A Physics-Informed Vector Quantized Autoencoder for Data Compression of Turbulent Flow

Mohammadreza Momenifar
Enmao Diao
Vahid Tarokh
Andrew Bragg

Pratt School of Engineering
Duke University

Motivation

- Large-scale data from **three-dimensional (3D)** high-fidelity simulations of **turbulent flows** is **memory intensive**, requiring significant resources to store, transfer and process the data

- This major challenge highlights the need for **data compression** techniques. This motivated us to propose an attractive solution for situations where **fast, high quality and low-overhead** encoding and decoding of large data are required

Proposed Framework

- We propose a physics-informed Deep Learning framework to generate a **discrete** low-dimensional representation of velocity field data of **three-dimensional** turbulent flow simulations

- Our method can offer at least **compression ratio** (**CR**) **= 85** and predictions that **faithfully reproduce** the statistics of the flow, except at the very smallest scales

Model Architecture

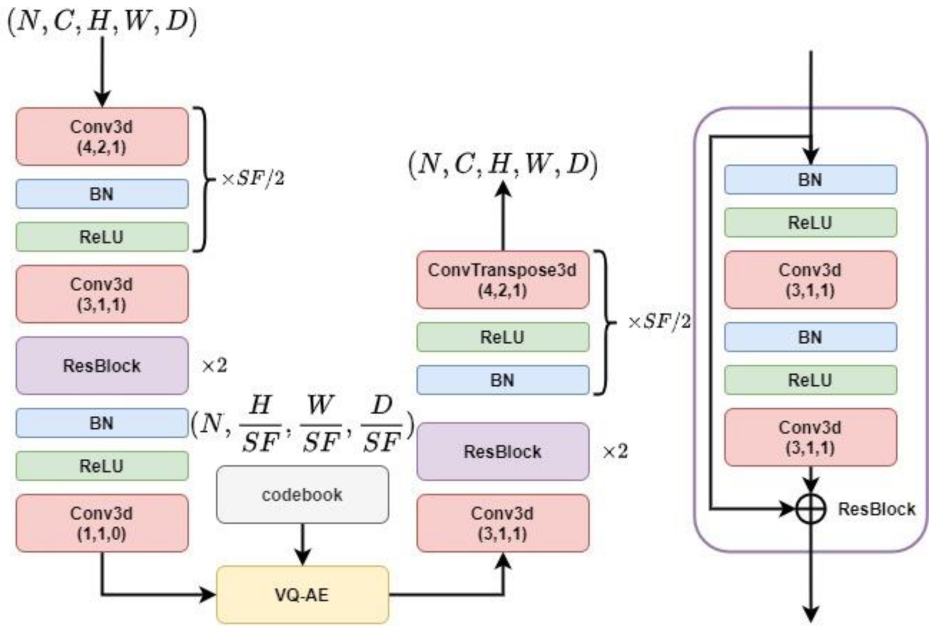
- Vector Quantized Autoencoder (VQ-AE) [1] learns a discrete (rather than continuous), low-dimensional representation of data
- Scaling Factor (**SF**) and Compression Ratio (**CR**)

$$SF \in \{2,4,6\}$$

$$CR = \frac{3 \times 32}{1 \times 9} \times SF^3$$

$$Quantize(E(x)) = e_k, \text{ where } k = \arg\min_j \| E(x) - e_j \|_2$$

$$\mathcal{L}(x, D(e)) = \underbrace{\| x - D(e) \|_2^2}_{reconstruction\ loss} + \underbrace{\| sg\{E(x)\} - e \|_2^2}_{codebook\ loss} + \underbrace{\beta \| sg\{e\} - E(x) \|_2^2}_{commitment\ loss}$$



[1] Van Den Oord, Aaron, and Oriol Vinyals. "Neural discrete representation learning." *Advances in neural information processing systems* 30 (2017).

