Super-Resolution Reconstruction of Particle-Laden Turbulent Flows Using Conditional Deep learning

H. Tofighian[1,2], J. Denev[2], N. Kornev[1]

[1] Chair of Modelling and Simulation, *University* of Rostock, Rostock 18051, Germany

[2] Scientific Computing Center, Department of Scientific Computing and Mathematics, Karlsruhe Institute of Technology, 13 Karlsruhe 76131, Germany

Keywords: turbulence, DNS, particle-laden flows, Generative Adversarial Network

Abstract. The paper introduces the Conditional Enhanced Super-Resolution Generative Adversarial Network (CESRGAN) for reconstructing high-resolution turbulent velocity fields from low-resolution inputs. CESRGAN consists of a conditional discriminator and a conditional generator, the latter being called CoGEN. CoGEN incorporates subgrid-scale (SGS) turbulence kinetic energy as conditional information, improving the recovery of small-scale turbulent structures with the desired level of energy that match the target level of detail in the original flow field. By being aware of SGS turbulence kinetic energy, CoGEN is relatively insensitive to the degree of detail in the input. Its advantages become more pronounced when the model is applied to heavily filtered input. The model is evaluated using direct numerical simulation (DNS) data of forced homogeneous isotropic turbulence. The results show that the proposed CoGEN reconstructs fine-scale vortical structures more precisely compared to the traditional generator. Particle-pair dispersion simulations validate the physical fidelity of CoGEN-reconstructed fields, closely matching DNS results across various Stokes numbers and filtering levels.

1.Introduction

Resolving small-scale turbulence remains a fundamental challenge in flow modelling. Among the existing methods, several directions can be distinguished. For instance, the stochastic approximate-deconvolution (AD) technique proposed in [2] combines deconvolution with stochastic methods and kinematic simulation models to capture subgrid-scale effects. In Pseudo-Direct Numerical Simulation (P-DNS) [3] the fine scale motion is solved at the stage of preliminary calculation in simplified representative volume elements (RVE) with various boundary conditions written in dimensionless form. The fine scale solutions are stored in a database, which according to the authors is problem-independent, and then it is utilized in coarse simulations. In VπLES method [4] the subgrid fluctuations are restored using a solution obtained for large resolved scales. The method is based on the idea of dividing the flow into large-scale and small-scale motions, with the first being solved on a grid, and the second using the vortex particle method. The method presented in our work leverages recent advances in machine learning to reconstruct turbulent fields. We introduce a novel conditional deep learning technique to enhance the recovery of small-scale turbulent structures from low-resolution inputs.

2. Model description

The architecture builds upon ESRGAN, a generative adversarial network (GAN) framework comprising two adversarially trained neural networks [5]: a generator (G) and a discriminator (D). For super-resolution applications, the generator learns to reconstruct high-resolution (HR) data from low-resolution (LR) inputs [6], while the discriminator evaluates the quality of generated samples. The original ESRGAN was enhanced in [1] through the incorporation of input conditioning, resulting in a model named CESRGAN. In CESRGAN, the generator is conditioned on the subgrid stress kinetic energy (k_{SGS}) and the discriminator is conditioned on the corresponding LR velocity field as supplementary input. This dual conditioning ensures the discriminator critiques HR outputs in the context of their LR counterparts.

The architecture of the Generator and Discriminator are shown in Figure 1. The generator and discriminator primarily use convolutional layers with a kernel size of 3^3. The generator's backbone is a Residual in Residual Dense Block (RRDB), formed by linking multiple Residual Dense Blocks (RDB) with a residual scaling factor of $\beta = 0.2$. The generator includes four RRDBs, each with three RDBs, as shown in Figure 1. RDBs consist of convolutional layers with leaky ReLU activations and dense skip connections to enhance detail extraction and prevent gradient issues. Post-RRDBs, two interpolation blocks upsample the embedding by a factor of 4. The discriminator consists of stacked CNN layers, Leaky ReLU activations, and Batch Normalization. A dropout block is included to prevent overfitting.

