Turbulence Enrichment with Physics-informed Generative Adversarial Network

Akshay Subramaniam[*] **Man-Long Wong, Raunak Borker, Sravya Nimmagadda and Sanjiva Lele**
Department of Aeronautics & Astronautics
Stanford University, Stanford, CA, USA

Abstract

Generative Adversarial Networks (GANs) have been widely used for generating photo-realistic images [1, 2, 3]. In this work, we develop physics-informed methods for generative enrichment of turbulence. We incorporate a physics-informed learning approach to minimize the residuals of the governing equations for the generated data. We analyze two physics-informed models including a GAN model, and show that they outperform tricubic interpolation. We also show that using physics-informed learning can significantly improve the model's ability to generate data that satisfies the physical constraints. Finally, we analyze the generated enriched data to show that it is able to recover statistical metrics of the flow field including energy metrics.

Predicting turbulence accurately is extremely challenging especially in capturing high-order statistics due to its intermittent nature. While numerical methods provide a path to attain approximate solutions, the computational resources required for engineering applications can be quite large, especially when the flows are highly turbulent, where a broad range of spatial and temporal scales appear. As a result, reduced-order modeling of turbulent flows for low cost computations has been a popular research area in fluid dynamics for decades. The use of machine learning (ML) for turbulence modeling has not received much attention until recent years [4, 5, 6, 7, 8, 9, 10, 11]. All of these approaches have focused on improving traditional turbulence models in some way. In some other works [12, 13], physics-informed neural networks (NNs) were developed but only tested on 1D problems.

Ledig et al. [14] developed a GAN for super-resolution of images with state-of-the-art results. GANs have been shown to perform better than other data driven approaches like principal component analysis (PCA) in capturing high-order moments [15], and thus may be beneficial for turbulence modeling. Directly extending the super-resolution GAN (SRGAN) [14] for turbulence enrichment of 3D low resolution turbulence data may not be physically realistic as physical constraints are not enforced. Incorporating the physical constraints into deep NNs (DNNs) such as the SRGAN framework would be crucial to its performance in this context.

In this work, we propose a convolutional NN (CNN)-based residual block neural network [16] (TEResNet) and a generative model based on SRGAN [14] (TEGAN) to enrich low resolution turbulent fields with high wavenumber content using physics-informed technique. The input to our model is a low resolution turbulent flow field that consists of four 3D fields. We then apply both TEResNet and TEGAN to enrich each of these four fields by a factor of four in each spatial dimention. Our results show that the proposed TEResNet and TEGAN models have better performance than using tricubic interpolation for upsampling and the turbulence enriched data compares well with the high resolution data.

[*] Email: `akshays@alumni.stanford.edu`. Code used for this work can be found at `https://github.com/akshaysubr/TEGAN`.

Third Workshop on Machine Learning and the Physical Sciences (NeurIPS 2020), Vancouver, Canada.