Prior Physics-Informed Knowledge

- Incorporating prior physics-informed knowledge into framework

Embed physics constraints of isotropic turbulence, particularly those of velocity gradient tensor $A_{ij} = \partial u_i / \partial x_j$ and "Betchov relations":

$$\langle S_{ij} S_{ij} \rangle = \langle R_{ij} R_{ij} \rangle = \frac{1}{2} \langle \omega_i \omega_i \rangle \; ; \; \langle S_{ik} S_{kj} S_{ij} \rangle = -\frac{3}{4} \langle S_{ij} \omega_i \omega_j \rangle$$

where $S_{ij} \equiv (1/2)(A_{ij} + A_{ji})$ is the strain-rate, $R_{ij} \equiv (1/2)(A_{ij} - A_{ji})$ is the rotation-rate, and $\omega_i = \epsilon_{ijk} R_{jk}$ is the vorticity

$$\text{Velocity Gradient Constraint (VGC)} = \underbrace{\text{MSE}(A_{ij}, \widehat{A_{ij}})}_{i=j} + a \times \underbrace{\text{MSE}(A_{ij}, \widehat{A_{ij}})}_{i \neq j}$$

$$\text{Higher Order Constraints (HOC)} = \text{MAE}(\langle S_{ij} S_{ij} \rangle, \widehat{\langle S_{ij} S_{ij} \rangle}) + \text{MAE}(\langle R_{ij} R_{ij} \rangle, \widehat{\langle R_{ij} R_{ij} \rangle}) +$$
$$\text{MAE}(\langle S_{ik} S_{kj} S_{ij} \rangle, \widehat{\langle S_{ik} S_{kj} S_{ij} \rangle}) + \text{MAE}(\langle S_{ij} \omega_i \omega_j \rangle, \widehat{\langle S_{ij} \omega_i \omega_j \rangle}),$$

$$\text{Overall Loss (OL)} \;\; = \text{VQ-AE loss} + \alpha \times \text{VGC} + \gamma \times \text{HOC}$$

The Forward and Backward paths are performed in **Fourier space** using FFT

Experimental Setup

- We **train** our model via only **40 realizations** from Direct Numerical Simulation (DNS) data of a **three-dimensional, statistically stationary, isotropic turbulent** flow simulated on a cubic domain with 128 grid point in each direction

- We implemented this framework with **PyTorch**. The training (**one** NVIDIA Pascal P100 GPU) was completed within approximately **8 hours**, and the maximum GPU memory consumed was around **5 GB**

Evaluation

- The model is evaluated using statistical, comparison-based similarity and **physics-based metrics** (PDFs of velocity gradient tensor and its invariants, Turbulence kinetic energy spectra)

- The performance of this lossy data compression scheme is evaluated on a variety of test data on the order of increasing complexity: **stationary isotropic turbulence**, **decaying isotropic turbulence**, **Taylor-Green vortex flow**

Evaluation

Table 1: Summary of the performance on unseen data from statistically stationary isotropic, decaying isotropic, and decaying Taylor-Green vortex turbulence.

Turbulent Flow	CR	Method	MSE	MAE	MSSIM	HOC
stationary isotropic	85	VQ-AE	**0.0044**	**0.0499**	**0.977**	**18.37**
	683		0.0201	0.1070	0.909	51.25
	5461		0.1900	0.3240	0.600	112.59
decaying isotropic	64	SVD [8]	2.8043	2.2944	0.198	N/A
	64	AE [8]	0.0865	0.3744	0.946	
	85	VQ-AE	**0.0018**	**0.0326**	**0.970**	**9.81**
	683		0.0080	0.0693	0.882	20.39
	5461		0.0504	0.1720	0.598	37.83
decaying Taylor-Green vortex	64	SVD [8]	0.0253	0.2112	0.398	N/A
	64	AE [8]	**0.0017**	0.0483	**0.953**	
	85	VQ-AE	0.0027	**0.0395**	0.830	**1.33**

