

Variational Autoencoder for Turbulence Generation

Kevin Grogan
Stanford University
450 Serra Mall, Stanford, CA 94305
krogan@stanford.edu

Abstract

A three-dimensional convolutional variational autoencoder is developed for the random generation of turbulence data. The variational autoencoder is trained on a well-resolved simulated database of homogeneous isotropic turbulence. The variational autoencoder is found to be sufficient in reconstructing a non-trivial turbulent vector field. Additionally, a physical metric of the reconstructed velocity field showed improvement during the training process without explicit enforcement. It is concluded that the variational autoencoder framework shows promise towards a reduced-order model of turbulence generation.

1. Introduction

Turbulence is the chaotic motion of fluids and is one of the outstanding unsolved problems of classical physics. The great early 20th-century fluid mechanician Horace Lamb once purportedly said:

"I am an old man now, and when I die and go to heaven there are two matters on which I hope for enlightenment. One is quantum electrodynamics, and the other is the turbulent motion of fluids. And about the former I am rather optimistic."

The primary model of fluid mechanics is the Navier-Stokes equations:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\nabla \left(\frac{p}{\rho} \right) + \nu \nabla^2 u, \quad (1a)$$

$$\nabla \cdot u = 0, \quad (1b)$$

where u is the velocity vector, p is the pressure, ρ is the density, and ν is the kinematic viscosity. Equation 1a ensures momentum conservation while Eq. 1b ensures mass conservation. In incompressible flows (e.g., flows of liquids or low-speed gases), the density is constant and is included in the pressure as is done in Eq. 1.

The Navier-Stokes equations are so infamously intractable that the proof of their uniqueness and smoothness would yield a Millennium Prize valued at \$1 million.

More so, the production of turbulence data requires computationally expensive numerical solutions to the Navier-Stokes equations, which often takes hundreds of thousands of CPU-hours to generate. Hence, an efficacious reduced-order model of a turbulent flow-field could significantly reduce the computational cost of turbulence simulations.

Therefore, it is the objective of this project to bypass the computationally intensive process of producing turbulence data via a discretization of Eq. 1 and to create a reduced-order model capable of producing turbulence data. Towards this purpose, a variational autoencoder [5] is used as the generative model for producing random turbulence data.

Possible applications of an accurate reduced-order model of turbulence could be useful defining turbulent inlet conditions for high-fidelity simulations, Monte-Carlo ray tracing for radiation problems, fluid-structure interaction modeling, and turbulence closure modeling.

2. Background/Related Work

The initial conditions and boundary conditions of a turbulent simulation are often consequential in the production of turbulent flow field. Often, the velocity at the inlet of the simulation is crudely modeled with one-dimensional profile with random fluctuations superimposed, and the pressure is taken to be constant. However, with this method one can find quantities of interest poorly-predicted [7].

An auxiliary simulation is sometimes employed as a higher-fidelity method to generate realistic turbulence [11]. However, while the statistics of the turbulent flow are well-replicated, this method comes with a significant computational cost.

Synthetic turbulence may be randomly generated from a known or a model kinetic energy spectrum [8]. However, ensuring that the resultant velocity field is real-valued in three-dimensions is non-trivial. Additionally, this method is insufficient at generated statistics beyond second order, and care must be taken to generate realistic phase angles from a model spectrum. However, improvements have been found via the method of digital filtering [6].

Hence, this work introduces a three-dimensional convo-

lutional variational autoencoder as a novel method to randomly generate three dimensional turbulent fields.

Recent advances in deep learning algorithms have yielded promising results in the field of scientific computation. Oliveira *et al.* [12] utilized a generative adversarial neural network [3] for the production of two-dimensional jet images of energy deposits. Furthermore, Tompson *et al* [14] utilized convolutional neural networks to accelerate the computation of fluids simulations for computer graphics applications. Additionally deep neural networks were employed by Ling *et al.* [10] to predict the components of the turbulence anisotropy tensor.

