Beyond Closure Models: Learning Chaotic-Systems via Physics-Informed Neural Operators

Chuwei Wang
Caltech

Joint work with Julius Berner, Zongyi Li, Di Zhou, Jiayun Wang, Jane Bae, and Anima Anandkumar

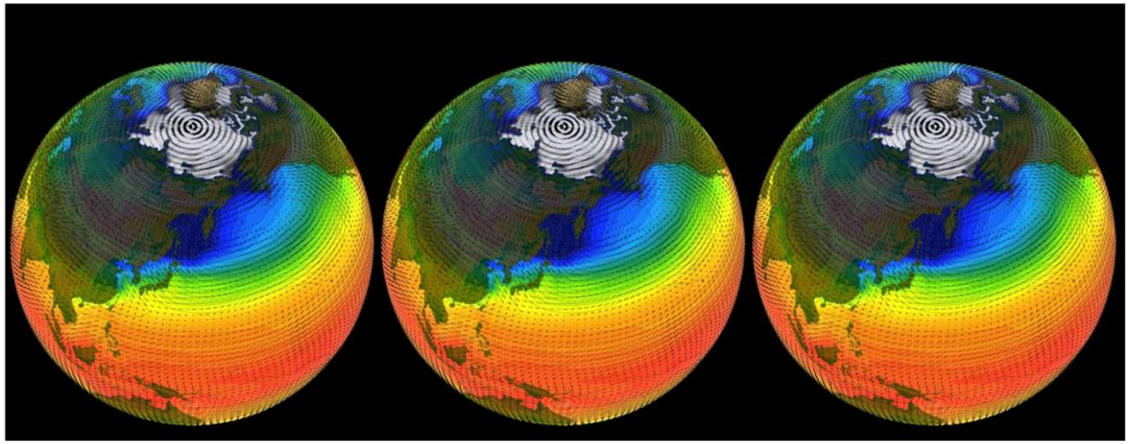
IAIFI Summer Workshop
2024.8.12

Chaotic Systems

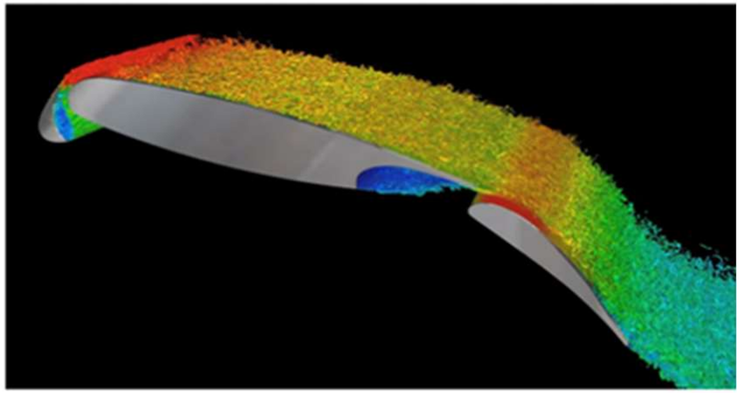
Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?

Small (numerical) error explodes along time!

- Long-term behavior/ statistics is of great practical importance in applications.



Climate Modeling



Aircraft Design

Chaotic Systems

Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?

Small (numerical) error explodes along time!

- Long-term behavior/ statistics is of great practical importance in applications.

➢ Method 1: Fully-Resolved Simulations (FRS): simulate with sufficiently small meshes/grids. **Too Expensive!**

➢ Method 2: Coarse-grid simulations (CGS) + further modification ('error-correction')

Chaotic Systems

Does the flap of a butterfly's wings in Brazil set off a tornado in Texas?

Small (numerical) error explodes along time!

- Long-term behavior/ statistics is of great practical importance in applications.