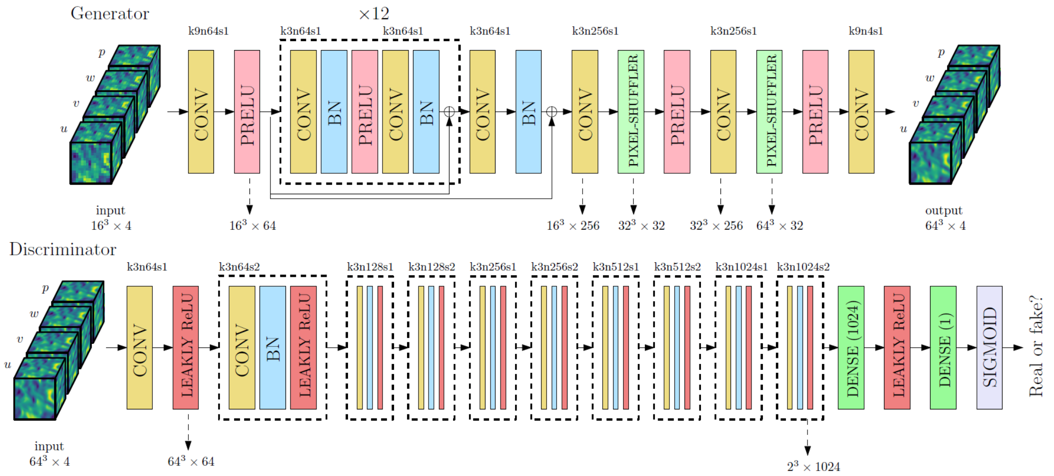

Figure 1: Architecture of generator and discriminator network with corresponding kernel size (k), number of channels (n) and stride (s) indicated for each convolutional layer.

1 Problem definition and Dataset

The demonstration problem that we have chosen is the incompressible forced isotropic homogeneous turbulence problem in a 3D periodic domain with size $[0, 2\pi) \times [0, 2\pi) \times [0, 2\pi)$. From the incompressible Navier–Stokes equations, the physical constraints for velocity and pressure fields are given by the continuity and pressure Poisson equations respectively:

$$\nabla \cdot \boldsymbol{u} = 0, \qquad -\nabla^2 p = \nabla \boldsymbol{u} : \nabla \boldsymbol{u}^T, \tag{1}$$

where $\boldsymbol{u} = [u, v, w]^T = [u_1, u_2, u_3]^T$ and p are the velocity vector, and kinematic pressure respectively.

We perform a time-evolving Direct Numerical Simulation (DNS) on a $N \times N \times N$ uniform grid and collect snapshots separated in time by more than one integral time scale. This ensures that each sample is statistically decorrelated from the rest of the samples. Each snapshot is comprised of four fields - u, v, w and p. Low resolution data is then generated by filtering the high resolution data down to $N/4 \times N/4 \times N/4$ using a compact support explicit filter. The filter is derived as an approximation to the sharp spectral low-pass filter at cutoff of a quarter of the Nyquist wavenumber with a compact stencil. The velocity components of the high resolution data are normalized (and non-dimensionalized) by the root mean square of the velocity magnitude and the pressure by the mean square of the velocity magnitude. An 80%-10%-10% split of the dataset is used for training, testing and validation respectively. Sample images of the high and low resolution data are presented in Section 3.

2 Methods

In this section, we describe the architecture and training methodology of the models employed for the task of turbulence enrichment. The proposed GAN (TEGAN) is contrasted with a convolutional residual network (TEResNet) which has the same architecture as the generator in TEGAN.

For the task of upsampling the low resolution data in a physically consistent manner, we use a GAN[1] in a fashion similar to super-resolution applications for image data [14]. The generator has a deep residual network architecture with each residual block having convolutional layers with batch normalization. The discriminator has a deep convolutional architecture with fully connected layers in the end for binary classification. The architectures of the generator and discriminator are depicted pictorially in Figure 1.

2.1 Loss functions and physics-informed learning

As discussed in a previous section, the flow fields are constrained by Equation (1). These equations might not be satisfied by the model's generated output. To counter this, the residuals of these equations can be used as regularizers for the model through a physics loss term. The loss function minimized for the generator network during training is a combination of a content loss $\mathcal{L}_{\text{content}}$ and a physics loss $\mathcal{L}_{\text{physics}}$.

$$\mathcal{L}_{\text{GAN}} = (1-\lambda_{\text{A}})\,\mathcal{L}_{\text{resnet}} + \lambda_{\text{A}}\mathcal{L}_{\text{adversarial}}, \qquad \mathcal{L}_{\text{resnet}} = (1-\lambda_{\text{P}})\,\mathcal{L}_{\text{content}} + \lambda_{\text{P}}\mathcal{L}_{\text{physics}},$$
$$\mathcal{L}_{\text{content}} = (1-\lambda_{\text{E}})\,\mathcal{L}_{\text{MSE}} + \lambda_{\text{E}}\mathcal{L}_{\text{enstrophy}}, \qquad \mathcal{L}_{\text{physics}} = (1-\lambda_{\text{C}})\,\mathcal{L}_{\text{pressure}} + \lambda_{\text{C}}\mathcal{L}_{\text{continuity}}.$$

