Learning non-linear spatio-temporal dynamics with convolutional Neural ODEs

Varun Shankar
Carnegie Mellon University
varunshankar@cmu.edu

Gavin D. Portwood
Los Alamos National Laboratory
portwood@lanl.gov

Arvind T. Mohan
Los Alamos National Laboratory
arvindm@lanl.gov

Peetak P. Mitra
University of Massachusetts
pmitra@umass.edu

Christopher Rackauckas
Massachusetts Institute of Technology
crackauc@mit.edu

Lucas A. Wilson
Dell Technologies
luke.wilson@dell.com

David P. Schmidt
University of Massachusetts
schmidt@acad.umass.edu

Venkatasubramanian Viswanathan
Carnegie Mellon University
venkvis@cmu.edu

Abstract

Advances in both scientific computing and machine learning have led to techniques to improve, replace, or speed up methods to solve problems in the physical sciences. For many nonlinear systems, traditional methods of calculation often carry a heavy computational burden due to complex dynamics, physical constraints, or multi-scale behavior. We introduce a machine learning approach to vastly decrease computational complexity of modeling spatiotemporal systems, leveraging prior physical knowledge of the system with flexible reduced representations of the underlying dynamics. We demonstrate this technique on three-dimensional turbulent fluid flow, wherein the modeling of such systems continues to be a formidable problem for engineers and physicists. We merge convolutional layers with a Neural ODE approach to form a differentiable ML architecture to build a system model. Here, we test the applicability of this method to model homogeneous isotropic turbulence, as well as the effect of different architectural elements on turbulence statistics. We find that the largest scales of the flow, which are relevant to engineering applications, are well-approximated and some dynamic characteristics of interest are maintained.

1 Introduction

The reproduction of spatio-temporal dynamics by physical models, often governed by partial differential equations, is critical for a variety of engineering applications. Despite many decades of progress with scientific computing and numerical methods, solution accuracy has to be sacrificed to model realistic systems. Turbulent fluid flow, which governs many practical engineering applications, from earth sciences to biology, is an example of this trade-off. The challenge lies in solving the governing Navier-Stokes equations to an appropriate degree of accuracy with computational fluid dynamics (CFD) techniques and current computational capabilities. Among several strategies in CFD, Direct Numerical Simulation (DNS) of the equations aims to fully resolve all scales of the flow, thus leading to extremely high-fidelity solutions. However, computational complexity of DNS limits its application to simple flow configurations which are typically oversimplified for general engineering

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

or geophysical applications. For most practical engineering problems, resolving only a subset of the larger scales is necessary to gain meaningful insight. To this end, numerous models informed by known physical laws have been developed to model turbulent interactions while accurately resolving only large scales of interest.

While CFD techniques rely on known physics, empirical fitting, and mathematical derivations, data-driven techniques offer an alternative approach with enhanced flexibility and richer model representations. Despite being a nascent area of research, ML in CFD has seen broad interest in the community. Approaches have focused on integrating neural networks into existing CFD algorithms [1, 2, 3, 4], improving predictive capacity [5] and generalization by modeling intermediate quantities used in flow simulations [6]. Alternate approaches include the task of whole field prediction, in 2D [7, 8] and 3D [9], using partial observations [10], and with physics-informed deep learning architectures [11, 12]. These data-driven models have shown significant promise with their ability to improve or efficiently replace CFD methods. In parallel, advancements in deep-learning algorithms have opened new doors for their applicability towards scientific research. Chen et al. popularized the use of Neural ODEs (NODEs), which permit differential equations to be easily integrated within a deep-learning framework [13]. The ability to adapt neural networks to include differential equations has far-reaching implications for modeling physical phenomena.

1.1 Our Contributions

Inspired by the underlying physics, we investigate the use of NODEs to model spatiotemporal dynamics. While smaller NODEs have been used successfully in various implementations such as reduced-order models or other parameterized differential equations [14], we explore modeling whole field dynamics with this framework. In this work, we develop a data-driven, reduced order model to forecast three-dimensional turbulence and show the effect of kernel size and compression ratios on the predictive ability of the latent space model. As we are interested in fluid dynamics applications, we take inspiration from existing CFD techniques in constructing our model architecture and include components to force physical solution (e.g. mass balance constraint) outputs.

