Physics-Driven Convolutional Autoencoder Approach for CFD Data Compressions

Alberto Olmo
National Renewable Energy Laboratory
Golden, CO 80401
aolmoher@asu.edu

Ahmed Zamzam
National Renewable Energy Laboratory
Golden, CO 80401
ahmed.zamzam@nrel.gov

Andrew Glaws
National Renewable Energy Laboratory
Golden, CO 80401
andrew.glaws@nrel.gov

Ryan King
National Renewable Energy Laboratory
Golden, CO 80401
ryan.king@nrel.gov

Abstract

With the growing size and complexity of turbulent flow models, data compression approaches are of the utmost importance to analyze, visualize, or restart the simulations. Recently, in-situ autoencoder-based compression approaches have been proposed and shown to be effective at producing reduced representations of turbulent flow data. However, these approaches focus solely on training the model using point-wise sample reconstruction losses that do not take advantage of the physical properties of turbulent flows. In this paper, we show that training autoencoders with additional physics-informed regularizations, e.g., enforcing incompressibility and preserving enstrophy, improves the compression model in three ways: (i) the compressed data better conform to known physics for homogeneous isotropic turbulence without negatively impacting point-wise reconstruction quality, (ii) inspection of the gradients of the trained model uncovers changes to the learned compression mapping that can facilitate the use of explainability techniques, and (iii) as a performance byproduct, the update of the network to accommodate for our training losses shows to train up to 12x faster than the baseline model.

1 Introduction

With the advancement of high performance computing (HPC) there has been an increase in the interest of leveraging such systems for computational fluid dynamics (CFD) simulations. These have become more readily available with larger than ever data sizes and performance fidelity [Sprague et al., 2020, Fischer et al., 2021, Musser et al., 2022]. Further, recent advances in computational processing power achieved through heterogeneous architectures that couple traditional processors with graphics processing units (GPUs) have led to an increase in the gap between processing power and memory due to bandwidth constraints and memory-access times given by high latency input/output operations. This can lead to memory-bottleneck scenarios where HPC machines are limited by the need to save, analyze, visualize, or restore data from massive simulations. Given these factors and the increase in available datasets, it becomes critically important to develop in-situ data compression techniques to enable efficient use of the data without sacrificing accuracy. Additionally, a recent report from the U.S. Department of Energy (DoE), shows the analysis and visualization of CFD simulations as a central issue for next generation systems [Gerber et al., 2018].

Several lossy compression approaches have been proposed recently [Fukami et al., 2020, Glaws et al., 2020, Carlberg et al., 2019, Dunton et al., 2020] utilizing singular value decompositions or

Machine Learning and the Physical Sciences workshop, NeurIPS 2022.

neural networks. In particular, convolutional autoencoders have proved to be able to obtain better generalization results [Glaws et al., 2020]. However, these approaches only leverage sample quality metrics while missing the physical properties inherent in the CFD and the benefits of embedding them at training time. Therefore, motivated by the big successes of convolutional neural networks in the processing of CFD data [Guo et al., 2016, Tompson et al., 2017] and in-situ data compression tasks [Liu et al., 2019] and the increased ability to generate large CFD data sets, we develop a physics-driven convolutional autoencoder compression approach that builds on previous work [Glaws et al., 2020].

In this work, we show that using an autoencoder model enhanced with two physical properties of CFD leads to compression models that are more conformant with the physical characteristics of the data as measured by known metrics for homogeneous isomorphic turbulent flow, including the divergence-free condition of incompressible flow fields and the preservation of both enstrophy and dissipation ratio [Constantin and Foias, 2020]. In addition, analyzing the model performance shows a significant reduction in training time as well as a reduced amount of training data necessary to achieve same-quality reconstructions as compared to the baseline. Further, our preliminary analysis of the learned models shows better explicability compared to models trained using only sample quality metrics. All of this encourage the use of such networks with physical-losses and illustrate that gradient-based explainability techniques can be leveraged in the future. [1]