➢ Method 1: Fully-Resolved Simulations (FRS): simulate with sufficiently small meshes/grids. **Too Expensive!**

➢ Method 2: Coarse-grid simulations (CGS)+**further modification ('error-correction')**

$'\boldsymbol{Closure\ model}'$

Problem Setting

[Notations]

- A nonlinear (and chaotic) dynamics $\partial_t u = Lu, u \in H$
- S_t: the semigroup (of the true dynamics)
- 'Trajectory': $\{S_t u\}_{t \in R_{\geq 0}}$
- Attractor: $\Omega \subset H \ s.t. \lim_{t \to \infty} dist(S(t)u, \Omega) = 0$, $\forall u \in$ H.
- Invariant measure $\mu^* \coloneqq \lim_{T \to \infty} \frac{1}{T} \int_0^T \delta_{S(t)u} \, dt$, (independent of the initial u)
    - Measure in function space
    - Supported on Ω
- **Statistics**: For a functional O, the stat $\langle O \rangle \coloneqq E_{u \sim \mu^*} O(u) = \int O(u) \mu^*(du)$

The Goal of this Project.

Outline

- (1) Review: data-driven closure models (& their inherent shortcomings)
- (2) New insight through Measure flow in function space
- (3) Our approach: Physics-informed Neural Operator

Coarse-grid Simulations

- Filtering operator $F: u \to \bar{u}$, e.g. spatial down sampling, Fourier-mode truncation.
  (Simulations with coarse grids)

Nonlinear dynamics: $\partial_t u = Lu, u \in H$ (H: function space of interest)

➢ Dynamics for $\bar{u}$: $\partial_t \bar{u} = L\bar{u} + (FL - LF)\boldsymbol{u}$.

(nonlinear system F and L does not commute)

➢ Simulating on low-res grids (the space of $F(H)$) no access to $\boldsymbol{u}$.

➢ [**Closure Model**]: Evolve $\partial_t \bar{u} = L\bar{u} + clos(\bar{u}; \theta)$

Existing Data-driven Closure Models

- [**CGS**]: Evolve $\partial_t u = Lu + clos(u;\theta)$

$\boldsymbol{clos}: H(R^3) \to H(R^3)$

Learned Model:

(1)Fix an ansatz $clos(u;\theta)$

(2) Supervised loss: $\left|\left|(FL - LF)\boldsymbol{u} - clos(F\boldsymbol{u};\theta)\right|\right|$

(3*) More complex version:

Training data $\boldsymbol{u}$: comes from costly Fully-Resolved simulations

Existing Data-driven Closure Models

- [CGS]: Evolve $\partial_t u = Lu + clos(u;\theta)$

$$\boldsymbol{clos}: H(R^3) \to H(R^3)$$

Learned Model:

(1)Fix an ansatz $clos(u;\theta)$

(2) Supervised loss: $||(FL - LF)\boldsymbol{u} - clos(F\boldsymbol{u};\theta)||$

Training data $\boldsymbol{u}$: comes from costly Fully-Resolved simulations

(3*) More complex version:

History-aware models:

