Distribution shift in machine-learning augmented fluid simulation

Distribution shift review group

18th Sep, 2023

Content:

* Brief overview of machine-learning augmented fluid simulation

* Data generation and preprocessing

* Network model and simulation framework design

* Loss function and training algorithm

* Performance evaluation

Simulation framework

To be concrete and simple, suppose the task is given as

$$X_{t+\Delta t} = X_t + NN(X_t; \theta), \tag{1}$$

where $NN(X_t;\theta)$ is usually learned under supervision. The "rollout" trajectory produced by iteratively applying learned model for K steps is denoted $(\widehat{X}_0, \widehat{X}_1, \cdots, \widehat{X}_n)$, where $\widehat{X}_0 = X_0$ represents initial conditions given as input.

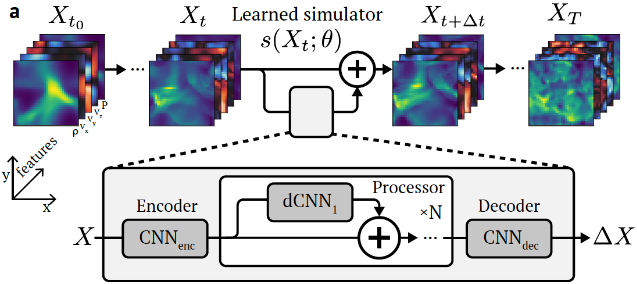


Figure: Framework

Difficulties in the fluid simulation

There are several outstanding difficulties in the fluid simulation:

* Intrinsic nature of chaotic dynamics: KS equation, isotropic turbulence, etc.

* High resolution grid results in extremely high dimension dynamics

* There does not exist a unified performance criterion.

^{JX}[After stating each of the difficulties, trying to answer whether any of them can be addressed with the help of machine learning, especially when you are talking about your own work]

Difficulties in the fluid simulation

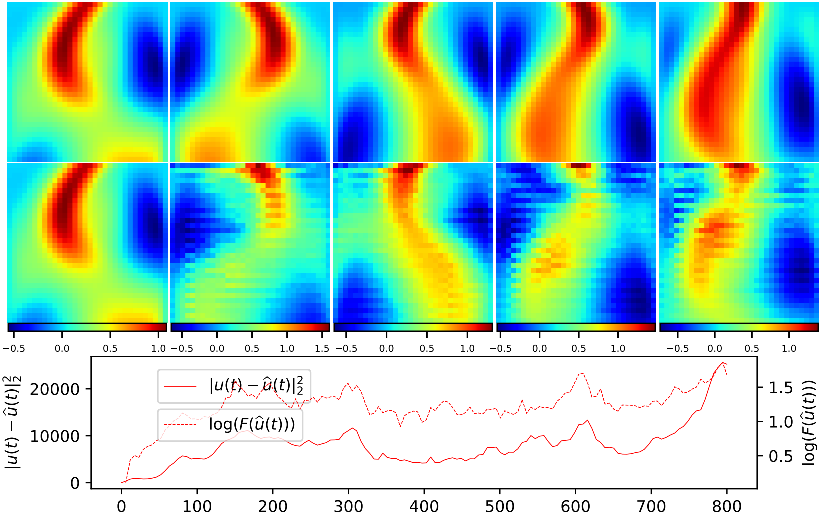


Figure: Difficulty of fluid simulation

Data generation

Current methodology, use a high-fidelity numerical solver to generate, including random initialized state, warmup, trajectory, labeling.