2 Approach

While lossless data compression approaches can be options for this problem [Fout and Ma, 2012, Lindstrom and Isenburg, 2006], they pose memory and execution time burdens for the system. On the other hand, lossy data compression aims to reduce the memory consumption by incurring some manageable loss of information after decompression. Hence, in our problem of reducing the dimensionality of CFD data and their memory expense, lossy compression methods are more suitable. Thus, a general lossy data compression function with full data space $\mathcal{X}$ and compressed data space $\mathcal{Y}$, can be defined in two parts: the compression step $\phi : \mathcal{X} \to \mathcal{Y}$ and reconstruction step $\psi : \mathcal{Y} \to \mathcal{X}$ where the degree of compression is measured by the compression ratio (CR). To this end, we design a convolutional autoencoder with compression function ϕ defined by the encoder E, data x and embedded data z such that $E(x) = z$ and decompression function ψ by the decoder D such that $D(z) = \hat{x}$. Thus, the compressed data is obtained from $E(x)$ which can be stored more easily than the full data, and the full data reconstruction can be recovered with $D(z)$.

Data The dataset we use consists of simulated snapshots of fluid velocities from incompressible decaying isotropic flows with component velocities on the x, y and z dimensions. Thus, the vector fields are comprised of 3-dimensional meshes of $128 \times 128 \times 128$ datapoints generated by the spectralDNS package [Mortensen and Langtangen, 2016]. To increase robustness of the network, we introduce turbulences in the simulations as measured by Taylor-scale Reynolds numbers between (65, 105) and gather a total of 1300 snapshots for our training dataset.

Physics-informed loss The network follows a fully convolutional architecture. In general, the main goal of a parameterized autoencoder is to minimize the reconstruction error with some pointwise metric such as the squared 2-norm:

$$\Theta_E, \Theta_D = \underset{\Theta_E, \Theta_D}{\operatorname{argmin}} ||x - D(E(x; \Theta_E); \Theta_D)||_2^2. \tag{1}$$

Recent works have shown the advantages of using the physical properties of the domain during training. For instance, in Raissi et al. [2019], the authors train shallow neural networks with losses that include domain-specific physics laws and show improved generalization performance of the trained models. Similarly, Cai et al. [2022] show how physics-informed learning improves the inference performance for CFD domains such as three-dimensional wake flows or supersonic flows. Thus, in addition to MSE (1), we design the autoencoder loss with two physical laws that are applicable to the domain in consideration, the divergence-free condition and the preservation of enstrophy. For the former, due to the incompressibility of the flow field, the density of the CFD remains constant expressed by $\nabla \cdot \vec{v} = 0$ and therefore is a property that can be enforced. Similarly, the enstrophy of a fluid measures the kinetic energy in the flow that corresponds to dissipation

[1] The code of this work is attached as supplementary material and will be made publicly available.

Model		Dissipation Rate	MSE	Mean Divergence Loss	Mean Enstrophy Loss
Vanilla ($\lambda = 0$, $\beta = 0$)		0.296	0.040	0.632	9.4e-5
Divergence λ	1e-2	0.476	0.047	0.080	10e-5
	1e-1	0.296	0.042	0.018	11e-5
	1	0.496	0.083	**0.003**	11e-5
	10	0.0001	0.967	3.6e3	6.9e5
Enstrophy β	1e-2	0.304	0.037	0.581	8.7e-5
	1e-1	0.411	**0.036**	0.664	9.6e-5
	1	0.304	0.037	0.578	8.2e-5
	10	**0.504**	0.066	0.682	7.5e-5

Ground truth dissipation rate: 0.544 Baseline model's MSE: 0.0604

Table 1: Quantitative evaluations between our vanilla model and our model when trained with varying λ and β. Each was trained with the same amount of data (1300 snapshots) and epochs (150). Ground truth dissipation rate is 0.544 and closer values imply a better preservation. Baseline [Glaws et al., 2020] MSE is 0.0604 when trained with same data (only one channel at a time), epochs and learning rate.

effects and can be ensured to remain consistent between the original data and its reconstruction. Incorporating these properties into the training procedure, the loss becomes:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}(x_i - f(x_i;\theta))^2 + \frac{\lambda}{N}\sum_{i=1}^{N}(\nabla \cdot f(x_i;\theta))^2 + \frac{\beta}{N}\sum_{i=1}^{N}(g(x_i) - g(f(x_i;\theta))^2, \quad (2)$$

where we denote a forward compression and decompression pass of the mesh x_i on the network with parameters θ as $f(x_i;\theta) = D(E(x_i))$ and enstrophy $g(x)$ expressed in terms of the flow velocity as $g(x) \equiv \int_S |\nabla \times u|^2 dS$. We include the hyperparameters λ and β that allow tuning the sensitivity of the divergence-free minimization and preservation of enstropy respectively.