$$\boldsymbol{clos}: H(R^3 \times [0,t')) \to H(R^3)$$

$$u(x,t), x \in D, t \in [t_0 - t', t_0) \to clos(u)(x,t_0)$$

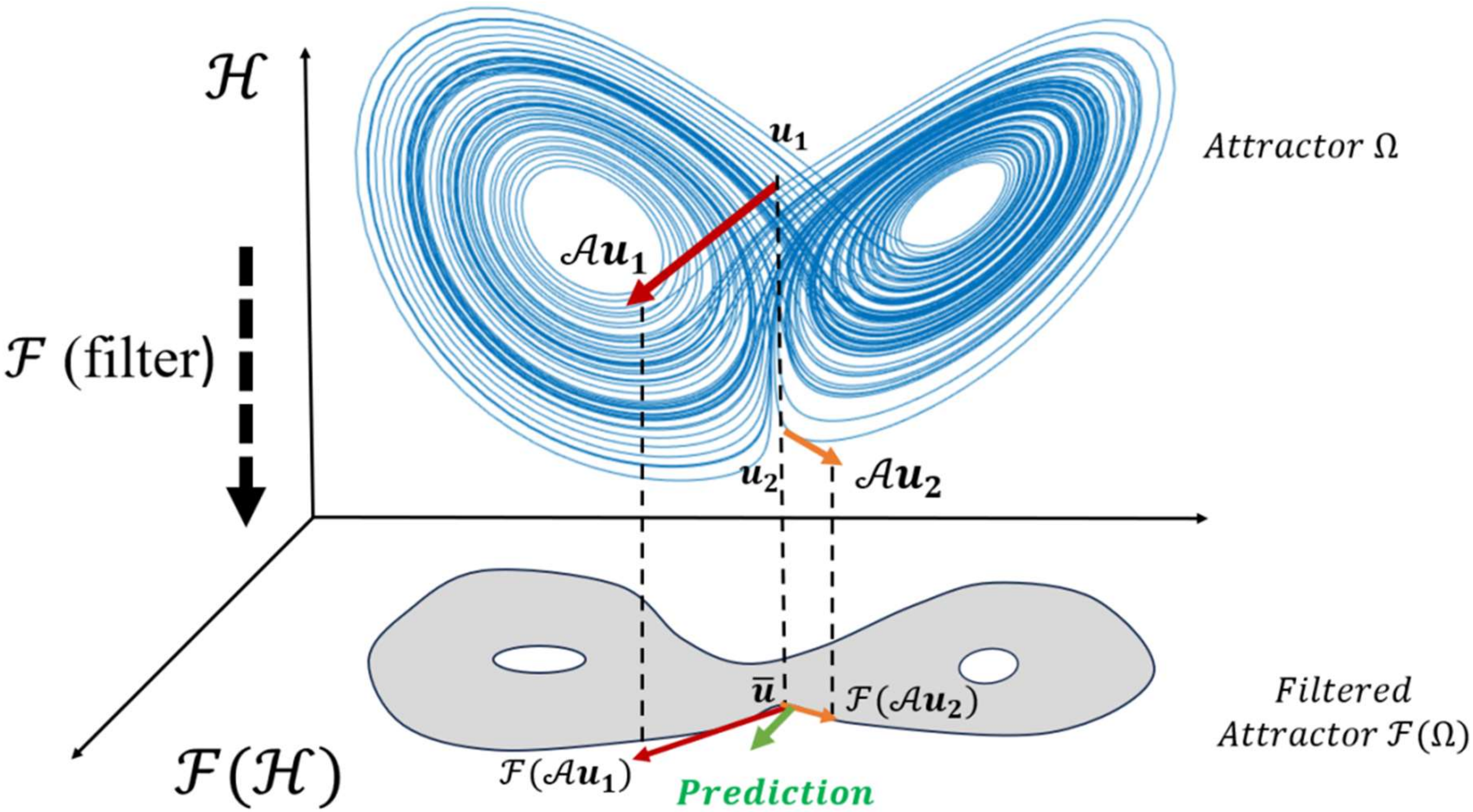
Stochastic models

Different loss functions

➢[**Closure Model**]: Evolve $\partial_t \bar{u} = L\bar{u} + clos(\bar{u};\theta))$

- [**Result 1**] The target mapping $clos(u)$ **is not well-defined** for all types of ansatz in the literature. (There are multiple possible outputs for the same input.)
- [**Result 2**] Following this scheme (directly adding a closure term), the model **has to** be trained with a large amount of fully-resolved data that suffices to compute the statistics.

$\Longrightarrow$ We no longer need such a model!



(1) This is not a well-defined mapping.

(2) The resulting coarse-grid simulation might deviate from the filtered attractor.

(And the performance highly depends on the training data)

! We can only assign one moving direction in the reduced space *F(H)*.

Theoretical Intuition

- [**FRS**(ground truth)] $\partial_t u = Lu$
- [**CGS**]: Evolve $\partial_t u = Lu + clos(u;\theta)$
  - Functions(states) u : infinite-dimensional particle systems in H.
  - Only need to care about distributions (the invariant measure)

    Liouville flow/ Fokker-Planck equations for measure transformations.

    $\partial_t \rho = -\nabla \cdot (\boldsymbol{f}\rho)$
  - The invariant measure is the solution to stationary Liouville/Fokker-Planck equation!