There are many “hyperparameters” that can be tuned to generate a SciML dataset:

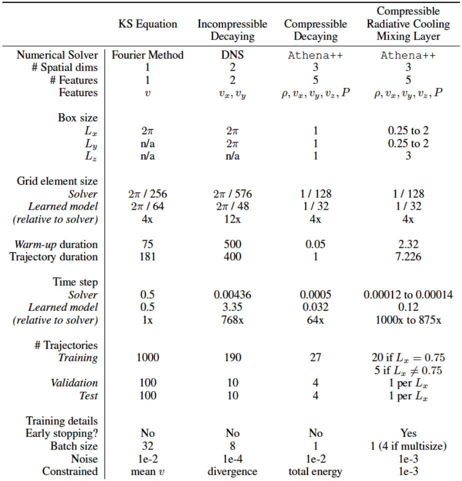
Initialization: sampled from some distribution.

Sampling: same as the time discretization of the numerical solver

Warmup time: truncated the first several iterations

^{JX}[Warmup and sampling are more or less related, e.g. considering the simulating of a dynamics with an invariant distribution. In Lin Bo's work where they try to learn the transition at the transition between two metastable states which corresponds to rare events, we indeed hope the data to be distributed uniformly instead of concentrating in certain area.]

Data generation



	KS Equation	Incompressible Decaying	Compressible Decaying	Compressible Radiative Cooling Mixing Layer
Numerical Solver	Fourier Method	DNS	`Athena++`	`Athena++`
# Spatial dims	1	2	3	3
# Features	1	2	5	5
Features	v	v_x, v_y	ρ, v_x, v_y, v_z, P	ρ, v_x, v_y, v_z, P
Box size				
L_x	2π	2π	1	0.25 to 2
L_y	n/a	2π	1	0.25 to 2
L_z	n/a	n/a	1	3
Grid element size				
Solver	2π / 256	2π / 576	1 / 128	1 / 128
Learned model	2π / 64	2π / 48	1 / 32	1 / 32
(relative to solver)	4x	12x	4x	4x
Warm-up duration	75	500	0.05	2.32
Trajectory duration	181	400	1	7.226
Time step				
Solver	0.5	0.00436	0.0005	0.00012 to 0.00014
Learned model	0.5	3.35	0.032	0.12
(relative to solver)	1x	768x	64x	1000x to 875x
# Trajectories				
Training	1000	190	27	20 if $L_x = 0.75$ 5 if $L_x \neq 0.75$
Validation	100	10	4	1 per L_x
Test	100	10	4	1 per L_x
Training details				
Early stopping?	No	No	No	Yes
Batch size	32	8	1	1 (4 if multisize)
Noise	1e-2	1e-4	1e-2	1e-3
Constrained	mean v	divergence	total energy	1e-3

Figure: Dataset details.

Data preprocessing

There are several methods, here we list three

* No preprocessing: physical information is important.

* Same as most statistical learning, entrywise preprocessing: resolve the issue of multiscale in different components.

* Similar to some CV task: group all the entries together.

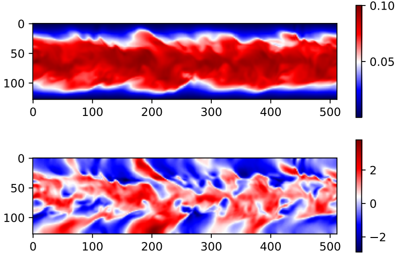
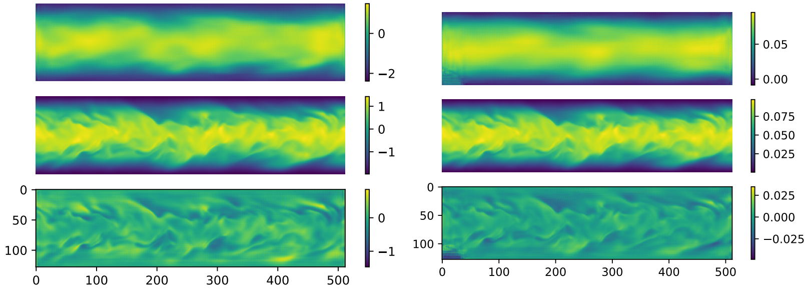


Figure: Comparison of data preprocessing: No preprocessing v.s. entrywise preprocessing

Data preprocessing



(a) Data with preprocessing

(b) Data without preprocessing

Figure: Comparison of data preprocessing

Data Preprocessing

Which of the following data form is most similar to dynamical fluid prediction?