3. Method

A variational autoencoder [5] is employed in this work to randomly generate turbulence data. The variational autoencoder consists of two connected neural networks: an encoder, which transforms the input into latent variables (e.g., the means and standard deviations of the latent random vector, z); and a decoder, which produces a reconstruction of the example from the latent random vector. The application of this model will randomly sample the latent variables to produce turbulence data.

Turbulence is inherently three-dimensional due multi-dimensional instability mechanisms. Hence, the architecture utilizes three-dimensional convolution neural networks for the encoder and decoder. However, turbulent statistics of the data set are stationary, and therefore, the dynamics of the turbulence are not incorporated in this project. Additionally, it is postulated that the incorporation of spatial locality information would mimic the dynamics of turbulent eddies and would produce a superior model of the turbulent flow than those tried previously.

A depiction of the three-dimensional convolutional neural network used for this work is shown in Fig. 1. The convolutional neural network is trained on the homogenous, isotropic turbulence database available from the Johns Hopkins Turbulence Database [9]. Figure 2 shows a snapshot of the database¹ for reference. The data set consists of a time series of $1024 \times 1024 \times 1024$ four-component spatial data (i.e., $[u_x, u_y, u_z, p/\rho]$). This data was generated via a discretization of Eq. 1. The statistics of homogeneous, isotropic turbulence are by definition invariant with spatial location, and direction, respectively.

The encoder network consists of three 3D strided convolution/batch normalization/leaky ReLU layers to reduced the dimensionality of the inputted example. A $5 \times 5 \times 5$ kernel is used for the convolution in all layers with SAME padding. Strides of two are used in all directions to reduce the total dimensionality of the input by a factor of 8. Additionally, the number of filters were reduced per layer by

¹Taken from <http://turbulence.pha.jhu.edu/images/isotropic.jpg>

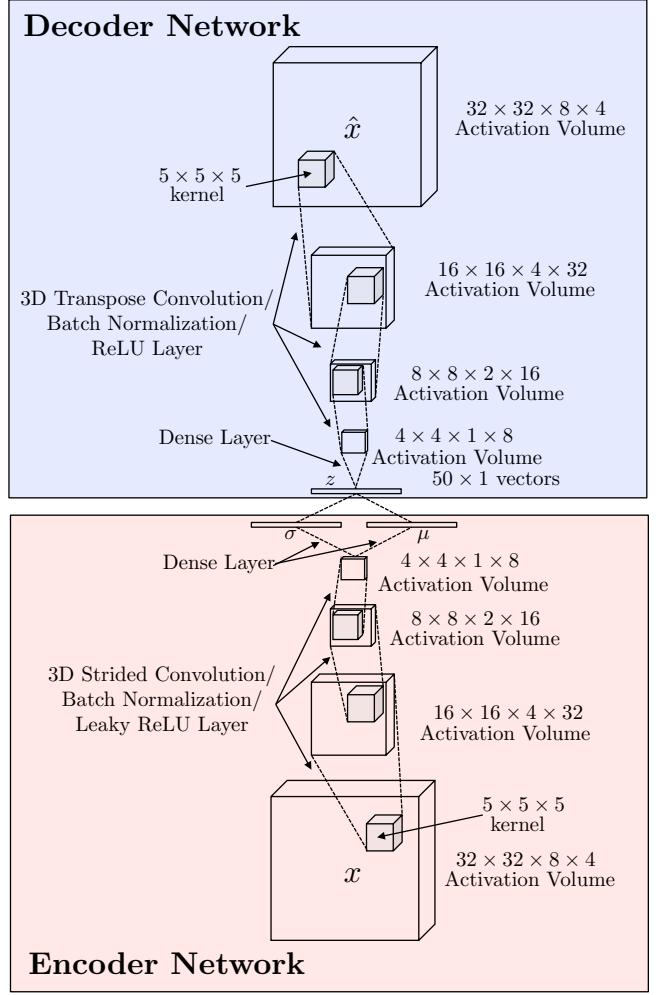


Figure 1. Schematic of the three-dimensional varaiational autoencoder.

a factor of two since experiments were shown to produce a lower loss in this configuration.