where $\mathcal{L}_{\text{MSE}}$ is the mean squared error between the high resolution and generated fields and $\mathcal{L}_{\text{enstrophy}}$ is the mean squared error in the derived enstrophy field Ω ($\Omega = \boldsymbol{\omega}\cdot\boldsymbol{\omega}, \text{ where } \boldsymbol{\omega} = \nabla\times\boldsymbol{u}$) to sensitize the generator to high wavenumber content. Residuals of the continuity ($\mathcal{L}_{\text{continuity}}$) and pressure Poisson ($\mathcal{L}_{\text{pressure}}$) equations given above similar to [12] make up the physics loss. The inclusion of physics loss forces the network to generate physically more realizable solutions. Finally, $\mathcal{L}_{\text{adversarial}}$ is the adversarial logistic loss function similar to that defined in [14]. To train the discriminator, we use the logistic loss based on predicted labels for real and generated data.

2.2 Training

A model with just the residual network generator and without the adversarial component is termed TEResNet. We first train TEResNet to convergence and tune hyperparameters like the number of residual blocks and the physics loss parameters. The model with both the residual network generator and the discriminator depicted above is termed TEGAN. The generator in TEGAN is first initialized using the weights from a trained TEResNet while the discriminator is initialized using the Xavier–He initialization [17, 18]. For the first few iterations in the training process (~ 300), only the discriminator alone is trained to negate the advantage that the generator has because of its pre-trained weights. Then, both the generator and discriminator are trained together with the active adversarial loss until the losses saturate and the discriminator's output saturates at 0.5. The Adam optimizer [19] is used for updating the weights and training both the networks.

3 Experiments

All experiments presented in this section are performed with $N = 64$. The Taylor scale Reynolds number R_λ is ~ 30. A total of 1160 snapshots were randomly split into 920, 120 and 120 as the train, test and validation datasets.

3.1 Training convergence

A batch size of 5 is chosen for training of both TEResNet and TEGAN because of memory constraints of the GPU used for training. We choose $\alpha = 1.0\text{e}{-4}$ as the learning rate based on a hyper-parameter search. We also examine the effect of the physics loss weight λ_{P} through experiments on TEResNet. Figure 2 shows the content and physics losses during training of TEResNet with different values for λ_{P}. We see that adding a non-zero weight to the physics loss improves the physics residual by almost an order of magnitude. We choose $\lambda_{\text{P}} = 0.125$ as this gives good compromise between the two losses. Another interesting observation is that for higher weightage to the physics loss, the trivial solution of zero fields becomes a local minimum.

To improve the stability for training TEGAN, we add a staircase decay for the learning rate. We set the decay rate to 0.5 and choose a decay step of 400 by running a case without learning rate decay and estimating the number of steps required to get close to the minimum. Figure 2 shows the convergence of TEGAN during training. It can be seen that the discriminator output for generated data saturates at 0.5 and the physics loss converges to a smaller value compared to the initial value from TEResNet.

3.2 Model evaluation

Figure 3 compares the qualities of upsampled data from tricubic interpolation, TEResNet and TEGAN. Both TEResNet and TEGAN outperform the tricubic interpolation in reconstructing small-scale features. This is also evident from the plots of the velocity energy spectra in Figure 4. The output

