$

$$\text{subject to} \quad \mathcal{L}(\mathbf{u}, p) = \begin{pmatrix} \nabla \cdot \mathbf{u} \\ \partial_t \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{u} + \frac{1}{\rho}\nabla p - \nu \nabla^2 \mathbf{u} \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{0} \end{pmatrix} \text{ in } \Omega$$

$$\text{subject to} \quad \mathbf{u}|_{\partial\Omega} = \mathbf{g} \text{ in } \Omega$$

where $\{\cdot\}$ denotes a set. $\mathbf{x} = [x, y, z]^T$ is any arbitrary point in the bounded domain Ω, $\mathbf{x}_i$ is any arbitrary point in the discrete domain Ω_{LR}, Ω_{LR} is the discrete domain for sparse measurements within Ω. $\mathbf{g}$ is the boundary condition.

2.2. Transformer-based Neural Operator

Owing to the advantages of NOs, the proposed transformer-based neural operator is designed to not only approximate the SR operator $\mathcal{S}$ but across a range of varied flow conditions acting as additional parameters, namely a parametric SR operator $\mathcal{S}_\theta$ presented as:

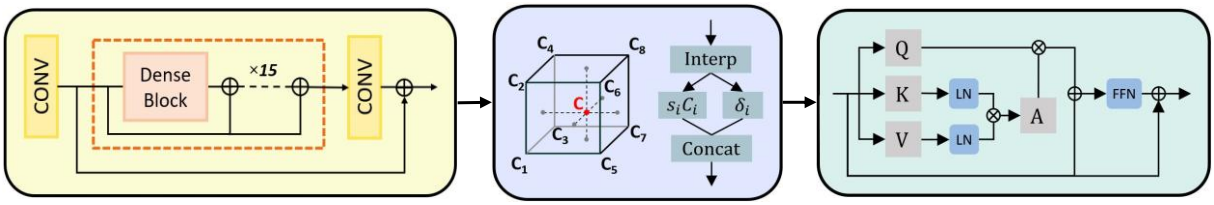
$$\mathcal{S}_\theta : \left\{\mathbf{x}_i, \left\{\mathbf{x}_j, \mathbf{q}(\mathbf{x}_j)\right\}_{j=1}^{n}\right\} \mapsto \tilde{\mathbf{q}}(\mathbf{x}_i), \qquad i = 1, \ldots, N \tag{4}$$

where $\mathcal{S}_\theta$ (parameterised by θ) employs a low-resolution uniform-grid context $\mathbf{q}(\mathbf{x}_j) = \left[\mathbf{u}_{LR}(\mathbf{x}_j), p_{LR}(\mathbf{x}_j)\right]$ as additional parameters to map sparse point coordinates $\mathbf{x}_i$ onto the corresponding approximate solutions $\tilde{\mathbf{q}}(\mathbf{x}_i) = [\tilde{\mathbf{u}}(\mathbf{x}_i), \tilde{p}(\mathbf{x}_i)]$.

As shown in figure 1, the architecture of the proposed NO consists of three components: (1) a CNN-based encoder, (2) an interpolation-based conditioning module, and (3) a Galerkin-type attention decoder. **The CNN-based encoder.** As fed with a uniform grid of point values $\left\{\mathbf{q}(\mathbf{x}_j)\right\}_j^n$, a lifting operation is performed using the CNN to elevate the input into a high-dimensional latent feature space, facilitating the extraction of complex, non-local flow features as the LR grid context. Besides, due to the convolution layers' translation equivalence, the grid position is inherently embedded into the grid context as the encoder output $\phi_{CNN}(\mathbf{x}_j)$ without an explicit position encoding. **The Interpolation-based conditioning module.** After taking in the encoder output $\phi_{CNN}(\mathbf{x}_j)$ and a coordinate set of sparse points $\{\mathbf{x}_i\}_i^N$, the module applies the trilinear interpolation scheme to embed the LR grid context into the sparse point set based on the information of the corresponding neighbouring points $\mathcal{N}(\mathbf{x}_i) = \left\{\mathbf{x}_{i,n}\right\}|_{n=1}^{8}, \forall \mathbf{x}_{i,n} \in \left\{\mathbf{x}_j\right\}$ extracted from $\phi_{CNN}(\mathbf{x}_j)$. **The Galerkin-type attention decoder.** As inspired by the Galerkin-type attention introduced by Cao [10], the decoder is an iterative architecture $\mathbf{z}_0 \mapsto \mathbf{z}_1 \mapsto \cdots \mapsto \mathbf{z}_T \mapsto \tilde{\mathbf{q}}$, where $\mathbf{z}_0(\mathbf{x}_i)$ is the linear transformation of the module output $C\left(\mathbf{x}_i, \left\{\phi_{CNN}(\mathbf{x}_{i,n})\right\}_n^8\right)$, and each $\mathbf{z}_t \mapsto \mathbf{z}_{t+1}$ is the mapping between two vector-valued functions, formulated as:

