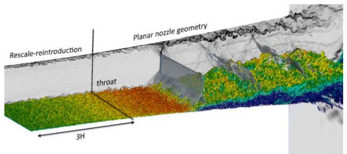
Turbulence Enrichment using Generative Adversarial Networks

Akshay Subramaniam, Man-Long Wong, Raunak Borker and Sravya Nimmagadda

{akshays,wongml,rborker,sravya}@stanford.edu

Introduction

Turbulent flow is important in many engineering applications. However, simulating turbulence is computationally very expensive due to extremely high resolution requirements. Large Eddy Simulations (LES) that simulate only the large scales have become popular due to their much lower cost, but require modeling of the small scales. Here, we propose to enrich LES data by populating it with small scales obtained using a Generative Adversarial Network [1] (GAN).



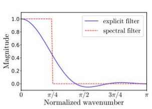
Typical simulation of a turbulent flow
ref: B. Olson, LLNL

Problem Description

Aim: Given a low resolution realization of a flow field, can we generate a physically realistic upsampled field that satisfies the governing equations?

Data

- High-resolution (HR) data is generated by numerically solving the governing equations given by the incompressible Navier-Stokes equations using an in-house solver (PadeOps) on 64 processors and collecting 1280 snapshots in time
- Each snapshot is comprised of four fields: 3 components of the velocity vector (u, v, w), and the kinematic pressure (p) each of size $64 \times 64 \times 64$
- Low-resolution data is generated by filtering the HR data down to $16 \times 16 \times 16$ using the explicit filter shown below that's derived as a best approximation to the sharp spectral filter

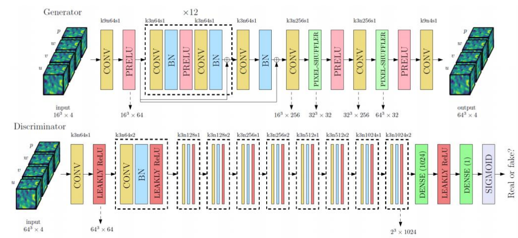


Transfer function of the filter

- Train/Dev/Test split:
920 (79.3%)/120 (10.3%)/120 (10.3%)

Model

The architecture of TEGAN is similar to that in SRGAN [2]



Training methodology

- TEResNet - the residual network generator without an adversarial component - is trained first with different no. of residual blocks and physics loss parameters
- The discriminator is trained for few iterations without updating the generator
- Train TEGAN (both generator and discriminator)

3D filter with periodic padding is used in the convolutional layer of the generator and discriminator networks.

Loss Functions

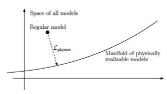
The flow field is constrained by the continuity and pressure Poisson equations:

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

Loss function minimized for the generator network during training is a combination of

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

- **Content loss**: $\mathcal{L}_{\text{content}}$
$\mathcal{L}_{\text{MSE}}$: Mean squared error between the high resolution and generated fields
$\mathcal{L}_{\text{enstrophy}}$: Mean squared error in the derived enstrophy field Ω $(\Omega = \boldsymbol{\omega} \cdot \boldsymbol{\omega},$ where $\boldsymbol{\omega} = \nabla \times \boldsymbol{u})$ to sensitize the generator to high frequency content
- **Physics loss**: $\mathcal{L}_{\text{physics}}$
Residuals of the continuity ($\mathcal{L}_{\text{continuity}}$) and pressure Poisson ($\mathcal{L}_{\text{pressure}}$) equations given above similar to [3]

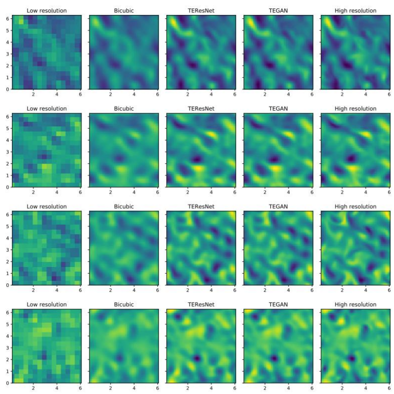


- **Adversarial loss**: $\mathcal{L}_{\text{adversarial}}$

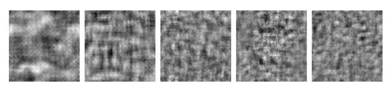
To train the discriminator, we use the logistic loss based on predicted labels for real and generated data.

Results

Comparison to bicubic interpolation and the ground truth

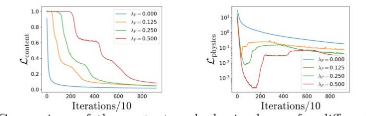


Comparisons of u, v, w velocity fields and pressure p from top to bottom

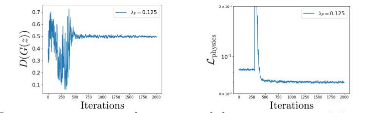


Evolution of the continuity residual during training

Discussion



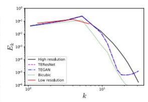
Comparison of the content and physics losses for different physics loss weights in TEResNet. The steps observed in the content loss correspond to local minima of the physics loss as seen in the figure on the right.



Discriminator output for generated data saturates at 0.5 and the physics loss of TEGAN is smaller than that of the original TEResNet.

	$\mathcal{L}_{\text{content}}$		$\mathcal{L}_{\text{physics}}$	
	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 comparing the content and physics losses on the dev and test datasets for the TEResNet and TEGAN models



Energy spectra of the generated velocity fields

Conclusion & Future Work

- While both TEResNet and TEGAN outperform traditional bicubic interpolation, it's the TEGAN that best captures the physics
- Use WGAN-GP for improved learning stability
- Include more variety in the training data (using different filters, experimental data, etc.)
- Task the discriminator with physics based classification along with discrimination for improved performance

References

[1] I. Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[2] C. Ledig et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: *arXiv preprint* (2016).

[3] M. Raissi et al. "Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations". In: *arXiv preprint arXiv:1711.10561* (2017).