Secondly, we adapt the architecture from Glaws et al. [2020] in two ways. First, the model is adapted to handle simultaneous compressions of 3-channel velocity data in contrast to the one dimensional input of the former. Second, as wider layers are introduced to account for compressing this 3-velocity data, we adjust the network by removing its residual block layers. This proved to be an effective regularization method for the network and shows the network to be learning meaningful connections that the single-velocity baseline could have never inferred as we show next.

3 Experiments and Results

We measure the improvements of our model by comparing it to the *vanilla* version (when $\lambda = 0$ and $\beta = 0$) as well as the version from Glaws et al. [2020] (referred to as the *baseline* version) in terms of quantitative and performance measurements which we outline below. For all experiments and models (both baseline and ours) we use the optimal baseline parameters as stated in Glaws et al. [2020]: a compression ratio of $CR = 64$, batch size of 12, learning rate of $\eta = 1e^{-4}$ with scheduled decay and Adam optimizer [Kingma and Ba, 2015]. We train the models on a node from an HPC machine containing two NVIDIA Tesla V100 GPUs with 16 GB of dedicated memory and dual Intel Xeon Gold Skylake 6154 processors.

Quantitative Experiments We use four main evaluation metrics to assess the quality of the reconstructions over 1300 samples: (i) the divergence of the flow field, (ii) the enstrophy mismatch loss, (iii) the pointwise mean squared error (as shown in (1)), and (iv) the dissipation rate mismatch (given by $2\nu\langle S_i j S_i j\rangle$). We compile our results in Table 1. Only one of λ or β was changed at each experiment while leaving the other hyperparameter at 0. From these results, we obtain several promising observations: our vanilla model ($\lambda = 0$ and $\beta = 0$) overperforms the baseline when trained under the same constraints in terms of the MSE reconstruction loss by a 33% (0.04 vs 0.06). Further, the divergence loss of the new model is effectively reduced proportionally to higher λ values (with

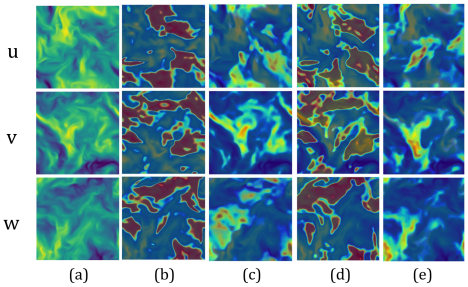

Figure 1: Grad-CAM heatmaps over the u, v and w velocities on the learned gradients of the last layer of our model trained with $\lambda = 0$, $\beta = 0$ (c) $\lambda = 1$ and $\beta = 0$ (d) and $\lambda = 0$ and $\beta = 1$ (e). (a) and (b) columns represent the 3-channel snapshot data and the baseline's last layer gradients respectively.

Model	Training samples	Training Time	MSE Loss
Baseline	1300	24h (50 epochs)	0.045
Ours	1300	**6h (50 epochs)**	**0.028**
Baseline	566	20.5h (100 epochs)	0.045
Ours	566	5.5h (100 epochs)	**0.030**
Ours	566	**2.8h (50 epochs)**	0.045
Baseline	200	12h (180 epochs)	0.052
Ours	200	**54m (45 epochs)**	**0.045**

Table 2: Training times and corresponding mean squared error loss for both the baseline model and ours with $\lambda = 0$ and $\beta = 0$ when trained with different amounts of data and epochs.

the exception of $\lambda = 10$ which makes the reconstruction loss explode) making the reconstructed snapshots more conformant with the divergence-free condition of the fluid data. On the other hand, the ground truth dissipation rate sits at 0.544 and while training with $\lambda > 0$ values on average seem to yield its better preservation, our model achieves the closest at $\beta = 10$ with 0.504. It is worth noting that, overall, the inclusion of our physics terms during training shows to consistently improve the dissipation rate when compared to the vanilla as given by 6 out of the 8 models.

Therefore, we show that the inclusion of the divergence-free and enstrophy-preserving terms to the loss prove not only to make the reconstructions more conformant with the physical laws of the domain but also these come at no significant expense in terms of reconstruction quality as measured by the MSE, even improving it in some cases. In the future, experimentation with combinatorial values of λ and β can be tested.