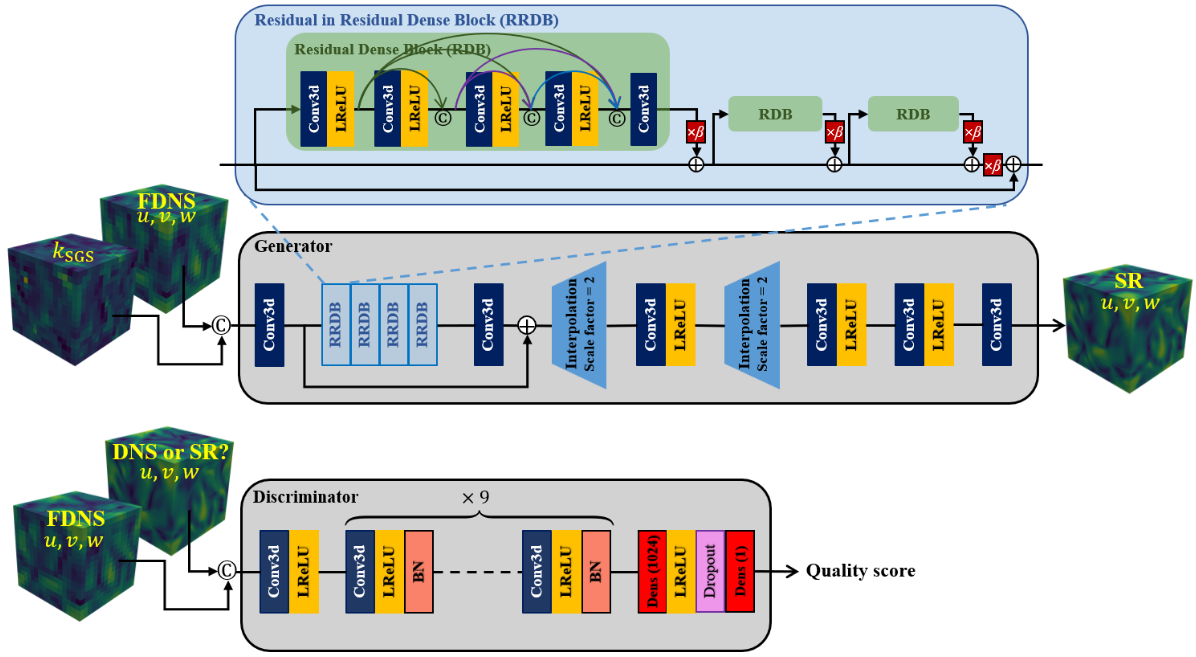

Figure 1. Overview of the conditional generator (CoGEN) and conditional discriminator architecture. The generator receives filtered DNS (FDNS) velocity components (u, v, w) and k_{SGS} to generate super-resolution (SR) velocity fields. The ⊕ symbol represents element-wise summation, while © symbol indicates concatenation. The discriminator evaluates the realism of HR input velocity fields.

The generated output is mathematically represented as $Y_g = G(X|k_{SGS})$, where Y_g corresponds to a sample set drawn from the generated HR data distribution (P_{Y_g}) and X is a sample set of LR inputs. In CoGEN, the generator is forced to learn the correlation between k_{SGS} and the smoothness level of X to avoid retraining whenever the level of smoothing changes.
To increase the training stability, a conditional discriminator, introduced in [7], is employed. In this method both LR and HR data are provided to the discriminator network. The LR data is up sampled through nearest neighbors' interpolation to align with the size of HR data and then concatenated with it before being fed to the discriminator. By conditioning the discriminator on the LR input, we ensure that the discriminator incorporates LR data into its assessment. We extend this conditional framework by modifying the objective function based on the suggestion from [8]:

$$\min_G \max_D \Big\{E_{Y_r \sim P_{Y_r}}\left[\log D_{R_a}\left(Y_r, Y_g|X\right)\right] + E_{Y_g \sim P_{Y_g}}\left[\log\left(1 - D_{R_a}\left(Y_g, Y_r|X\right)\right)\right] + E_{Y'_g \sim P_{Y_g}}\left[\log\left(1 - D_{R_a}\left(Y'_r, Y_g|X\right)\right)\right]\Big\}, \quad (1)$$