$$\mathbf{z}_{t+1}(\mathbf{x}_i) = \sigma\left(W_t^1\left(\mathcal{K}_t(\mathbf{z}_t; \theta_{attn})(\mathbf{x}_i) + \mathbf{z}_t(\mathbf{x})\right)\right) \tag{5}$$

where W_t^1 is a linear transformation, σ is a non-linear activation, $\mathcal{K}_t(\mathbf{z}_t; \theta_{attn})$ is a kernel integral transformation which can be approximated by a multi-head Galerkin-type attention layer via the Monte Carlo method. Eventually, the transformer-based NO output is the projection of $\mathbf{z}_T$ using a single-layer feedforward neural network (FNN): $\tilde{\mathbf{q}}(\mathbf{x}) \approx \tilde{\mathbf{q}}(\mathbf{x}_i) = P(\mathbf{z}_T(\mathbf{x}_i))$.



(1) CNN-based Encoder (2) Interpolation-based Conditioning Module (3) Galerkin-type Decoder

Figure 1. Architecture of the proposed transformer-based neural operator.

2.3. Physics-informed Loss Function

To enforce the physical constraints of the model's reconstruction, we propose a physics-informed loss function by incorporating the residuals of mass, pressure and energy as additional physical losses to complement the standard mean squared (MS) loss. Therefore, the whole loss function $\mathcal{L}$ is shown as:

$$\mathcal{L} = w_1\mathcal{L}_{MSE} + w_2\mathcal{L}_{continuity} + w_3\mathcal{L}_{pressure} + w_4\mathcal{L}_{enstrophy} \tag{6}$$

where $[w_1, w_2, w_3, w_4]$ are weights for each loss term, $\mathcal{L}_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\|\mathbf{q}(\mathbf{x}_i) - \tilde{\mathbf{q}}(\mathbf{x}_i)\|$, $\mathcal{L}_{continuity} = \frac{1}{N}\sum_{i=1}^{N}|\nabla \cdot \tilde{\mathbf{u}}(\mathbf{x}_i)|$, $\mathcal{L}_{pressure} = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{1}{\rho}\left(\nabla^2\tilde{p}(\mathbf{x}_i)\right) + \nabla \cdot \left(\nabla \cdot \tilde{\mathbf{u}}(\mathbf{x}_i)\tilde{\mathbf{u}}(\mathbf{x}_i)\right)\right|$, and $\mathcal{L}_{enstrophy} =$

$\frac{1}{2N}\sum_{i=1}^{N}|\boldsymbol{\omega}(\mathbf{x}_i)^2 - \tilde{\boldsymbol{\omega}}(\mathbf{x}_i)^2|$. Now, the super-resolution problem transitions into an optimisation problem to be tackled using the Adam optimiser [11].

2.4. Data Preparation

In terms of data for model training, a 3D fully developed turbulent channel flow is selected as a canonical flow configuration. The channel flow, simulated using DNS, has a computational domain with a length of 10 m, a height of 1 m, and a depth of 2 m. Periodic boundary conditions are applied along the length and depth while no-slip wall conditions are used along the height to mimic a rectangular duct with a fixed height, and an extensive length and height. The bulk inflow velocity u_b is normalised to 1 m/s, and four Reynolds numbers ($\mathrm{Re} = 1/\nu = 6000, 12000, 18000$ and 24000) are considered by adjusting the kinematic viscosity ν.