Performance Experiments We also assess the performance of the models in terms of training time by feeding and comparing both with same amounts of training samples (1300, 566 and 200 snapshots) and configurations. We report the results in Table 2. We note that our model significantly reduces the training time on average by a factor of 4 consistently over varying amounts of training data. At the same time, our model reaches better MSE loss when trained for the same number of epochs as shown in the first four sub-rows of Table 2. Thus, this allows the model to perform equally or better with fewer epochs or data as shown in the second row where ours gets the same reconstruction quality in half of the epochs (50) and a speedup of 7.3x. It is worth noting that our model trains with 3-channeled data as opposed to the baseline which does one channel at a time, thus the speedup of the reported training times is further improved by a factor of 3 leading to up to a total of 12x speedup.

Explainability Experiments To examine the explainability of the reconstructions, we use a second order gradient-based technique that produces visual explanations via heatmaps from the gradient information of the last convolutional layer after a reconstruction [Selvaraju et al., 2020]. We show preliminary results in Fig. 1 and observe that the attention of the network changes depending on the regularization loss it was trained on. While further experimentation is needed to draw conclusions, it is noticeable that the models trained using physics-informed losses focus on regions with higher magnitudes while the baseline models focus on random parts of the data. Due to this and the physical improvements of training with our loss, shown in Table 1, we can infer that our showcased heatmaps can potentially become more focused on physically-relevant features and open up encouraging directions to further explore them due to the increasing need for explainability in deep learning for CFDs.

4 Conclusion and Future Work

In this work, we tackle the problem of CFD data compression using autoencoders. Unlike recent literature focused on pointwise losses and motivated by the success of physics-informed constraints, we adapt and improve a baseline model to account for two physical laws of the CFD data that is trained on. We show that the benefits of the proposed approach are threefold: (i) the reconstructions effectively

learn these laws, becoming more physics-conformant with the domain at no expense (or even improvement) in reconstructions quality, (ii) the adaptation of the model for 3-channel flow field data makes the network learn patterns between them allowing to reduce its depth, consequently speeding up its expensive training times significantly by a factor of up to 12x, and (iii) we apply gradient-based heatmaps on the last layer of the models and show that potentially more explicable patterns arise when trained with different weighted values of the physics-informed terms. This highlights pathways for future investigations to the interpretability of the compressions, and encourages the use of autoencoders and physics-informed losses as a potential candidate to tackle the rising interest in CFD models' explainability.

5 Acknowledgements

This work was authored by the National Renewable Energy Laboratory (NREL), operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. This work was supported by the Laboratory Directed Research and Development (LDRD) Program at NREL and by the DOE Office of Science under Advanced Scientific Computing Research (ASCR) program. This research was performed using computational resources sponsored by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy and located at the National Renewable Energy Laboratory. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

6 Broader Impact

As computational fluid dynamics (CFD) simulations become more computationally expensive, it is of utmost importance to develop efficient methods that compress them for these to become more tractable in the future in terms of restart checkpointing, improving space storage constraints and visualization. Our work follows along these lines and enhances a current version of such compression methods (autoencoder model) where we show that not only we maintain the original compression ratio of the baseline method we compare against, but also that these become more physically-conformant with the CFD dataset they are being trained at no pointwise similarity expense. At the same time, we improve the model such that it becomes up to 12x faster to train. Thus, our work encourages the use of physics-informed losses for CFD data compression autoencoders to the physics and machine learning communities.

This work is nonetheless not free from limitations and, as we mention above, our gradient-based explainability results are still in their infancy. However, we remark these are promising directions and we encourage them in order to potentially explain how autoencoders perform their compressions.

References

Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, pages 1–12, 2022.

Kevin T Carlberg, Antony Jameson, Mykel J Kochenderfer, Jeremy Morton, Liqian Peng, and Freddie D Witherden. Recovering missing CFD data for high-order discretizations using deep neural networks and dynamics learning. *Journal of Computational Physics*, 395:105–124, 2019.

Peter Constantin and Ciprian Foias. *Navier-Stokes Equations*. University of Chicago Press, 2020.

Alec M Dunton, Lluís Jofre, Gianluca Iaccarino, and Alireza Doostan. Pass-efficient methods for compression of high-dimensional turbulent flow data. *Journal of Computational Physics*, 423: 109704, 2020.