where Y'_r is a shuffled version of Y_r, intended to mismatch with the X and $\mathrm{D_{R_a}}$ is the conditional relativistic discriminator [1]. The last term of Equation 1 forces discriminator to give a low score to the mismatched inputs. Incorporating this term additionally facilitates the alignment of adversarial loss with pixel-wise content loss, leading to a more stable training. The loss function is composed of the Discriminator Loss:

$$\mathcal{L}_D = -E_{Y_r \sim P_{Y_r}}\left[log\left(D_{Ra}(Y_r, Y_g|X)\right)\right] - E_{Y_g \sim P_{Y_g}}\left[log\left(1 - D_{Ra}(Y_g, Y_r|X)\right)\right] - E_{Y'_r \sim P_{Y_r}}\left[log\left(1 - D_{Ra}(Y'_r, Y_g|X)\right)\right], \quad (2)$$

and the Generator Loss:

$$\mathcal{L}_G = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{vel}\mathcal{L}_{vel} + \lambda_{grad}\mathcal{L}_{grad} + \lambda_{phys}\mathcal{L}_{phys}, \quad (3)$$

where $\mathcal{L}_{adv} = -E_{Y_g \sim P_{Y_g}}\left[log\left(D_{Ra}(Y_g, Y_r|X)\right)\right]$ is the adversarial loss derived from Equation 1, $\mathcal{L}_{vel} = \frac{1}{N}\sum_{i=1}^{N}\left|Y_g^i - Y_r^i\right|_2^2$ is the velocity field pixel-wise loss, $\mathcal{L}_{grad} = \frac{1}{N}\sum_{i=1}^{N}\left|\nabla Y_g^i - \nabla Y_r^i\right|_2^2$ is the velocity gradient field pixel-wise loss, and $\mathcal{L}_{phys} = \frac{1}{N}\sum_{i=1}^{N}\left|\nabla \cdot Y_g^i\right|_2^2$ is the physical consistency loss based on the divergence-free condition for incompressible flows, $\lambda_{adv} = 10^{-4}$, $\lambda_{vel} = 0.89$, $\lambda_{grad} = 0.085$, and $\lambda_{phys} = 0.025$.

3. Data preparation and training

A pair of DNS and filtered DNS (FDNS) data serves as the training dataset for the networks in this study. The three-dimensional velocity field of a forced, homogeneous, isotropic turbulent flow is chosen as the basis for training and testing. This particular case has been extensively employed both for assessing super-resolution methods and for investigating the particle-turbulence interactions, rendering it well-suited for this study. The flow field data are generated using a DNS simulation implemented in OpenFOAM. The computational domain is a periodic cubic box, discretized using Cartesian uniform grids with a resolution of 64^3 and the cell size of $\Delta = 2\pi/64$. The Uhlenbeck-Ornstein (UO) random process-based forcing term is utilized to sustain statistically stationary turbulence within the computational domain. This is achieved by continuously injecting kinetic energy solely into the low-wavenumber modes in the Fourier space. The resulting sustainable turbulence has the following statistical parameters: Kolmogorov length scale $\eta = 0.0646\,m$, Kolmogorov time scale $\tau_\eta = 208\,s$, integral length scale $l = 2.415\,m$, large vortex turnover time $t_l = 2.925\,s$, Taylor microscale $\lambda = 0.704\,m$, Taylor Reynolds number $Re_\lambda = 36.457$, turbulence kinetic energy TKE $= 1.138\,m^2/s^2$ and the dissipation rate $\varepsilon = 0.458\,m^2/s^3$. To prepare the LR data for the super-resolution training, we apply a Gaussian filter with a standard deviation of σ to the original DNS data. The filtered field is then downsampled using a stride of 4, reducing the resolution from 64^3 to 16^3. This two-step approach, filtering followed by downsampling, separates the smoothing process from the resolution reduction allowing for more flexibility in generating training data with various levels of detail loss. After filtering and downsampling, we calculate the k_{SGS} for each filtered sample using the difference between the DNS velocity field and the filtered DNS (FDNS) velocity field. This k_{SGS} value serves as the conditioning information for the conditional generator, providing information about the level of detail lost during the filtering process. A possible application of the CESRGAN is a LES with an explicit equation for k_{SGS}. The filtering and reconstruction are illustrated in Figure 2.