Outline

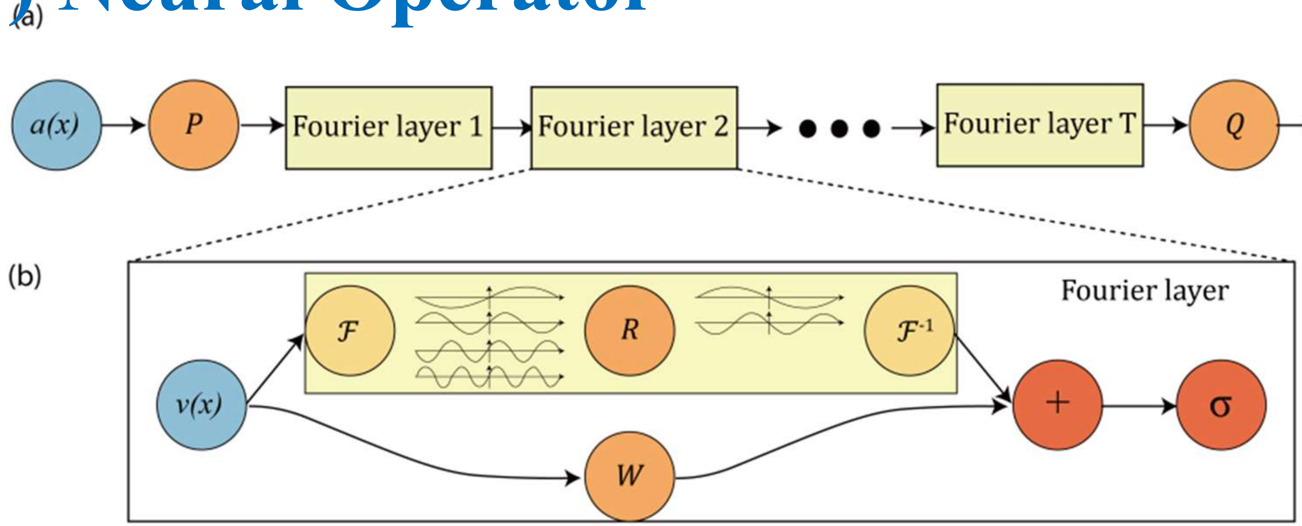
- (1) Review: data-driven closure models (& their shortcomings)
- (2) Insight through Measure flow in function space
- (3) Our approach: Physics-informed Neural Operator

➢[**Previous Scheme**]: Evolve $\partial_t \bar{u} = L\bar{u} + clos(\bar{u}; \theta)$

Why do we have to adopt this form?

- New scheme: Evolve $\partial_t \bar{u} = \boldsymbol{L(\theta)}\bar{u}$
- Intuition: the error-correcting term is implicitly involved in the simulation.

(Fourier) Neural Operator



(a) The full architecture of neural operator: start from input a. 1. Lift to a higher dimension channel space by a neural network P. 2. Apply four layers of integral operators and activation functions. 3. Project back to the target dimension by a neural network Q. Output u. **(b) Fourier layers**: Start from input v. On top: apply the Fourier transform $\mathcal{F}$; a linear transform R on the lower Fourier modes and filters out the higher modes; then apply the inverse Fourier transform $\mathcal{F}^{-1}$. On the bottom: apply a local linear transform W.

Figure 2: **top:** The architecture of the neural operators; **bottom:** Fourier layer.

$$\mathcal{G}_{FNO} := \mathcal{Q} \circ (W_L + \mathcal{K}_L) \circ \cdots \circ \sigma(W_1 + \mathcal{K}_1) \circ \mathcal{P},$$

The input can be either course-grid or fine-grid.

- New scheme: Evolve $\partial_t \bar{u} = \boldsymbol{L(\theta)}\bar{u}$
- Intuition: the error-correcting term is implicitly involved in the simulation.

[Method]

- Physics-informed Neural Operator

Much fewer DNS data needed

Ansatz for $L(\theta)$
Merit: resolution invarianc

Physics-informed loss function:

$$J_{pde}(\theta; \mathfrak{D}) = \frac{1}{|\mathfrak{D}|} \sum_{i \in \mathfrak{D}} \|(\partial_t - \mathcal{A})\mathcal{G}_\theta u_{0i}(x)\|_{L^2(\Omega \times [0,h])}$$

- Provable convergence guarantee for statistics.

