Forecasting Variable-Density 3D Turbulent Flow

Xingyu Su[1], Robin Walters[2], Denis Aslangil[3], Rose Yu[4]

Zhejiang University, Northeastern University, Los Alamos National Laboratory, University of California San Diego

[1] su_xingyu@zju.edu, [2] rwalters@northeastern.edu, [3] denis.aslangil@gmail.com, [4] roseyu@ucsd.edu

Problem

How do we design deep learning models to forecast 3D turbulent flow when the flow is non-stationary and has variable density?

Our Contributions

- **Taylor-Net**: first deep learning method to predict 3D, variable-density, and non-stationary turbulent flows, by combining Taylor series approximation and U-net.
- Significant improvement in accuracy and physical consistency over competitive baselines on more general (e.g. non-stationary and anisotropic VD turbulence) fluid dynamics,
- Theoretical analysis of the interplay between forecasting horizon, step size, and the order of Taylor approximation.

Background

Limitation of State of the Art Current methods still rely on simplifying the assumptions in Navier-Stokes equations to make the problem more tractable, such as working on two-dimensional data, studying stationary flows, using low-resolution data, or asserting uniform density. These assumptions dramatically limit their applicability to real-world turbulent flows.

Variable Density Turbulence VD turbulence emerges from the mixing of two fluids with different densities. Using the Einstein summation convention, the governing equations of this physical system are:

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_j}{\partial x_j} = 0, \quad \frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \frac{\rho g_i}{Fr^2} \quad \text{with} \quad \frac{\partial u_i}{\partial x_i} = -\frac{1}{Re_0 Sc}\frac{\partial \rho}{\partial^2 x_i} \tag{1}$$

where ρ is the whole density field, u_i is the whole velocity field in direction i, p is the pressure, g_i is the gravity in direction i and the stress tensor is assumed Newtonian,

$$\tau_{ij} = (\rho/Re_0)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2\partial u_k}{3\partial x_k}\delta_{ij}\right).$$

The non-dimensional parameters are the computational Reynolds number, Re_0, Schmidt number, Sc, and Froude number, Fr, defined in [1].

Taylor-Net: Learning Taylor Series Remainder with Deep Nets

We employ a hybrid method which combines Taylor approximation and a U-net to learn the Taylor remainder. Note that our method does not explicitly use any part of the Navier-Stokes equations.

Taylor Approximation Given input data $x_{-t}, \dots, x_0$, we can interpolate the n-th order polynomial fit to the data where $n \le t$. Let $p_n(t) = a_0 + a_1 t + \dots a_n t^n$. The coefficients of p_n can be determined by the inverse of the Vandermonde matrix

$$\begin{pmatrix} x_{-n} \\ \vdots \\ x_0 \end{pmatrix} \begin{pmatrix} 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & n & \dots & n^n \end{pmatrix}^{-1} = \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix}.$$

Then the n-th order Taylor approximation is

$$\hat{x}_1 = T_n(x_{-n}, \dots, x_0) = p_n(1) = \sum_{i=-n}^{0} x_i \binom{n}{-i} (-1)^{i+1}. \tag{2}$$

For $n = 2$, this yields $x_{-2} - 3x_{-1} + 3x_0$ and for $n = 1$ this gives $-x_{-1} + 2x_0$.

Periodic Convolution Since the input and output spatial domain is periodic, we implement the layers of the U-net using periodic 3D convolutions. Consider input tensor x of size 64^3 and kernel ϕ of size $(2c+1)^3$. We index the kernel symmetrically about $(0, 0, 0)$. We define periodic convolution as

$$y_{m,n,p} = \sum_{i=-c,j=-c,k=-c}^{c,c,c} \phi_{i,j,k} x_{\overline{m-i},\overline{n-j},\overline{p-k}}$$

where we interpret the indices $\bar{i}$ of x modulo 64 . This is implemented using the "circular" padding mode of the PyTorch `conv3D` function.