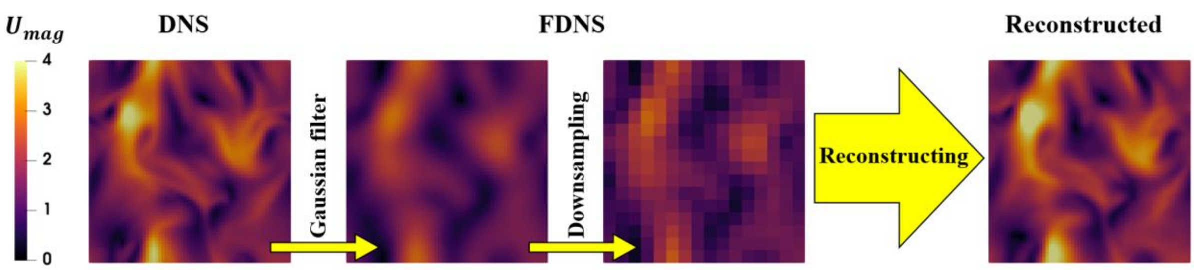

Figure 2. Visualization of the filtering and reconstruction procedure. Left: DNS velocity magnitude field. Center: FDNS field with $\sigma = 4\Delta$, Right: Reconstructed HR field from the FDNS input.

We used 6000 snapshots from the statistically stationary state of the simulation, with a time interval of $2t_l$ between consecutive snapshots for training. To prevent the overfitting, random rotations and reflections to the velocity field as a data augmentation technique were applied. Prior to feeding the data into the networks, we normalize each FDNS-DNS pair using the root mean square (RMS) of the FDNS snapshot's velocity fluctuations. The deep neural networks were implemented using PyTorch. We utilized the Adam optimizer with a fixed learning rate of 10^{-4} and a batch size of 40. Unlike previous studies, our model does not need adaptive learning rates or pre-training phases to achieve stable GAN training. This stability is achieved through the use of our conditional discriminator, which aligns the adversarial loss with the pixel-wise content loss. This conditional approach decreases training instabilities commonly encountered in GAN-based super-resolution models. The total training time was approximately 8 hours on one node of the HoreKa supercomputer at the Karlsruhe Institute of Technology (KIT), equipped with four NVIDIA A100 GPUs. To evaluate the trained generator, we integrate it with OpenFOAM using the C++ API of PyTorch. The generator uses the FDNS field to reconstruct a super-resolution (SR) velocity field on-the-fly. Table 1 details the testing configurations and naming conventions used in our study.

Table 1. SR testing configurations and naming conventions.

Filter	Description	Data type	Name
σ=2Δ	Moderate filtering	Filtered DNS (FDNS)	FDNS-2
	(typical coarse simulation)	TradGEN* output	TradGEN-2
		CoGEN output	CoGEN-2
σ=4Δ	Strong filtering	Filtered DNS (FDNS)	FDNS-4
	(challenging reconstruction)	TradGEN output	TradGEN-4
		CoGEN output	CoGEN-4

* *TradGen: traditional generator*

3. Results

Figure 3 visualizes vortex structures of reconstructed fields using iso-surfaces of the Q-criterion ($Q = 0.86\tau_\eta^2$). Color-coded results distinguish between three categories: yellow (vortex structures matching DNS ground truth), blue (structures omitted by reconstruction models), and green (model-generated structures deviating from DNS). Both CoGEN and TradGEN accurately reconstruct large-scale vortices across all smoothness levels of the filtered DNS (FDNS) input. However, CoGEN outperforms TradGEN in small-scale structure recovery. First, the conditional generator demonstrates superior performance in fine structure recovery. CoGEN was able to recover very fine structures, whereas the TradGEN missed several of these fine structures, which are shown in blue. This discrepancy in the small-scale structures is a consequence of the ill-posed nature of the super-resolution reconstruction process, where multiple high-resolution reconstructions can exist for a given low-resolution input. In the CoGEN, this ill-posedness is alleviated by using k_{SGS} as additional information, constraining the range of plausible solutions. Second, the CoGEN shows an ability to compensate for omitted structures by generating new vortical structures, shown in green in Figure 3.