During data preparation, 3D instantaneous snapshots of four flow quantities (u, v, w, p) are extracted as HR fields from the DNS database. According to figure 2, in each HR field, a number of HR patches with varied sizes $(ns)^3$ are randomly sampled depending on arbitrary SR scaling factors s selected from a range. Then, in each HR patch, the whole field is downsampled via the trilinear interpolation into a LR patch sized s^3. s^3 point values are randomly sampled from within to create a random value patch, while the corresponding coordinates of those points coordinates are recorded to form a random coordinate patch. Lastly, a grid of point values and point coordinates are randomly cropped from within to form a grid coordinate patch and a grid value patch. Both are sized s^3, respectively. The LR patch is used as the encoder input while the random patch and grid patch are for the conditioning module.

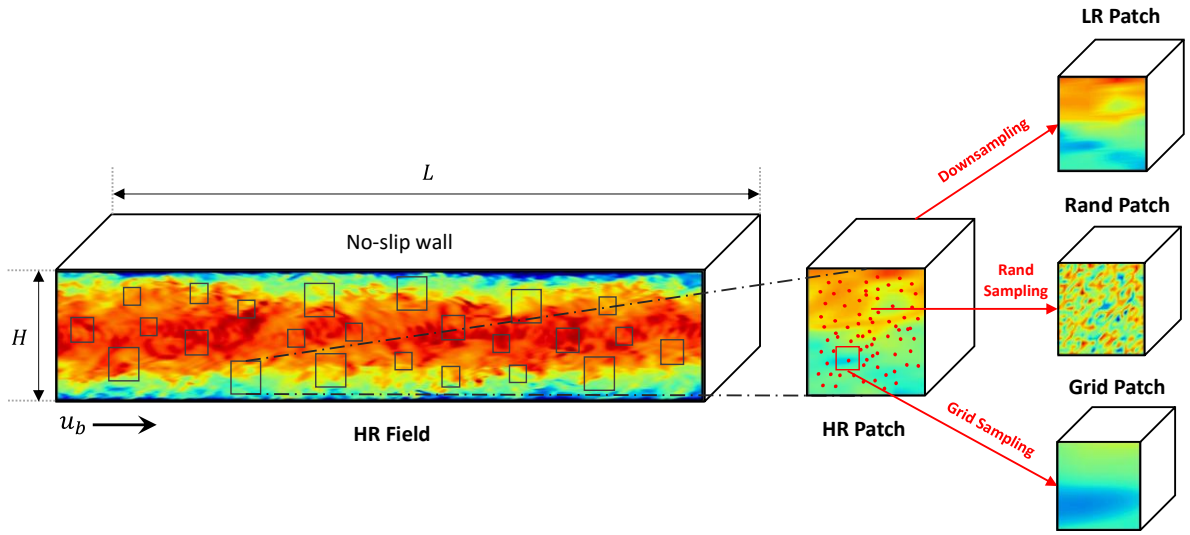


Figure 2. Data preparation process for model training.

3. Results and Discussion

Once trained, the performance of the proposed model is examined on a fully developed turbulent channel flow at $\mathrm{Re} = 18000$ across varied SR scaling factors. Figure 3 compares 2D slices (x-y plane) of 3D model reconstructions on streamwise velocity fields against ground truth at three different SR scaling factors. It is shown that the model predictions achieve high accuracy with a relative error of 5% at all three factors, although an increasing trend in errors is also observed as the factor gets larger. This demonstrates the model's good capability of capturing intricate turbulence features at varied scales resulting from the interactions between bulk turbulent flows and no-slip wall boundaries. Regarding physical realism, the residuals of continuity equation and Poisson equation for pressure along with enstrophy field are measured on the model reconstructions at the SR scaling factor of 4, illustrated in figure 4. As a result, the model can achieve minimal residuals on both continuity and pressure and a