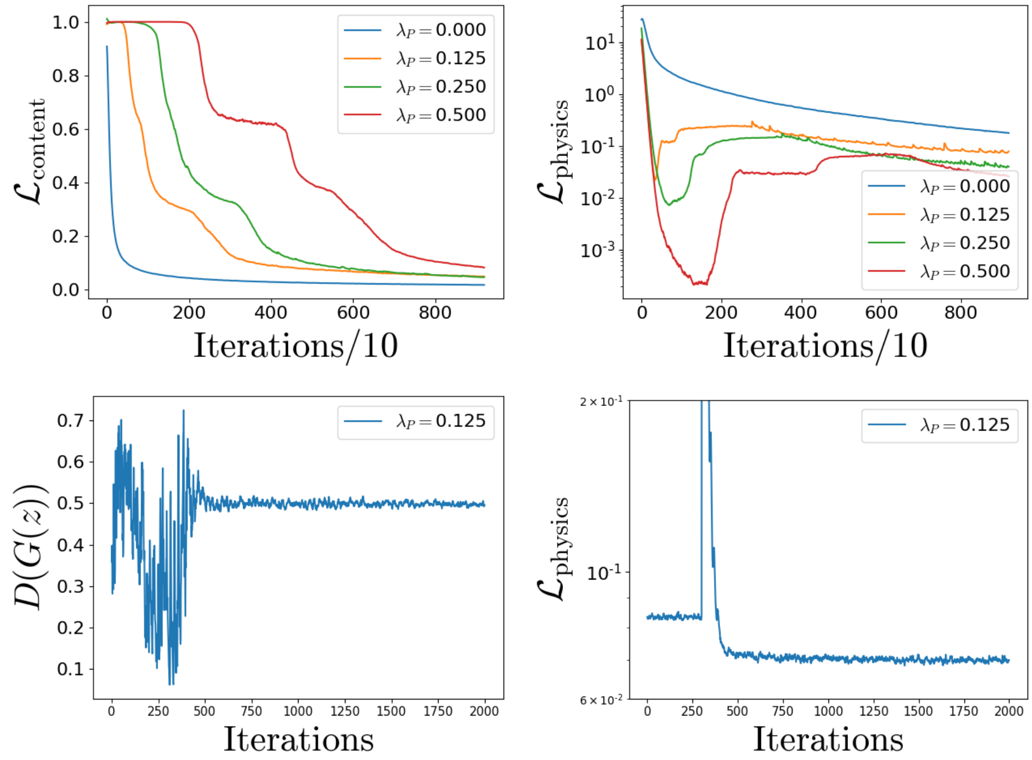

Figure 2: Content loss (top left); physics loss (top right); discriminator output for generated data (bottom left); physics loss of TEGAN (bottom right).

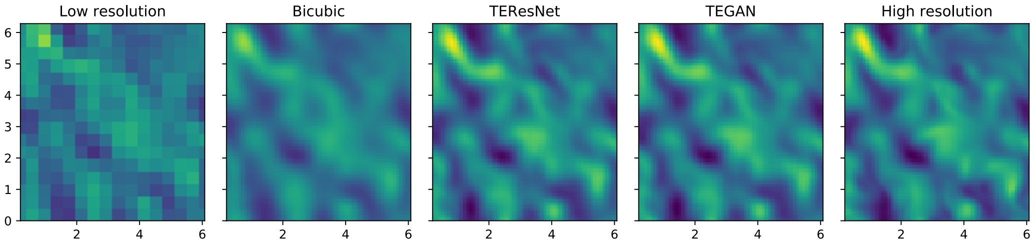

Figure 3: Plots are of the u component of velocity on a slice of the 3D field. From left to right: low resolution, tricubic interpolation, TEResNet, TEGAN and high resolution fields.

from TEResNet and TEGAN are visually indistinguishable but Table 1 shows that there is a $> 10\%$ improvement when using TEGAN over TEResNet in minimizing the physics loss while the content losses of both models are similar. This implies that while both TEResNet and TEGAN perform similarly when comparing the output fields to the high resolution data, TEGAN respects the physical equations better. This helps TEGAN generalize better as is seen in Table 1 when evaluating on the test dataset.

	$\mathcal{L}_{\text{content}}$		$\mathcal{L}_{\text{physics}}$	
	Dev	Test	Dev	Test
TEResNet	0.049	0.050	0.078	0.085
TEGAN	0.047	0.047	0.070	0.072
% Difference	4.1	6.0	10.3	15.2

Table 1: Comparison of losses between TEResNet and TEGAN at the end of training.