To assess the fidelity of the reconstructed velocity fields for turbulent particle dispersion we injected randomly 4096 pairs of particles with an initial separation distance of 0.5η and tracked them in time. The particle Stokes numbers were St_k = 0.125, 1 and 8. During the simulation, the ensemble averaged distance between paired particles, denoted as $\langle\delta\rangle$, was calculated for each time step. Results of the simulation are presented in Figure 4. It is evident that the FDNS significantly underpredicts particle dispersion across all cases. For the moderate filtering σ=2Δ, both the conditional and traditional generators perform remarkably well, closely matching the DNS dispersion curves for all Stokes numbers; therefore, their results are not presented. The advantages of the conditional generator become apparent only with the stronger filter σ=4Δ. In this case, the CoGEN maintains excellent agreement with DNS across all Stokes numbers, while the TradGEN shows noticeable deviations for smaller Stokes numbers. The success in reproducing particle dispersion further validates the physical fidelity of the reconstructed fields, as particle motion integrates the effects of multi-scale turbulent structures over time.

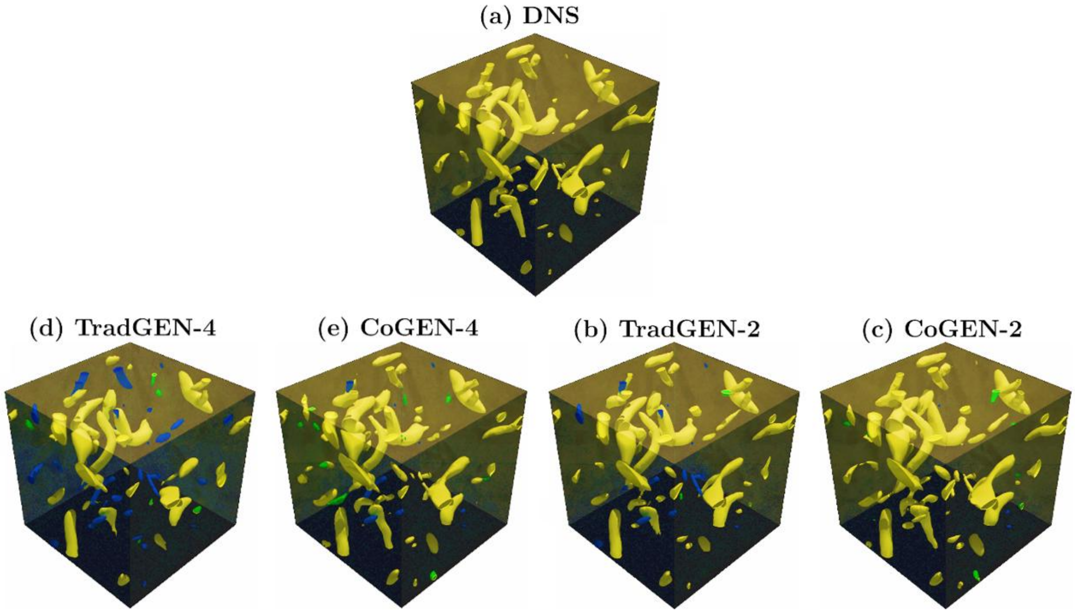

Figure 3. Q-criterion isosurfaces from (a) DNS, (b) TradGEN-2, (c) CoGEN-2, (d) TradGEN-4, and (e) CoGEN-4. Yellow: accurate DNS reconstructions; blue: structures missed in super-resolution (SR) outputs; green: model-generated features deviating from DNS.