Taylor-Net

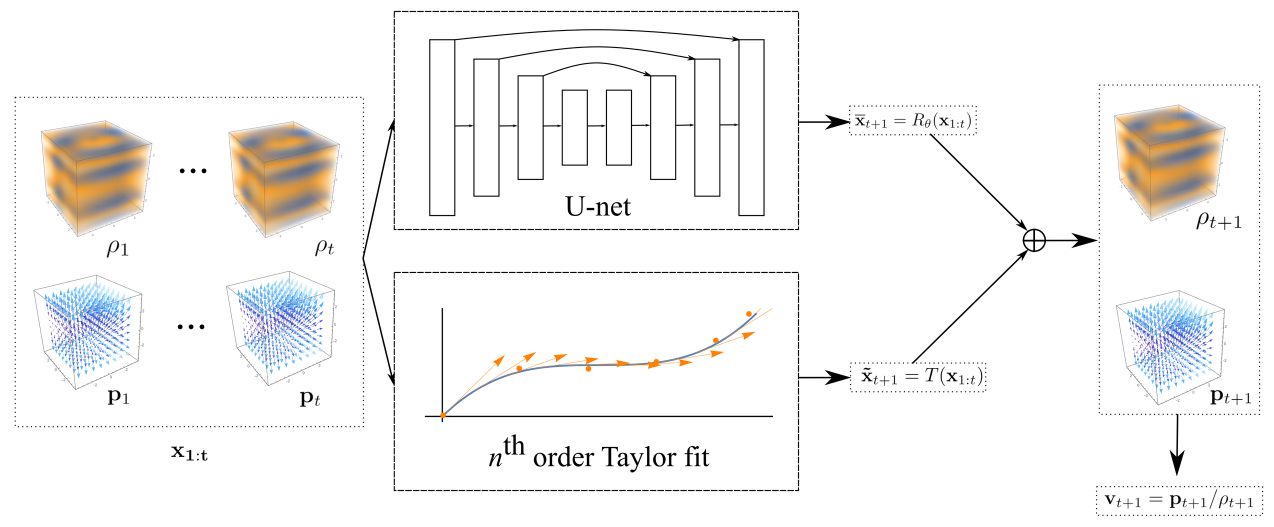

Figure: Diagram of the Taylor-Net-n method. First, the model computes the n^{th}-order polynomial fit, i.e., Taylor approximation, which it uses to approximate $\tilde{\mathbf{x}}_{t+1} = T(\mathbf{x}_{1:t})$. The U-net predicts the Taylor remainder $\overline{\mathbf{x}}_{t+1} = R_\theta(\mathbf{x}_{1:t})$. For the output we take the sum $(\rho_{t+1}, \mathbf{p}_{t+1}) = \mathbf{x}_{t+1} = \overline{\mathbf{x}}_{t+1} + \tilde{\mathbf{x}}_{t+1}$ from which we compute the velocity $\mathbf{v}_{t+1} = \mathbf{p}_{t+1}/\rho_{t+1}$.

Theoretical Analysis on Taylor Order Although higher order Taylor approximations achieve better fit over the short-term, they also diverge faster in the long term, a fact we see reflected in our experiments. Thus the optimal order of n represents a trade-off.

Proposition 1

Assume the true time series satisfy $|x_t| \le C$ for all t. For almost all values $x_{-n}, \dots, x_0$, the iterated n-th order Taylor progression $x_{i+1} = T_n(x_{i-n}, \dots, x_i)$ diverges and the rate of divergence is $\Theta(i^n)$.

Experiments

Experiment Setup We generate two datasets using HVDT Direct Numerical Simulation (DNS) in a triply periodic domain $[(2\pi)^3]$ with a 64^3 resolution. Our turbulent datasets contain flows with two different Atwood numbers.

We benchmark the performance of different methods w.r.t three metrics: Root Mean Square Error (RMSE), Energy Spectrum Error (ESE) and Mass Conservation. We compare with SoTA baselines for turbulent flow prediction including U-Net, TF-Net [3], and Fourier Neural Operator (FNO) [2].