Paul Fischer, Stefan Kerkemeier, Misun Min, Yu-Hsiang Lan, Malachi Phillips, Thilina Rathnayake, Elia Merzari, Ananias Tomboulides, Ali Karakus, Noel Chalmers, et al. NekRS, a GPU-accelerated spectral element Navier-Stokes solver. *arXiv preprint arXiv:2104.05829*, 2021.

Nathaniel Fout and Kwan-Liu Ma. An adaptive prediction-based approach to lossless compression of floating-point volume data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12): 2295–2304, 2012. doi: 10.1109/TVCG.2012.194.

Kai Fukami, Taichi Nakamura, and Koji Fukagata. Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data. *Physics of Fluids*, 32:095110, 2020.

Richard Gerber, James Hack, Katherine Riley, Katie Antypas, Richard Coffey, Eli Dart, Tjerk Straatsma, Jack Wells, Deborah Bard, Sudip Dosanjh, Inder Monga, Michael E. Papka, and Lauren Rotman. Crosscut report: Exascale Requirements Reviews, March 9–10, 2017 – Tysons Corner, Virginia. An Office of Science review sponsored by: Advanced Scientific Computing Research, Basic Energy Sciences, Biological and Environmental Research, Fusion Energy Sciences, High Energy Physics, Nuclear Physics. 1 2018. doi: 10.2172/1417653. URL https://www.osti.gov/biblio/1417653.

Andrew Glaws, Ryan King, and Michael Sprague. Deep learning for in situ data compression of large turbulent flow simulations. *Physical Review Fluids*, 5(11):114602, 2020.

Xiaoxiao Guo, Wei Li, and Francesco Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 481–490, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939738. URL https://doi.org/10.1145/2939672.2939738.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Peter Lindstrom and Martin Isenburg. Fast and efficient compression of floating-point data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1245–1250, 2006. doi: 10.1109/TVCG.2006.143.

Yang Liu, Yueqing Wang, Liang Deng, Fang Wang, Fang Liu, Yutong Lu, and Sikun Li. A novel in situ compression method for CFD data based on generative adversarial network. *J. Vis.*, 22(1):95–108, 2019. doi: 10.1007/s12650-018-0519-x. URL https://doi.org/10.1007/s12650-018-0519-x.

Mikael Mortensen and Hans Petter Langtangen. High performance python for direct numerical simulations of turbulent flows. *Computer Physics Communications*, 203:53–65, 2016. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2016.02.005. URL https://www.sciencedirect.com/science/article/pii/S0010465516300200.

Jordan Musser, Ann S Almgren, William D Fullmer, Oscar Antepara, John B Bell, Johannes Blaschke, Kevin Gott, Andrew Myers, Roberto Porcu, Deepak Rangarajan, et al. MFIX-Exa: A path toward exascale cfd-dem simulations. *The International Journal of High Performance Computing Applications*, 36(1):40–58, 2022.

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, 2019. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2018.10.045. URL https://www.sciencedirect.com/science/article/pii/S0021999118307125.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.*, 128(2):336–359, 2020. doi: 10.1007/s11263-019-01228-7. URL https://doi.org/10.1007/s11263-019-01228-7.

Michael A Sprague, S Ananthan, Ganesh Vijayakumar, and Michael Robinson. ExaWind: A multifidelity modeling and simulation environment for wind energy. In *Journal of Physics: Conference Series*, volume 1452, pages 012–071. IOP Publishing, 2020.

Jonathan Tompson, Kristofer Schlachter, Pablo Sprechmann, and Ken Perlin. Accelerating Eulerian fluid simulation with convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=ByH2gxrKl`.

7 Paper Checklist

1. Do the claims made in the abstract and introduction accurately reflect the paper's contributions and scope? **[Yes]**
2. Did you discuss any potential negative societal impacts of your work? **[No]**
3. Have you read the ethics review guidelines and ensured that your paper conforms to them? **[Yes]**
4. Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? **[Yes]**
5. Did you describe the limitations of your work? **[Yes]**
6. Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? **[Yes]**
7. Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? **[No, due to the expensive training of each model, we resort to only train a few and show the hyperparameters used.]**
8. Did you include the amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? **[Yes]**
9. If your work uses existing assets, did you cite the creators? **[Yes]**
10. Did you include any new assets either in the supplemental material or as a URL? **[Yes]**
11. Did you discuss whether and how consent was obtained from people whose data you're using/curating? **[No, the data used is publicly available]**
12. Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? **[No, there is no personally identifiable information in our data]**