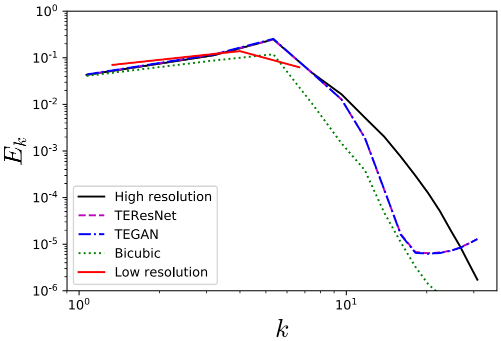

Figure 4: Comparison of the velocity energy spectra for the different upsampling methods.

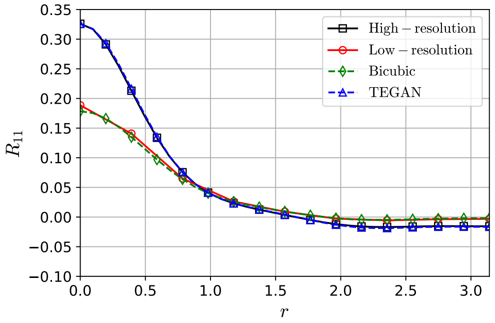

Figure 5: Comparison of the longitudinal two-point correlation function for the different upsampling methods.

Figure 5 shows the longitudinal two-point correlation function (see section 6.3 in [20]) for the different methods. TEGAN is able to faithfully recover the function compared to the true high resolution values. This implies that TEGAN recovers the energy in the fine scales as well as accurately represents the statistical properties of the turbulent fields including derived statistical quantities like the integral length scale.

4 Conclusions

In this work, we presented two physics-informed models, TEGAN and TEResNet, to enrich low resolution turbulence data with finer scales using physics-based losses as regularizers. We have shown the impacts of physics-based losses for improved performance of the networks. Both TEResNet and TEGAN outperform traditional tricubic interpolation. Low resolution data enriched by TEGAN can reconstruct important turbulent quantities such as the high wavenumber content in energy spectrum and the longitudinal two-point correlation function. The TEGAN architecture can be further extended with Wasserstein GAN models [21, 3] to improve training stability. Also, using wider distributions of data and tasking the discriminator with physics-based classification along with discrimination can be explored for better performance of the TEGAN in the future.

Broader Impact

The problem of physics-informed turbulence enrichment which we address is of broad interest in many application domains, including wind farm modeling and the ‘grey zone’ modeling in atmospheric science (where models of one fidelity are combined with models of different fidelity) and in eddy-resolving simulations of engineering applications. We are not aware of any ethical issues or societal concerns associated with this work. This work is of fundamental nature and not targeted at a specific system, and we are not aware of any group or community which could be put at a disadvantage by this work.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[2] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5769–5779, 2017.

[4] Michele Milano and Petros Koumoutsakos. Neural network modeling for near wall turbulent flow. *Journal of Computational Physics*, 182(1):1–26, 2002.

[5] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

[6] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52, 2019.

[7] Ze Jia Zhang and Karthikeyan Duraisamy. Machine learning methods for data-driven turbulence modeling. In *22nd AIAA Computational Fluid Dynamics Conference*, page 2460, 2015.

[8] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[9] J Nathan Kutz. Deep learning in fluid dynamics. *Journal of Fluid Mechanics*, 814:1–4, 2017.

[10] Kai Fukami, Yusuke Nabae, Ken Kawai, and Koji Fukagata. Synthetic turbulent inflow generator using machine learning. *Physical Review Fluids*, 4(6):064603, 2019.

[11] Corentin J Lapeyre, Antony Misdariis, Nicolas Cazard, Denis Veynante, and Thierry Poinsot. Training convolutional neural networks to estimate turbulent sub-grid scale reaction rates. *Combustion and Flame*, 203:255–264, 2019.

[12] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017.

[13] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017.

[14] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint*, 2016.

[15] Shing Chan and Ahmed H Elsheikh. Parametrization and generation of geological models with generative adversarial networks. *arXiv preprint arXiv:1708.01810*, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256, 2010.

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] Stephen B Pope. Turbulent flows, 2001.

[21] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.