Experimental Results

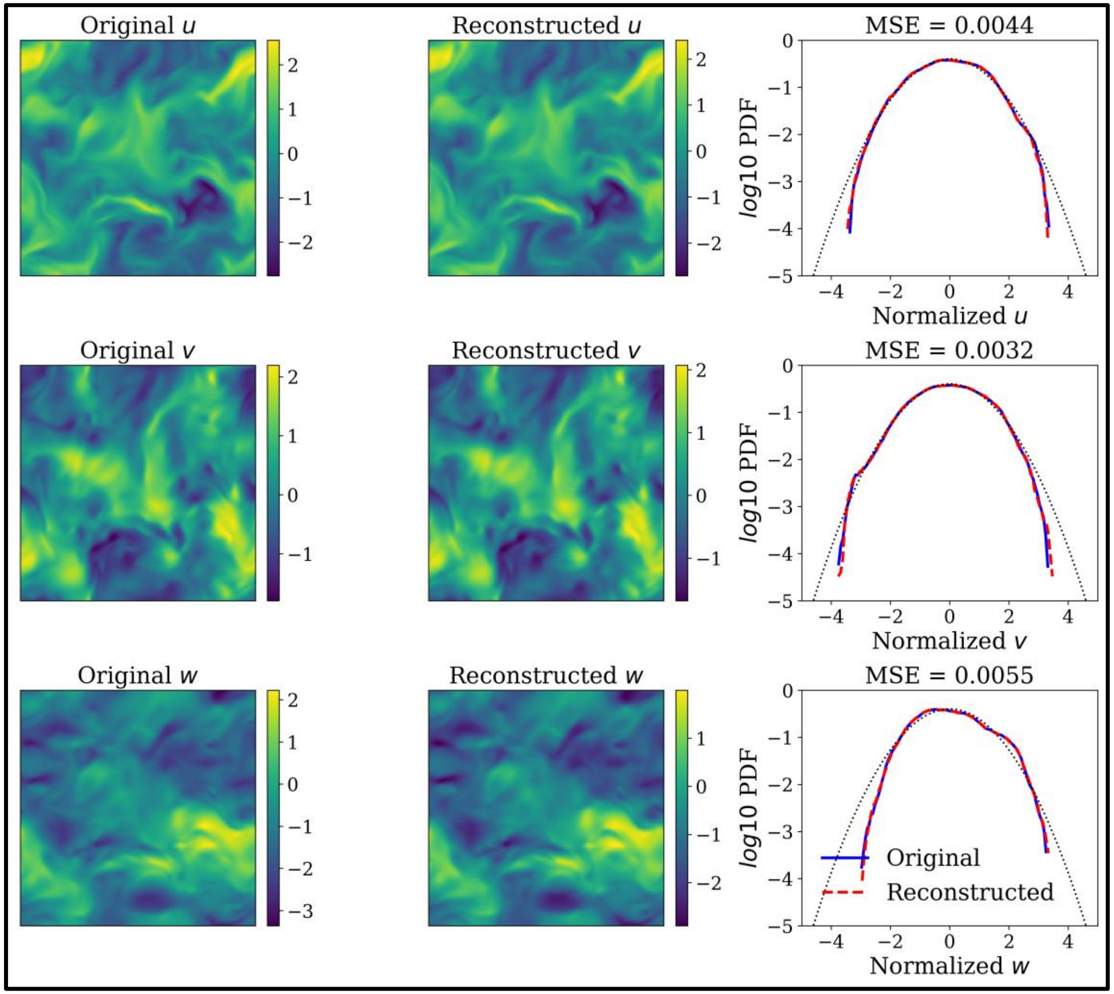


Figure 1. 2D snapshots and PDFs of velocity components. Comparing original and reconstructed 3D test data from **statistically stationary isotropic turbulence** compressed via VQ-AE with SF= 2 (CR= 85).