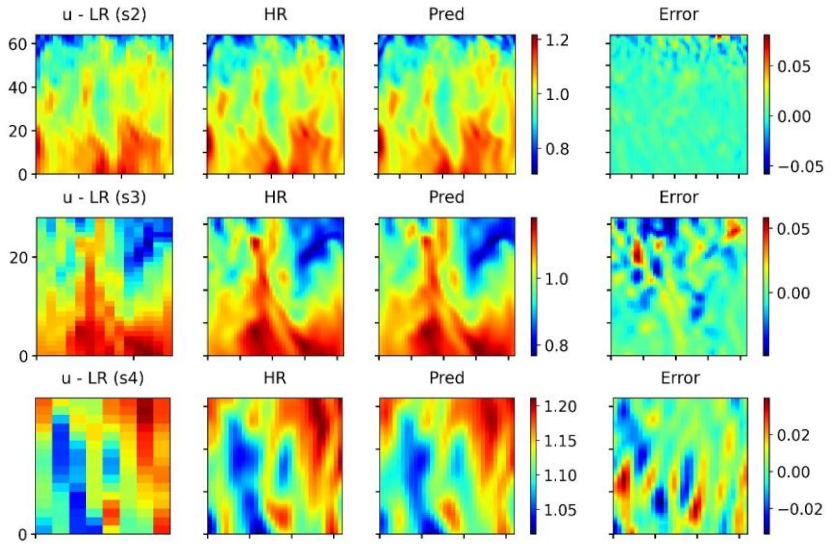


Figure 3. Comparison of predicted streamwise velocity fields (Pred) against ground truth (HR) at varied SR scaling factors (top: 2, middle: 3, bottom: 4). All contours are 2D slices (x-y plane) extracted from 3D flow fields.

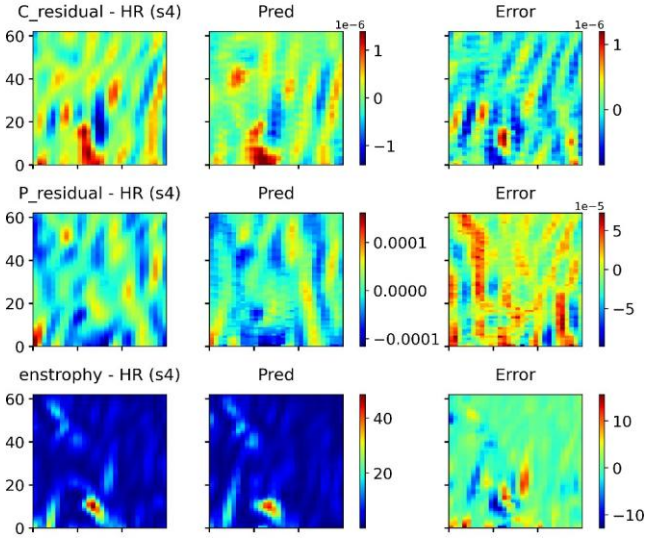


Figure 4. Comparison of predicted physical fields (top: continuity residual, middle: pressure residual, bottom: enstrophy) against ground truth at a SR scaling factor of 4. All contours are 2D slices (x-y plane) extracted from 3D flow fields.

good prediction on the peak value of enstrophy, exhibiting its promising ability to capture the conservation of mass and pressure equations as well as strong vortices caused by the Kelvin-Helmholtz instability. Moreover, near-wall flow statistics including normalised mean velocity $\overline{u}^+$ and Reynolds shear stress $\overline{u'v'}^+$ are presented in figure 5. The model predictions are observed having a near-perfect agreement with the true results regarding the mean velocity and most regions of the Reynolds shear stress except for the area around the peak value. It indicates that for $15 < y^+ < 45$ where flow transitions from the buffer layer to the log-law region, both streamwise and crosswise fluctuations become so intense due to weakened viscosity and strong velocity gradients, such that a broader range of turbulence structures are formed. Consequently, the model fails to capture all the turbulence structures and thereby underpredicts the intensity of the shear stress.