* Image

* Video

* Sequence

Preprocessing should respect physical properties.

The effect of data generation and preprocessing on simulation results and distribution shift is seldomly studied.

Network architecture

The backbone for most network models is UNet

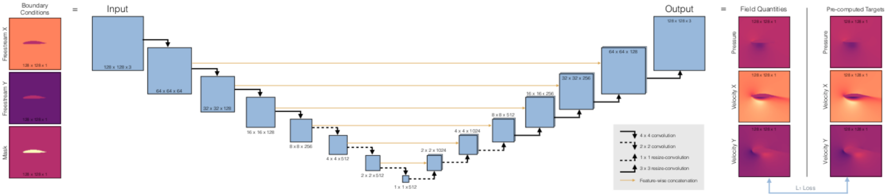


Figure: U-net structure for flow prediction[1]

It is similar to the autoencoder, performing non-linear dimension reduction. At the same time it could account for interaction at different scale.

There is large literature using GCN for irregular grid.

[1] Thuerey, Nils, et al. "Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows." AIAA Journal 58.1 (2020): 25-36.

Network architecture: TF-Net

The turbulent flow net[2]learns the temporal and spatial filter which follows the methodology of RANS-LES hybrid simulation.

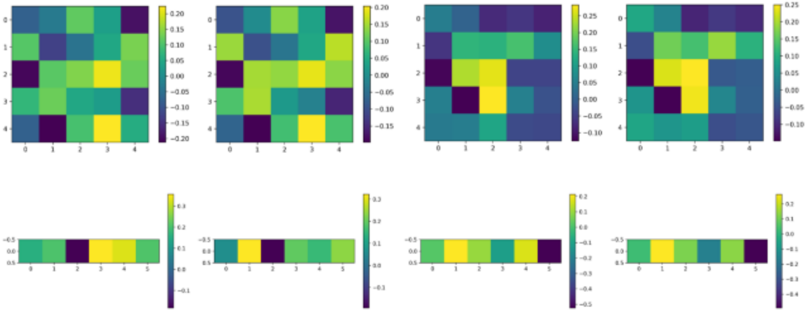


Figure: Learned spatial and temporal filter[3]

[3] Wang, Rui, et al. "Towards physics-informed deep learning for turbulent flow prediction." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020.

Simulation framework

Two mainstreams:

Viewing fluid prediction as an image task, using large CNN as parametrized model[4].

Using neural network to determine parameters for empirical model[5], more similar to the inverse problem and turbulence modeling.

[5] Stachenfeld, Kimberly, et al. "Learned coarse models for efficient turbulence simulation." arXiv preprint arXiv:2112.15275 (2021).

[5] Wang, Rui, et al. "Towards physics-informed deep learning for turbulent flow prediction." Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020.

Loss function

The basic choice is to train with one-step supervision under MSELoss.

$$l(\theta) = \mathbb{E}_{X\sim\rho} \left\| NN(X_t;\theta) - \Delta X_t \right\|_2^2 \tag{2}$$

Physics-informed constraints are used as regularization, such the divergence-free properties of the incompressible fluid simulations,

$$l(\theta) = \mathbb{E}_{X\sim\rho} \left\| NN(X_t;\theta) - \Delta X_t \right\|_2^2 + \left\| \nabla \cdot X_{t+\Delta t} \right\|_2^2 \tag{3}$$

In fluid mechanics community, sometimes the accuracy of fluid statistics such as energy spectrum is more important than the trajectorywise accuracy. If this becomes the ultimate goal, is there a notion of distribution shift?

Stability via adding noise

Stability issue: small errors can accumulate over rollouts and lead to a domain shift.