Experimental Results

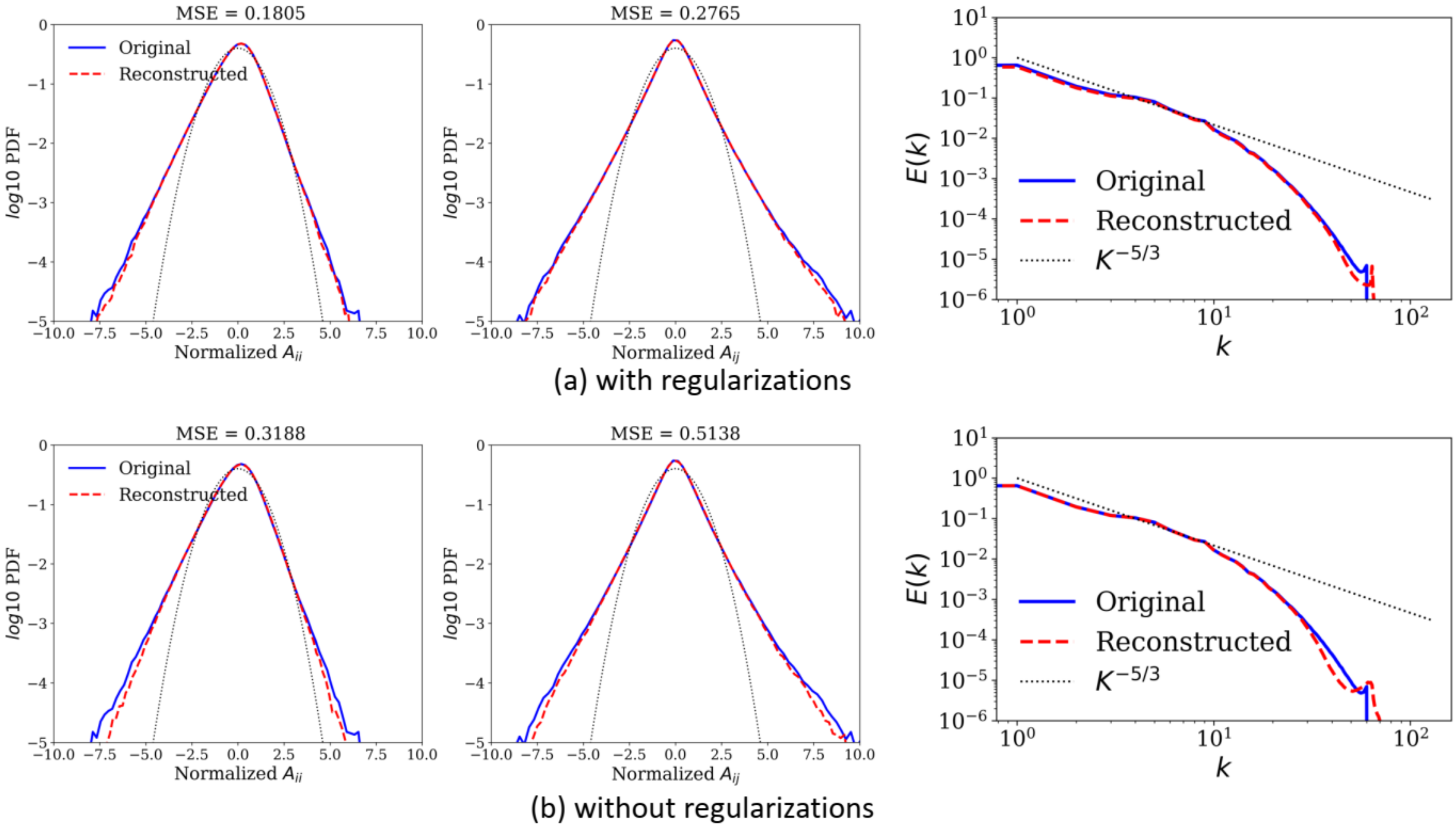


Figure 2. (a) with and (b) without regularizations for PDFs of normalized longitudinal (left), transverse (middle) components of velocity gradient tensor A, and Turbulence Kinetic Energy spectra (right) of stationary isotropic turbulence flow

Conclusion

- We propose a vector quantized deep learning framework, the so called vector quantized autoencoder or VQ-AE, for the compression of data from turbulent flow simulations
- We calibrate the loss function of the model to infuse prior physics-informed knowledge of the flow in the form of constraints in order to boost the model performance
- Our data compression framework is not limited to Computational Fluid Dynamics (CFD) simulations but can be easily applied to compress data from other complex physical simulations