2 Physical description of the problem

To evaluate our approach with regards to fluid dynamics, we consider an idealized case of turbulent flow – homogeneous isotropic turbulence. The convolutional neural ODE method stems from the fact that we wish to model the temporal dynamics of the system. For an incompressible fluid, there are two governing equations, a momentum balance and a mass balance, shown here in their non-dimensional form:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{1}{Re}\nabla^2 \mathbf{u} - \nabla p + \mathbf{f}; \quad \nabla \cdot \mathbf{u} = 0 \tag{1}$$

where $\mathbf{u}$ is the velocity vector, defined $\{u, v, w\}$, p is the pressure, f is a forcing function, and Re is a Reynolds number, a non-dimensional flow parameter which broadly describes the intensity of inertial forces over viscous ones. We can see from the momentum balance that the dynamics of the velocity field are both autonomous and a function of local spatial gradient information of the field. Given a spatially discrete field, then, we can construct an ODE system to describe the dynamics using a neural network. We note that making this neural network a convolutional neural network will incorporate the spatial information (derivatives) necessary to accurately approximate the true function. We construct the problem such that the model is provided with an initial snapshot of the flow. The continuous dynamics are forecasted with the model, and a series of temporal snapshots within a specified time window are saved and analyzed with regards to the ground truth data. We compare results using a series of three important turbulent statistics – energy spectra, velocity probability density functions (PDFs), and turbulent kinetic energy (TKE). The turbulent kinetic energy is defined as one half of the sum of the mean-square fluctuations of the velocity components:

$$e = \frac{1}{2}(\langle u^2 \rangle + \langle v^2 \rangle + \langle w^2 \rangle) \quad . \tag{2}$$

3 Machine Learning approach

3.1 ConvAE + NeuralODEs

Due to the combined multi-scale and non-linear behaviour of the Navier-Stokes equations, discretized numerical solutions to (1) are subject to unfavorable scaling of computational complexity with

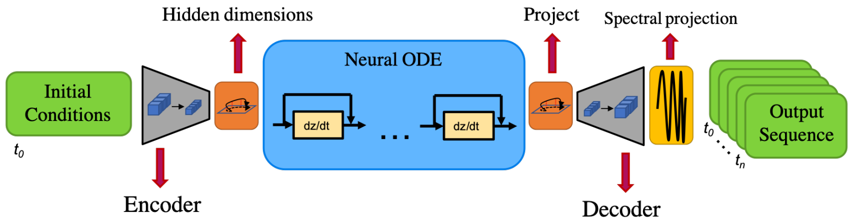

Figure 1: Schematic of the overall model architecture. Initial conditions are given and encoded into a latent space. Augmented channels are concatenated and the dynamics are forecasted through the neural ODE approximator. The sequence is decoded and the divergence-free condition is enforced through the last layer.

respect to the Reynolds number, where computational complexity can be shown to scale with $Re^{9/4}$ [15]. To improve computation tractability, much work has been done in reduced order modeling of dynamical systems. Proper orthogonal decomposition, often used with fluid flows, takes a statistical approach, projecting the data onto a linear subspace comprised of dominant modes [16]. The idea of a latent space representation is common in many fields, including computer vision. Convolutional autoencoders (ConvAE) have been used for tasks like image compression or feature extraction, both of which closely parallel our goal of a rich reduced representation of our turbulent flow field. We take advantage of this technique to reduce our snapshots into a much smaller latent space. The dynamics of the system can then be performed on the latent representation, easing computation. Using a latent space in conjunction with Neural ODEs is not new and has been used successfully for modeling many other dynamical systems [17].

The design of our ConvAE + Neural ODE system is sketched in fig. 1. Here, the latent space dynamics $\frac{d\mathbf{z}}{dt}$ are approximated using a 4 layer convolutional network with dropout prior to the last layer. We investigated multiple kernel sizes and settled on 7 for the first layer and 5 for the other three based on balancing computational training cost with results quality. The computation time scales roughly equivalently with the number of parameters in the Neural ODE, which itself scales as k^3, where k is the kernel size. The encoder is a 2 layer convolutional network, with 1 layer of stride 2 to downsample the output. The decoder shares this architecture with convolutional transpose layers. We employ a compression ratio of 6, which is the size of the original data relative to the latent space.