The dimensions of the inputted sample are $32 \times 32 \times 8 \times 4$. The z -direction was chosen to have a reduced dimensionality to reduce the computational cost of training; this reduction likely has a negligible effect on the description of the statistics of the turbulence since the field is isotropic.

A dense layer is utilized to to reduce the final activation volume into a vector means, μ , and standard deviations, σ , which parameterize the multivariate Gaussian distribution of the random vector $z \sim \mathcal{N}(\mu, \sigma)$. A dense layer yields greater flexibility over the dimensionality of z without significant alterations of the architecture. No outstanding improvement was found for $\text{dim}(z)$ approximately greater than 50; this is attributed to the dimensionality of the prior reduction.

The decoder network mirrors the encoder network. While the relative activation volumes are congruent, 3D

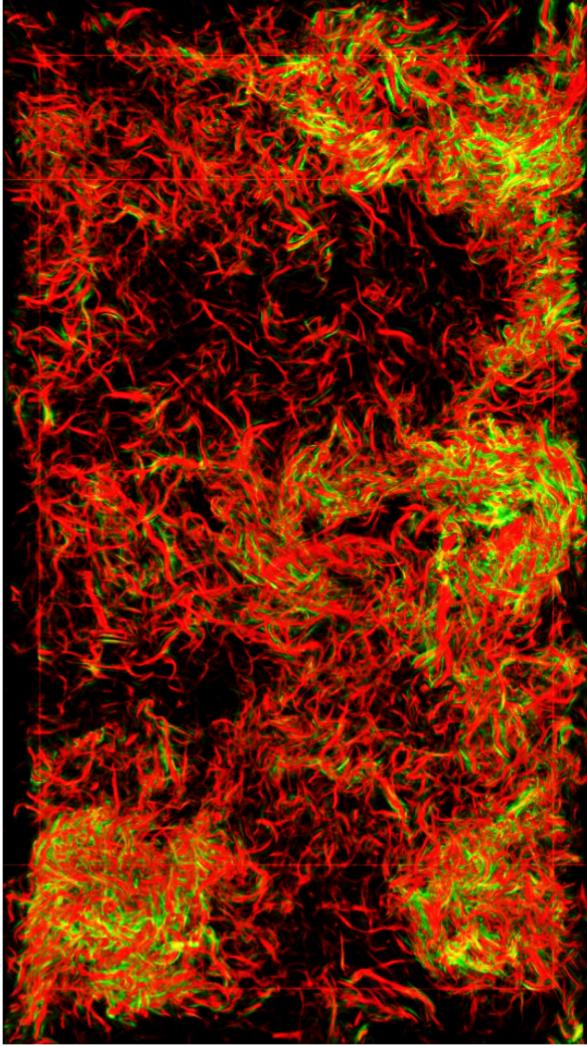


Figure 2. Planar cut of the vorticity of the flow in the database. Vorticity is defined as the curl of the velocity field, $\omega = \nabla \times u$ and is representative of the rotation of the fluid.

transpose convolutions are utilized to increase the dimensionality between layers.

An L_2 regularization is applied to the weights of the model and the weights are initialized from a standard normal distribution. Xavier initialization [2] was not observed to yield a significant improvement in the stability of the network.

The cost function for the variational autoencoder is based on a lower bound for log-likelihood function:

$$\mathcal{L}_i = \mathbb{E}_z [\log p_\theta(x^{(i)}|z)] - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z)) \quad (2)$$

where the index i corresponds to a particular example, $p_\theta(x^{(i)}|z)$ is the distribution parameterized by the output of

the probabilistic decoder, $q_\phi(z|x^{(i)})$ is the distribution parameterized by the output of the probabilistic encoder network, D_{KL} is Kullback-Leibler divergence, and $p_\theta(z)$ is the prior distribution for the latent vector z . This lower bound is optimized across all training examples, $x^{(i)}$, and the cost function is simply the negative of the log-likelihood function.