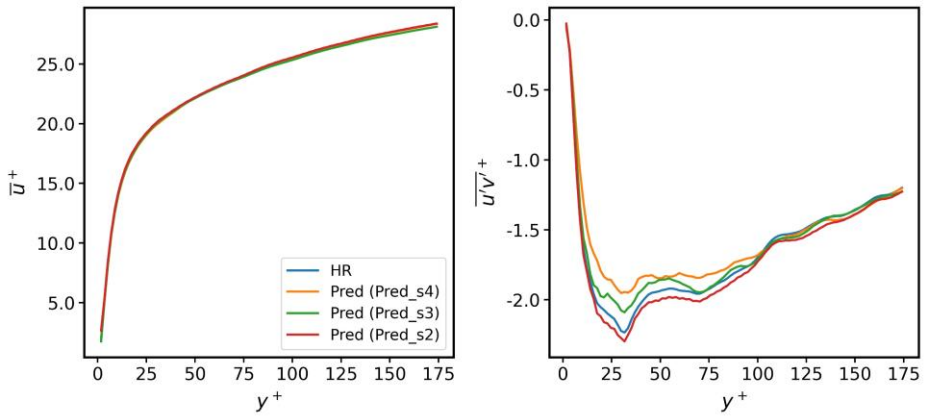


Figure 5. Comparison of the predicted mean velocity $\overline{u}^+$ and Reynolds stress component $\overline{u'v'}^+$ against ground truth (HR) at varied SR scaling factors (2, 3, and 4).

4. Conclusions

In this study, a physics-informed transformer-based neural operator is proposed to perform super-resolution reconstruction of turbulent channel flows from spatially sparse data. The proposed model contains a conditioning module and a NO architecture to achieve parametric super resolution while ensuring physically accurate solutions by training with a physics-informed loss function. As compared to standard NN-based SR methods, our proposed model can offer mesh-invariant predictions due to operator learning and better interpretability by leveraging the physical insight of the Petrov-Galerkin approach. Furthermore, as trained only using LR and sparsely distributed data, the model minimises HR data requirement and thus can achieve a significant time reduction for model training. However, the current model is identified having a limit to accurately capture intense fluctuations at near-wall regions, unless more training data points are provided (using lower SR scaling factors). Therefore, in the future study, we tend to conduct a comprehensive hyperparameter search to further improve model performance, particularly at high Res. Additionally, its adaptation to unstructured meshes will also be explored to better capture near-wall boundary conditions.

5. References

[1] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence,* vol. 38, no. 2, pp. 295-307, 2015.

[2] K. Fukami, K. Fukagata, and K. Taira, "Super-resolution reconstruction of turbulent flows with machine learning," *Journal of Fluid Mechanics,* vol. 870, pp. 106-120, 2019.

[3] B. Liu, J. Tang, H. Huang, and X.-Y. Lu, "Deep learning methods for super-resolution reconstruction of turbulent flows," *Physics of fluids,* vol. 32, no. 2, 2020.

[4] A. Subramaniam, M. L. Wong, R. D. Borker, S. Nimmagadda, and S. K. Lele, "Turbulence enrichment using physics-informed generative adversarial networks," *arXiv preprint arXiv:2003.01907,* 2020.

[5] M. Z. Yousif, L. Yu, and H.-C. Lim, "Super-resolution reconstruction of turbulent flow fields at various Reynolds numbers based on generative adversarial networks," *Physics of Fluids,* vol. 34, no. 1, 2022.

[6] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational physics,* vol. 378, pp. 686-707, 2019.

[7] P. Clark Di Leoni, K. Agarwal, T. A. Zaki, C. Meneveau, and J. Katz, "Reconstructing turbulent velocity and pressure fields from under-resolved noisy particle tracks using physics-informed neural networks," *Experiments in Fluids,* vol. 64, no. 5, p. 95, 2023.

[8] C. C. Margossian, "A review of automatic differentiation and its efficient implementation," *Wiley interdisciplinary reviews: data mining and knowledge discovery,* vol. 9, no. 4, p. e1305, 2019.

[9] N. Kovachki *et al.*, "Neural operator: Learning maps between function spaces with applications to pdes," *Journal of Machine Learning Research,* vol. 24, no. 89, pp. 1-97, 2023.

[10] S. Cao, "Choose a transformer: Fourier or galerkin," *Advances in neural information processing systems,* vol. 34, pp. 24924-24940, 2021.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.