3.2 Dataset and Additional Elements

The dynamic model is trained with high-fidelity solutions to the three-dimensional Navier-Stokes equations, defined by 1. Solutions for a triply-periodic domain are obtained by a standard Fourier psuedo-spectral method [c.f. 18]. The system is forced at low-wavenumbers to keep the total energy in the system constant [19]. The resulting data are statistically stationary with approximately stationary energy spectra. Our training, validation and test datasets span 100 integral time scales τ, with a temporal sampling rate of approximately $\tau/100$. The turbulent Reynolds number, $Re_T \equiv e^2/(\nu\epsilon)$ with ϵ the turbulent kinetic energy dissipation rate, is approximately 380, and the flow remains fully turbulent for the duration of the solution. The solutions are discretized with 64 collocation points in each direction.

We incorporate a few additional architectural elements to improve prediction and enforce physical constraints. As the dataset includes periodic boundary conditions, we impose this constraint by padding inputs to all convolutions circularly. We also employ the augmented neural ODE approach [20] to the latent dynamics to increase flexibility and ease computation of ODE. This is accomplished by concatenating channels of zeros to the output of the encoder, with the predicted results projected back into the original latent space. Lastly, we apply a spectral projection layer to the final model output to enforce the divergence-free field condition required for a constant density fluid [21].

4 Results

4.1 A-posteriori statistical analysis

We analyze the predictive capabilities of our model by examining turbulent energy spectra over a window of one integral time scale. The energy spectrum decomposes the turbulent kinetic energy in the flow as a function of wavenumber, where large and small eddies correspond to low and high wavenumbers respectively. As this system is chaotic, we are interested in statistical assessments of performance as opposed to local measures such as RMSE. We look at the PDFs of the spatial velocity

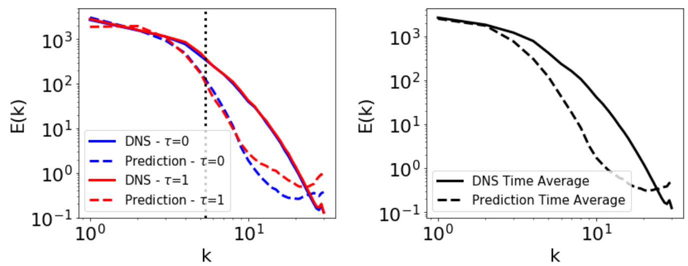

Figure 2: Both plots show the energy of the flow as a function of wavenumber. Left: Two snapshots corresponding to the initial and ending time steps are shown. The vertical line represents the compression ratio of 6 defined by the latent space size. Right: The energy spectra averaged over the entire time window is shown. Low wavenumbers show better agreement with DNS, which are sufficient for many practical flows.

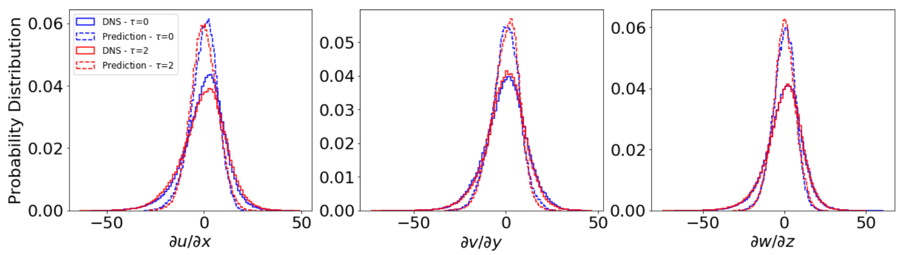

Figure 3: Distributions of the velocity gradients in the flow at initial and end temporal snapshots. Predictions exhibit tighter distributions, but they are consistent over time.

gradients at two snapshots, shown in fig. 3, which highlight the performance at the small scales of the flow. From the spectra, this approach models the large scales of the flow very well, shown in fig. 2 with increasing losses at higher wavenumbers. We note that for many engineering applications, resolving these largest scales is often sufficient to realize dynamics of interest for engineering or geophysical problems [15]. Both the velocity PDFs and the spectra plots indicate that the two do not vary appreciably over time, which is consistent with stationarity imposed in the training dataset.