In this work, the prior distribution is a centered, unit Gaussian, and the posterior approximation, q_ϕ , is Gaussian with μ and σ produced by the encoder network as shown in Fig. 1. The distribution of the decoder network is taken to be Bernoulli; for this, the inputted example is scaled and centered using an affine transformation to be between zero and one. The use of a Gaussian distribution for the decoder network was investigated; however, non-trivial reconstructions of the input vector field could not be obtained. Additionally, the expectation in Eq. 2 is replaced by the first order approximation of using the mean vector μ ; this was done to avoid the computational expense of a Monte Carlo estimate.

Tensorflow [1] is the API employed to create the neural networks. 20 epochs are used for a training set of 10,000 examples. For the current data set, a maximum of 131,072 samples are available for $32 \times 32 \times 8 \times 4$ cuts in the data. A batch size of 100 is used during training and the ADAM [4] optimizer is employed.

The code for the variational autoencoder is adapted from Metzen².

4. Experiments

The results of the training process are shown in Fig. 3. As shown in the figure, both the training cost and validation costs decrease through the experiment. There is some separation apparent between the two costs; however, it was found that regularization did help ameliorate this disparity. Additionally, it was found that the reconstruction term in Eq. [?] dominated the value of the loss by several orders of magnitude.

Additionally, the residual of the flow divergence is computed in the following manner:

$$\epsilon = \frac{1}{N} \sum_{i=1}^N \sum_{h=2}^{H-1} \sum_{w=2}^{W-1} \sum_{d=2}^{H-1} \left[(\nabla \cdot \hat{u})_{h,w,d}^{(i)} \right]^2 \quad (3)$$

where the divergence is computed via a centered, second-order finite difference. According to Eq. 1b, this quantity should be exactly zero in an incompressible flow field. However, the divergence residual increases greatly at the outset of training. This is due to the initial flow field being quite random and nearly divergence free on average. Interestingly, the model shows that the divergence does decrease

²<https://jmetzen.github.io/2015-11-27/vae.html>

after the initial increase; this indicates that the turbulence is, in fact, moving towards a realizable flow field; that is the autoencoder is learning to become more physically realizable. Additional experiments were run by an ad-hoc inclusion of the divergence residual in the cost computation; this is in effect similar adding the total variation to the cost function as is done with images. However, it was found that this addition promoted trivial results.

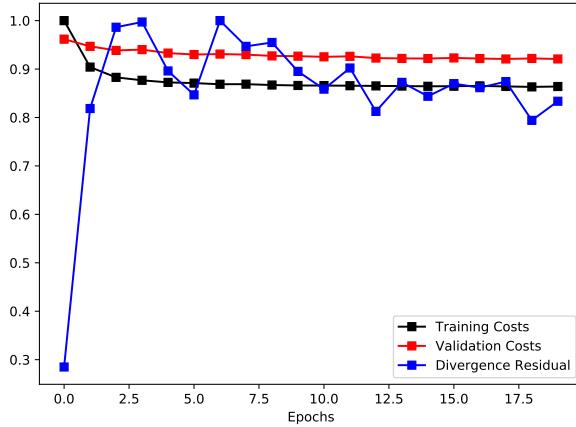


Figure 3. Comparison of the losses and flow divergence. All quantities are scaled by their maximum value.

The results of the fully-connected variational autoencoder applied to the 3D turbulent kinetic energy data is shown in Fig. 4. The turbulent kinetic energy is defined as

$$k = \frac{u^2 + v^2 + w^2}{2} \quad (4)$$

and is an important quantity for describing the energetics of the turbulent flow [13]. The reconstructions are shown for a test data set. As shown in the figure, the framework of the variational autoencoder is performing well for capturing the structure of the turbulent kinetic energy data with respect to an $\|I_{\text{ball}}\|$. In particular, one notices the relatively sharp features in the comparison on the second row. However, one does observe some blurriness in the reconstruction, which is characteristic of variational autoencoders. Additionally, smoother, finger-like structures are shown for the inputs while the reconstructions can be seen to be somewhat noisy.