Prediction Performance

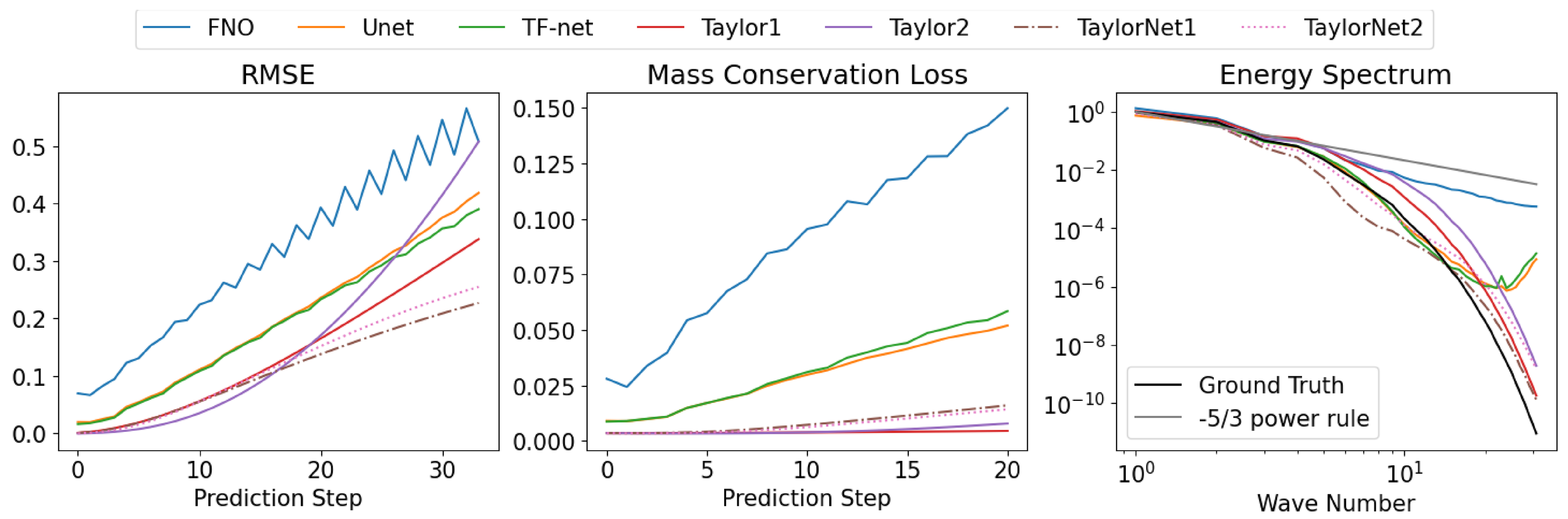

Figure: Performance comparison of our model versus several baselines across 3 metrics for $\mathbf{p}$ prediction: (a) RMSE (b) Mass-Conservation and (c) energy spectrum at the 20th prediction step. Our errors grow more slowly than others across the forecasting horizon. The flow predicted by our model has an energy spectrum closer to the truth.

Table: Parameter count and run time comparison of different models. Previous state-of-arts deep learning models consistently fail to outperform simple numerical approximations. Our combined numerical and deep learning method has the best performance and shows a great improvement in comparison with numerical approximations.

Models	DNS	Unet	TF-Net	FNO	TaylorNet
# params (1e7)	—	1.587	2.528	0.083	1.587
Runtime (s)	≈ 100	0.0407	0.0986	0.0478	0.0404

Experiments

Prediction Visualization

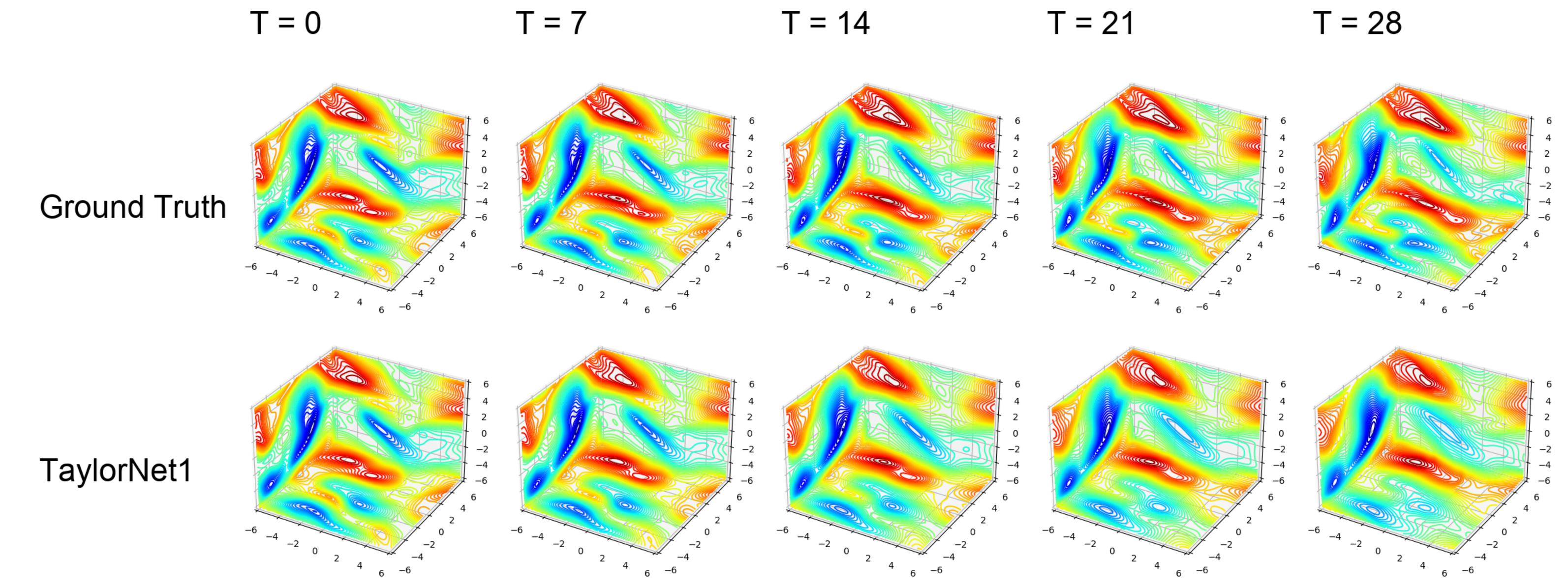

Figure: Prediction visualization for momentum field over 30 time steps. For each frame, we pick only the momentum field along x-axis and cast 3 surfaces of the data cube to xy, yz, zx planes. Comparison between Taylor-Net1 and ground truth shows our hybrid method can capture the small scale dynamics.