We can also calculate the average turbulent kinetic energy in the flow over time, simply the integral under the curve of the energy spectra, to gain insight into how the model behaves over time. For this dataset, we expect to see constant energy. While there is a decrease in kinetic energy for the predictions initially, we eventually see the kinetic energy approach an approximately stationary state.

4.2 Parametric Studies

Next, we compare these results with several other model architectures. Fig. 4 shows the energy spectra of a model with a larger compression ratio of 24. Again, we see reasonably good approximations of the large scales, but there is more substantial deviation and irregularity at the small scales and the model exhibits considerably less stationarity, indicating instability.

Lastly, we see significant correlation between the kernel size of the convolutional neural ODE and the resultant average TKE over time. From fig. 4, although every model under-predicts the true

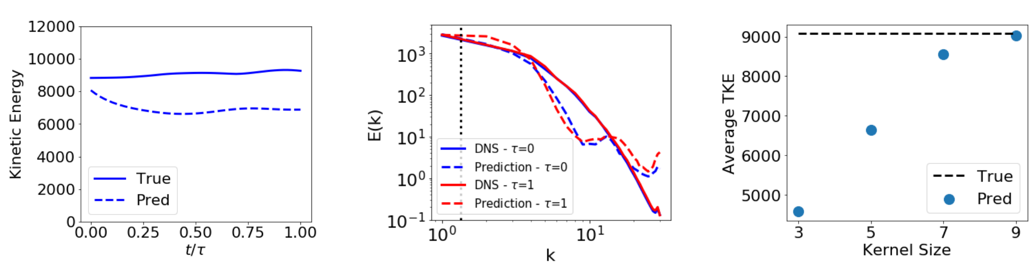

Figure 4: Left: Evolution of the kinetic energy in DNS and model flows. There is an initial loss of energy in the prediction, but it remains constant afterwards. Center: Energy spectra of a model with a much higher compression ratio of 24. The spectra are more variable over time, and contain artifacts from the compression at high wavenumbers, which is why there appears to be some agreement in the range $k \approx$13-20. Right: Average kinetic energy of 4 models with varying convolutional kernel sizes. Larger kernel models seem to approach the ground truth asymptotically.

energy, increasing the kernel size allows the model's TKE to approach the true value. As the kernel size corresponds to the order of spatial discretization scheme, we find that higher order schemes are particularly better at capturing the kinetic energy of the flow, which is a function of the magnitude of velocity fluctuations. The kernel size and overall receptive field of the neural ODE can also be connected the the length scales that are represented in the dynamics. Small receptive fields will be limited to local interactions while larger fields can include larger scale interactions. Incorporating an increased range of interactions effects better performance.

5 Conclusions

In this work, we have proposed a novel methodology for forecasting complex nonlinear spatiotemporal dynamics through purely data-driven methods. We use a trainable encoder-decoder pair to transform the field data into a latent space that retains the spatial nature of the data and solve the transient problem on this space using standard ODE methods. The governing dynamics of the latent space are approximated using a convolutional neural net. We find that this approach is able to capture important details of the flow, namely the large-scale dynamics, and produces stationary sequences in line with the physics of the problem. We also find convolutional network details, both the encoder-decoder and the dynamics estimator, have a significant impact on the results, informing future developments in this field.

6 Broader Impact Statement

This work introduces a data-driven method for learning turbulent flow dynamics using high-fidelity simulation data. Incorporating machine learning techniques into scientific computing has largely focused on balancing computation accuracy and speed, and in the proposed approach, we aim to replace traditional, computationally-intensive numerical simulations with a fast ML model. We show that while there is some loss in fidelity, our models are suitable for many engineering problems, which make them an ideal candidate to include in workflows that require rapid fluid dynamics calculations, such as down selection and design optimization. While our present work demonstrates this for this particular dataset, the results indicate a promising pathway to address challenges with current numerical techniques for non-linear spatio-temporal dynamics.