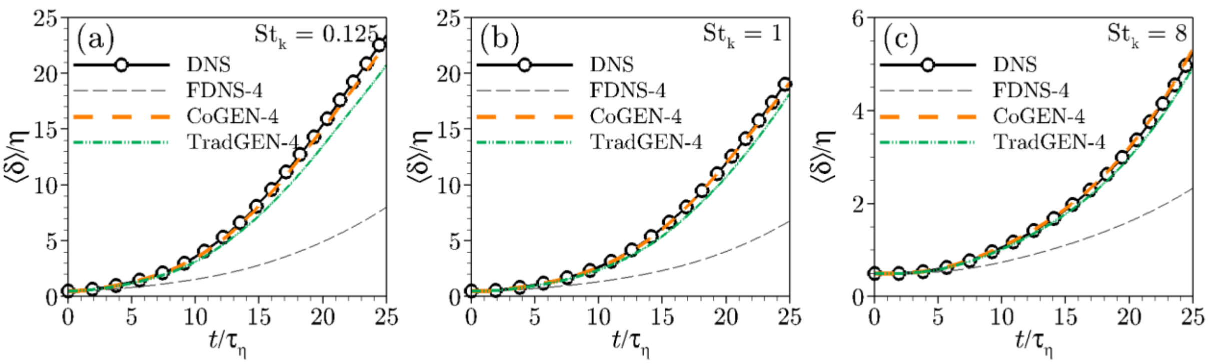

Figure 4. Evolution of particle pair dispersion. Pairs initially separated by 0.5η.

4. Conclusion

The paper presents a novel deep learning model for reconstruction of high-resolution turbulent velocity fields from low-resolution inputs. The model called CESRGAN consists of a conditional discriminator and a conditional generator. By incorporating subgrid-scale turbulence kinetic energy as an additional condition for reconstruction, this model significantly improves the recovery of small-scale turbulence structures compared to traditional super-resolution methods. The advantages of new model are illustrated for the reconstruction of fine-scales turbulent structures in the box turbulence. The model applied to particle dispersion simulations closely matches DNS across various Stokes numbers and filtering levels, highlighting the potential application of super-resolution models in particle-laden flows.

Acknowledgments

This work was performed on the HoreKa supercomputer with support from the National High-Performance Computing Center project (NHR@KIT), both funded by the Ministry of Science, Research and the Arts Baden-Württemberg and the German Federal Ministry of Education and Research.

References

[1] H. Tofighian, J. Denev, N. Kornev (2024) 'A Conditional Deep Learning Model for Super-Resolution Reconstruction of Small-Scale Turbulent Structures in Particle-Laden Flows', Physics of Fluids, https://doi.org/10.1063/5.0235192.
[2] G. Park, J. Urzay, M. Bassenne P. Moin (2017) 'A simple dynamic subgrid-scale model for LES of particle-laden turbulence', Physical Review Fluids, 2, 044301-1-044301-20.
[3] S. Idelsohn, J. Gimenez, A. Larreteguy, N.Nigro, F. Sivori, E. Oñate (2024), The P-DNS Method for Turbulent Fluid Flows: An Overview, Arch. Computat. Methods Eng., 31, 973—1021, DOI=10.1007/s11831-023-10004-3.
[4] N. Kornev, S. Samarbakhsh, J. Darji, (2024) 'Hybrid grid free and grid based method for simulation of turbulent flows', Physics of Fluids, https://doi.org/10.1063/5.0235192.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio (2020) 'Generative adversarial networks', Commun. ACM 804 63(11), 139–144
[6] Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et al. (2017) 'Photo-realistic single image super resolution using a generative adversarial network', Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4681–4690
[7] X. Zhang, H. Song, K. Zhang, J. Qiao, Q. Liu (2020) 'Single image super-resolution with enhanced Laplacian pyramid network via conditional generative adversarial learning', Neurocomputing, 398, 531—538.
[8] H. Yin (2023) 'Comparison and analysis of various deep learning models for super-resolution reconstruction of turbulent flows', Journal of Physics: Conference Series, 2634(1).