Ablation Study

Trade-off between Interval and Precision To understand why Taylor approximation improved the performance so significantly, we varied the input step size and found that higher order Taylor approximations are primarily helpful when the step size in numerical differentiation between inputs is small. As the step size between inputs increases, the additional benefit of higher order Taylor approximation decreases.

	U-net	Taylor 1	Taylor-Net1	Taylor 2	Taylor-Net2
RMSE $\Delta t = 1$ step	0.2713	0.03732	0.03036	0.01752	0.01548
RMSE $\Delta t = 2$ step	0.07167	0.04146	0.02603	0.02176	0.01691
RMSE $\Delta t = 4$ step	0.05096	0.06918	0.0232	0.03305	0.02656

Table: Interval vs Precision. Taylor n is n^{th} order Taylor approximation.

Zero Padding We perform an ablative study with periodic padding. Instead of periodic padding, we use zero padding. We find that the mass conservation is much worse.

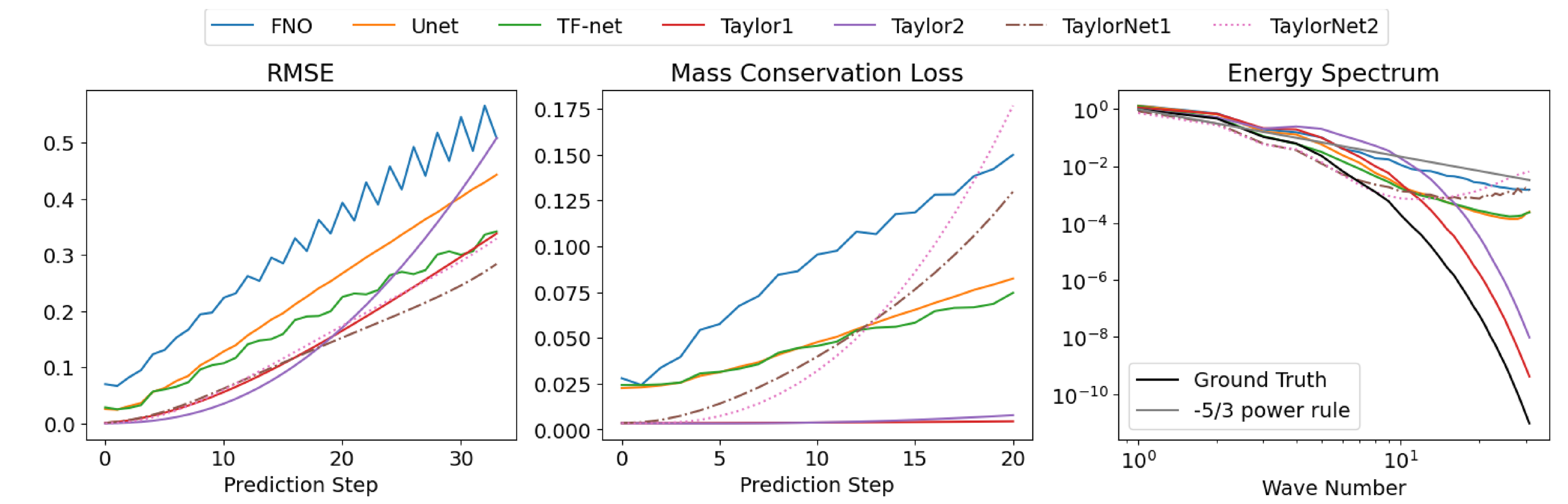

Figure: Comparison of our model with zero padding versus several baselines across 3 metrics for $\mathbf{p}$ prediction: RMSE, Mass Conservation, and Energy Spectrum.

References

[1] D. Aslangil, D. Livescu, and A. Banerjee. Effects of Atwood and Reynolds numbers on the evolution of buoyancy-driven homogeneous variable-density turbulence. *Journal of Fluid Mechanics*, 895:A12, 2020.

[2] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020.

[3] R. Wang, K. Kashinath, M. Mustafa, A. Albert, and R. Yu. Towards physics-informed deep learning for turbulent flow prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1457–1466, 2020.