Turbulence data generated by the convolutional variational autoencoder are shown in Fig. 5. Since turbulent flows have a chaotic structure, the correctness of these results cannot necessarily be ascertained directly from their appearance; however, one does notice that the generated samples show qualities in line with the reconstructed samples. That is, one can expect that the quality of the generated samples will be in line with the reconstructed samples.

The current results are encouraging, and additional quantitative metrics can be employed in future studies. In partic-

ular, by taking the divergence of Eq. 1a and applying Eq. 1a one obtains the pressure Poisson:

$$\nabla^2 \left(\frac{p}{\rho} \right) = -\nabla \cdot (u \cdot \nabla u). \quad (5)$$

This equation can be employed to investigate the generated turbulence as a metric to determine the realizability of the flow by applying a discretized version to the generated turbulence and aggregating the residual in a manner similar to the flow divergence; however, since the data is scaled in the current work, this equation is not directly applicable. Furthermore, additionally known properties of the generated turbulence may be inspected including the rotational invariance of the turbulent stress tensor and the kinetic energy spectrum [13]. Finally, the application of the randomly generated data to a practical simulation would be the final evaluation of the quality of the model.

5. Conclusion

The 3D convolutional variational autoencoder framework has been demonstrated to generate turbulent features reasonably well. It was found that a physical metric of the realizability of the turbulent flow improved through the training process without direct enforcement. Hence, it is concluded that generative models have potential in the useful production of meaningful turbulence data.

References

- [1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
- [2] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.
- [3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- [4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [5] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.
- [6] M. Klein, A. Sadiki, and J. Janicka. A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations. *J. Comp. Phys.*, 186:652–665.

- [7] M. Klein, A. Sadiki, and J. Janicka. Direct numerical simulations of plane turbulent jets at moderate reynolds numbers. *20th IUTAM Congress*, 2000.
- [8] S. Lee, S. Lele, and P. Moin. Simulation of spatially evolving compressible turbulence and the application of taylors hypothesis. *Phys. Fluids A*, 4:1521–1530.
- [9] Y. Li, E. Perlman, M. Wan, Y. Yang, C. Meneveau, R. Burns, S. Chen, A. Szalay, and G. Eyink. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. *J. Turbul.*, 9(31):1–29, 2008.
- [10] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.*, 807:155–166.
- [11] T. Lund, X. Wu, and D. Squires. Generation of turbulent inflow data for spatially-developing boundary layer simulations. *J. Comp. Phys.*, 140:233–258.
- [12] L. Oliveira, M. Paganini, and B. Nachman. Learning particle physics by example: Location-aware generative adversarial networks for physics synthesis. *arXiv:1701.05927v1*, 2017.
- [13] S. B. Pope. *Turbulent Flows*. Cambridge University Press, Cambridge, 2000.
- [14] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin. Accelerating eulerian fluid simulation with convolutional networks. *arXiv:1607.03597v5*, 2017.