Theorem 3.1. *For any $h > 0$, denote $\hat{\mu}_{h,\theta} := \lim_{N\to\infty} \frac{1}{N} \sum_{n=1}^{N} \delta_{\mathcal{G}_\theta^n v_0(x)}$, any $v_0(x)$ with $x \in D'$. For any $\epsilon > 0$, there exists $\delta > 0$ s.t. as long as $\|(\mathcal{G}_\theta u)(\cdot, h) - S(h)u\|_{\mathcal{H}} < \delta, \forall u \in \mathcal{H}$, we have $\mathcal{W}_{\mathcal{H}}(\hat{\mu}_{h,\theta}, \mathcal{F}_{\#}\mu^*) < \epsilon$, where $\mathcal{W}_{\mathcal{H}}$ is a generalization of Wasserstein distance in function space.*

Summary

Method	Optimal statistics	High-res. training data DNS Snapshots	Trajs.	Complexity
Fully-resolved Simulation, e.g., DNS [33, 34]	✓	-		$Re^{3.52}$
Coarse-grid Simulation, e.g., LES [33, 34]	✗	-		$Re^{2.48}$
Single-state model [28]	✗	24000	8	$Re^{2.48}$
History-aware model[35]	✗	250000	50	$Re^{2.48}$
Latent Neural SDE[32]	✗	179200	28	$\frac{1}{\delta t} Re^{1.86}$
Physics-Informed Operator Learning (Ours)	✓	110	1	$Re^{1.86}$

Re: Reynolds number

1. Require large number of DNS data (which are not available usually).
2. Require a coarse-grid solver (can be even faster).
3. Cannot give the optimal estimations of statistics in ideal case, i.e. perfect training.

1D Kuramoto–Sivashinsky (toy test example)

$$\partial_t u + u\partial_x u + \partial_{xx} u + \nu \partial_{xxxx} u = 0, \qquad (x,t) \in [0, 6\pi] \times \mathbb{R}_+,$$

- Viscosity: 0.01 L: 6π
- DNS: 1024 spatial grid
- LES: 128 spatial grid
- Note: baseline methods are trained with the same number of DNS data as ours.

Method	Avg. Eng.	Max Eng.	Avg. Cor.	Max Cor.	Velocity	Avg. TV	Max TV
CGS (No closure)	12.5169%	77.8223%	13.1275%	80.5793%	0.0282	0.0398	0.2097
Eddy-Viscosity [57]	7.6400%	48.3684%	8.7583%	56.5878%	**0.0276**	0.0282	0.1462
Single-state [28]	12.5323%	78.6410%	13.1052%	81.2461%	0.0280	0.0410	0.2111
Our Method	**7.4776%**	**20.4176%**	**7.8706%**	**22.7046%**	0.0284	**0.0272**	**0.0849**

Energy Spectrum Correlation Velocity Total Variation of each mode

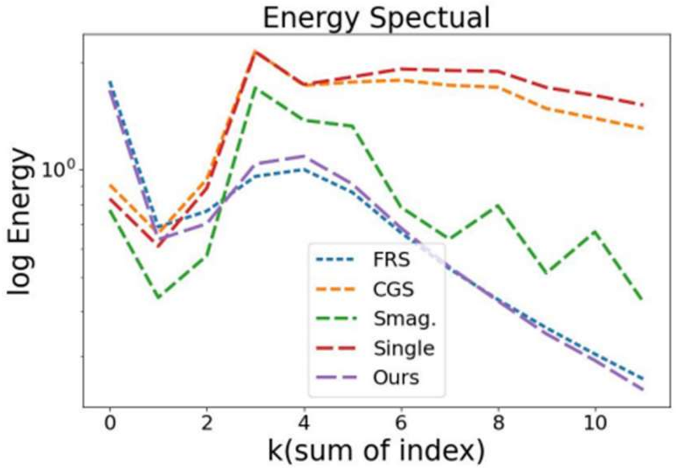
2D Kolmogorov Flow (toy test example)

$$\partial_t \mathbf{u} = -(\mathbf{u}\cdot\nabla)\mathbf{u} - \nabla p + \frac{1}{Re}\Delta\mathbf{u} + (\sin(4y),0)^T, \quad \nabla\cdot\mathbf{u} = 0, \quad (x,y,t) \in [0,L]^2 \times \mathbb{R}_+,$$