7 Acknowledgements

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE 1745016 awarded to VS. The authors from CMU acknowledge the support from the Technologies for Safe and Efficient Transportation University Transportation Center, and Mobility21, A United States Department of Transportation National University Transportation Center. The authors further acknowledge the support of NVIDIA via The ICEnet Consortium, for providing necessary GPU compute. The authors from LANL acknowledge funding from a Laboratory Directed Research & Development (LDRD) project entitled "Machine Learning for Turbulence (MELT)" (20180059DR). This work is approved for public release by the Los Alamos National Laboratory as LA-UR-20-27785.

References

[1] Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating Eulerian fluid simulation with convolutional networks. volume 70 of *Proceedings of Machine Learning Research*, pages 3424–3433, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

[2] Romit Maulik, Himanshu Sharma, Saumil Patel, Bethany Lusch, and Elise Jennings. Accelerating RANS simulations using a data-driven framework for eddy-viscosity emulation. *arxiv:1910.10878 [physics]*, 2019.

[3] Gavin D. Portwood, Balasubramanya T. Nadiga, Juan A. Saenz, and Daniel Livescu. Interpreting neural network models of residual scalar flux. *Journal of Fluid Mechanics*, 2020. in press.

[4] Peetak Mitra, Mateus Dias Ribeiro, and David Schmidt. A data-driven approach to modeling turbulent flows in an engine environment. *APS*, pages G16–003, 2019.

[5] Filipe de Avila Belbute-Peres, Thomas D. Economon, and J. Zico Kolter. Combining differentiable PDE solvers and graph neural networks for fluid flow prediction. *arXiv:2007.04439 [cs.LG]*, 2020.

[6] Gavin D. Portwood, Peetak P. Mitra, Mateus Dias Ribeiro, Tan Minh Nguyen, Balasubramanya T. Nadiga, Juan A. Saenz, Michael Chertkov, Animesh Garg, Anima Anandkumar, Andreas Dengel, and et al. Turbulence forecasting via neural ODE. *arXiv:1911.05180 [physics]*, Nov 2019.

[7] Nils Thuerey, Konstantin Weißenow, Lukas Prantl, and Xiangyu Hu. Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows. *AIAA Journal*, 58(1):25–36, Jan 2020.

[8] Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. Towards physics-informed deep learning for turbulent flow prediction. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Aug 2020.

[9] A. Mohan, Don Daniel, M. Chertkov, and D. Livescu. Compressed convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence. *arXiv:1903.00033 [physics.flu-dyn]*, 2019.

[10] Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. Learning dynamical systems from partial observations. *arXiv:1902.11136 [physics]*, Feb 2019.

[11] 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*, 378:686–707, Feb 2019.

[12] Arvind T Mohan, Nicholas Lubbers, Daniel Livescu, and Michael Chertkov. Embedding hard physical constraints in neural network coarse-graining of 3D turbulence. *arXiv preprint arXiv:2002.00021*, 2020.

[13] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. *arXiv:1806.07366 [cs, stat]*, Dec 2019.

[14] Yaofeng Desmond Zhong, Biswadip Dey, and Amit Chakraborty. Symplectic ODE-Net: Learning hamiltonian dynamics with control. In *International Conference on Learning Representations*, 2020.

[15] Stephen B. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

[16] Julien Weiss. A tutorial on the proper orthogonal decomposition. In *AIAA Aviation 2019 Forum*. American Institute of Aeronautics and Astronautics, Jun 2019.

[17] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 5320–5330. Curran Associates, Inc., 2019.

[18] Steven A Orszag and GS Patterson Jr. Numerical simulation of three-dimensional homogeneous isotropic turbulence. *Physical Review Letters*, 28(2):76, 1972.

[19] M. R. Overholt and S. B. Pope. A deterministic forcing scheme for direct numerical simulations of turbulence. *Comput. Fluids*, 27:11–28, 1998.

[20] Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural ODEs. In *Advances in Neural Information Processing Systems*, pages 3140–3150, 2019.

[21] Chiyu ”Max” Jiang, Karthik Kashinath, Philip Marcus, et al. Enforcing physical constraints in CNNs through differentiable PDE layer. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.