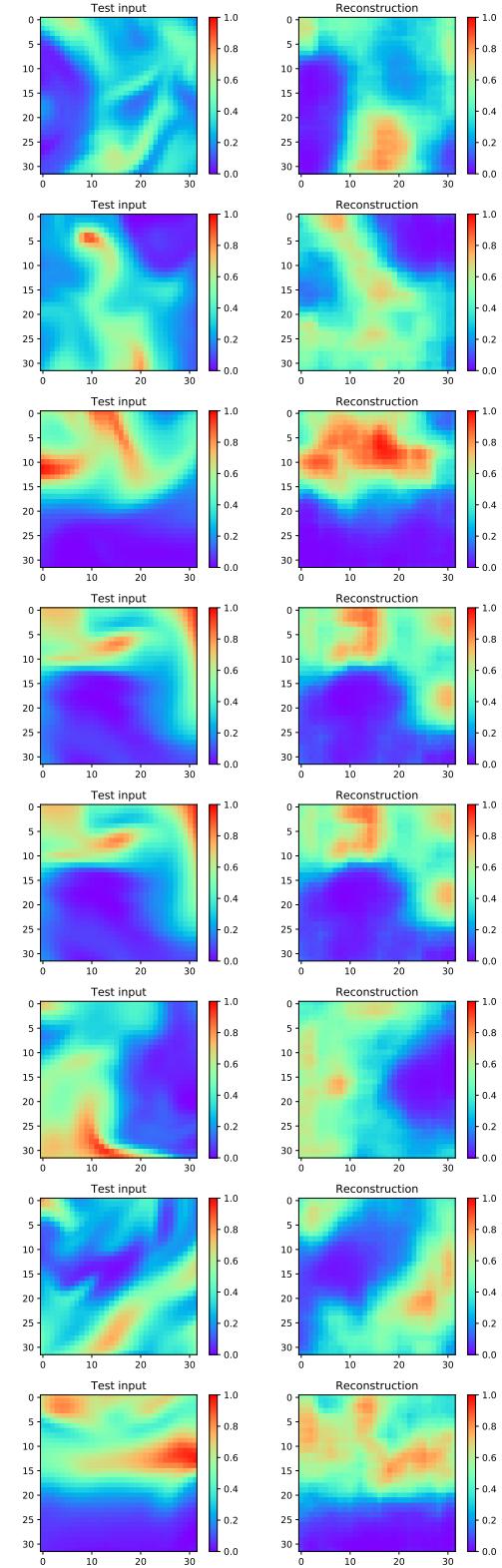


Figure 4. Comparison of the inputted turbulent kinetic energy to the reconstructed flowfield.

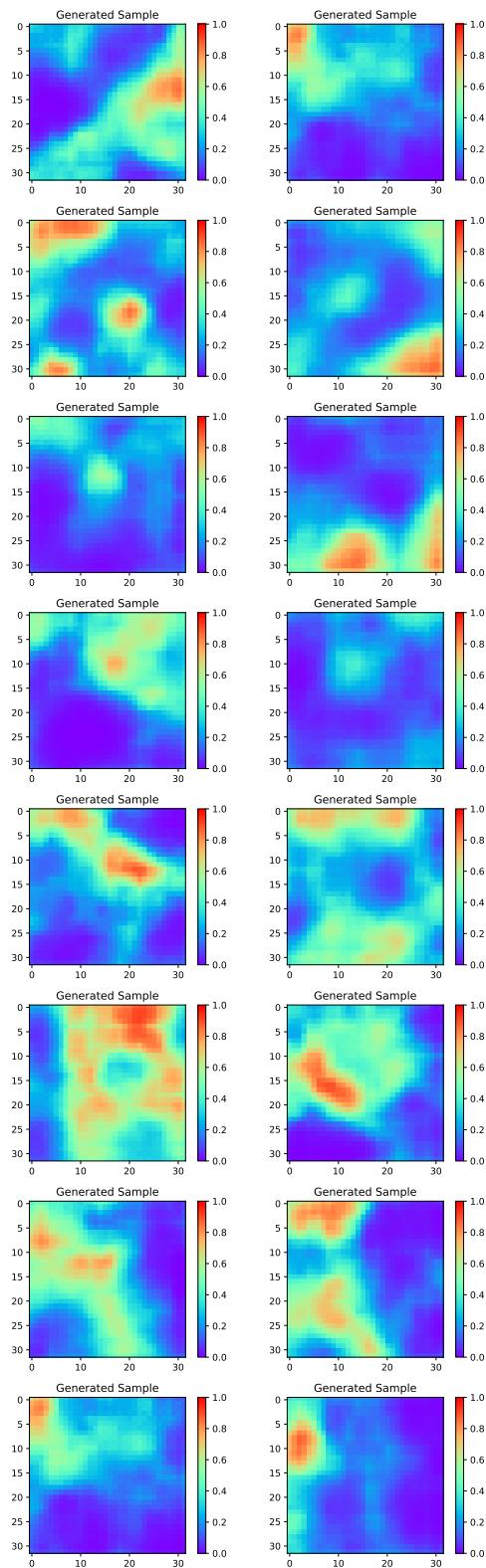


Figure 5. Examples of the generated turbulent data