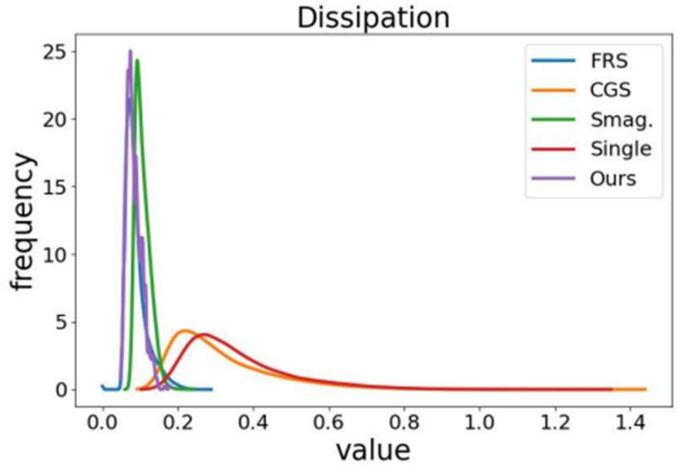
- Reynolds number: 100   L: 2π
- DNS: 128*128 spatial grid
- LES: 16*16 spatial grid
- Note: baseline methods are trained with the same number of DNS data as ours.

Method	Avg. Eng.	Max Eng.	Vorticity	Avg. TV	Max TV	Variance
CGS (No closure)	178.4651%	404.9923%	0.1512	0.4914	0.8367	253.4234%
Smagorinsky [14]	52.9511%	120.0723%	0.0483	0.2423	0.9195	20.1740%
Single-state [28]	205.3709%	487.3957%	0.1648	0.5137	0.8490	298.2027%
Our Method	**5.3276%**	**8.9188%**	**0.0091**	**0.0726**	**0.2572**	**2.8666%**

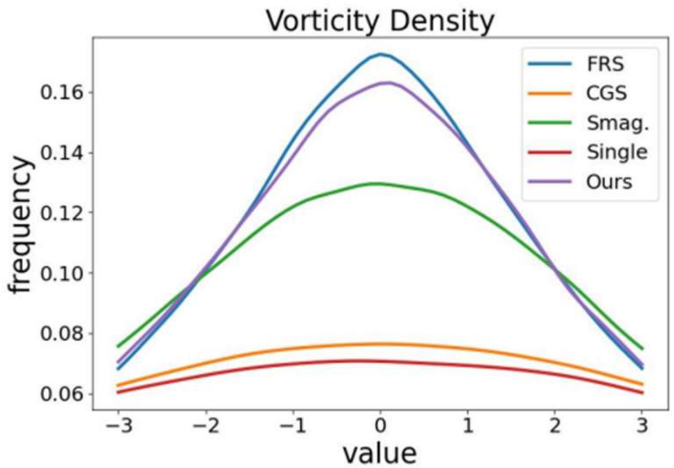
Energy Spectrum   Vorticity   Total Variation of each mode



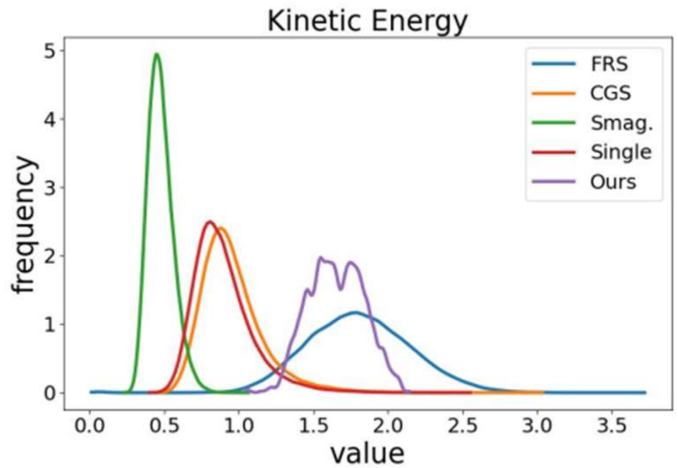
(a) Energy Spectrum



(b) Dissipation Distribution



(c) Vorticity Distribution



(d) Kinetic Energy Distribution

Summary

Takeaway: We need to ensure a well-defined target mapping before any ‘learning’.

See more details in https://arxiv.org/pdf/2408.05177

Thanks!