Add Gaussian noise to the input to robustify the learned models, which is similar to the adverserial training in CS literature. And the size of the noise serve a role in bias-variance trade-off.

$$\begin{aligned} X_t + \epsilon &\to \Delta X_t, \\ X_t + \epsilon &\to \Delta X_t + \epsilon. \end{aligned} \tag{4}$$

This is presumably because the training distribution has broader support and the model is optimized to map deviant inputs back to the training distribution. $^{\mathrm{JX}}$[Adding noise during the training makes more sense, otherwise there is no difference bewteen adding noise in simulating the trajectory, or simply including observation noise in collecting the trajectory data.]

Stability via adding noise

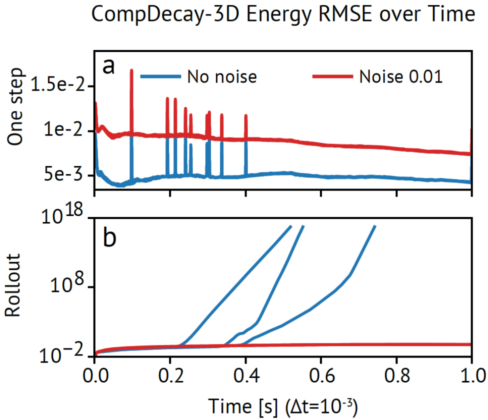


Figure: Effect of training with artificial noise[6]

[6] Stachenfeld, Kimberly, et al. "Learned coarse models for efficient turbulence simulation." arXiv preprint arXiv:2112.15275 (2021).

Stability via Temporal coarsening

An advantage of learned simulators is that they can exploit a much larger step size than the numerical solver, as they can discover efficient updates that capture relevant dynamics on larger timescale. This allows for faster simulation.

Choosing the timestep also has a trade-off: while larger time step has greater error each step, smaller step size has more iteration leading to unstability. ^{JX}[Think about if there is any similar phenomena in classical numerical analysis, Prof. Li suggests thinking about the conditional stability of the classical numerical schemes, e.g. FTCS is unconditionally unstable for transport equation, upwind scheme is conditionally stable, and many other similar criterion for numerical scheme of parabolic equation]

Stability via Temporal coarsening

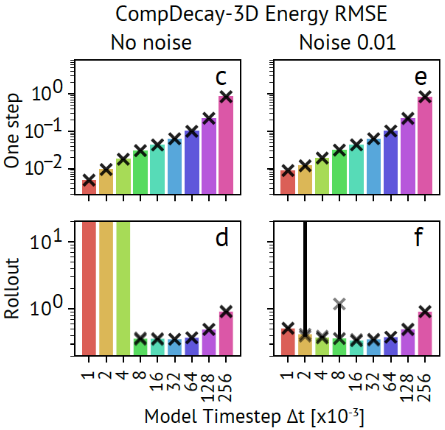


Figure: Effect of temporal coarsening[7]

[7] Stachenfeld, Kimberly, et al. "Learned coarse models for efficient turbulence simulation." arXiv preprint arXiv:2112.15275 (2021).

Performance evaluate

Accuracy is not the only thing we care about in fluid simulation. Sometimes they do not even care about accuracy as the dynamics is chaotic.

$$\int_{[0,T]} \left\| \widehat{X}_t - X_t \right\|_2^2 dt \tag{5}$$

Statistics & physical quantities: turbulent kinetic energy, energy spectrum

$$\text{TKE:} \quad \frac{\sum_0^T \left[(u_t - \bar{u})^2 + (v_t - \bar{v})^2 \right]}{2T}. \tag{6}$$

If we only care these physical quantities, is distribution shift still an issue here? If it is, how should we identify or quantify it?

JX [Here, distribution shift becomes something which is too special or narrow. The more general terminology we should use should be something like mismatch, which coule either represents the mismatch of the training and testing distribution or between the train loss function and the loss test function, e.g. train with L^2 while in inference we only cares about the behavior at the extreme value pts.]