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

Chuwei Wang
MURI Meeting
2024.7.2

Outline

- (0) Overview
- (1) Review: previous methods
- (2) Problem reformulation through functional Liouville flow
  Shortcomings of existing methods
- (3) Physics-Informed Neural Operator(PINO)
- (4) Results and Discussion

Problem Setup

- Nonlinear dynamics: $\partial_t u = Lu, u \in H$ (H: function space of interest)
- **[Goal]**: Long-term statistics of this dynamics.

- **Chaotic system** ⟹ **DNS (Direct numerical simulations): too expensive!**
- Filtering operator $F: u \to \bar{u}$, e.g. spatial down sampling, Fourier-mode truncation. (Simulations with coarse grids)

➢ 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 u.

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

Closure Model

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

Why do we have to adopt this form?

(An explicit decomposition of 'error term' and the equation)

Is it reasonable or good enough?

How to design these models?

Main Results

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

Why do we have to adopt this form?

Is it reasonable or good enough?

- 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.)

How to design these models?

- Handcraft models: '*more of an art than a science*'
- Data driven models: (We prove that) have to be trained with a large number of DNS data that suffices to compute the statistics.

→ We no longer need such a model!

Main Results (cont'd)

➢[**Closure Model**]: 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.

[Method]

- Physics-informed Neural Operator

Much fewer DNS data needed

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

- Provable convergence guarantee for statistics.

Outline

- (0) Overview
- (1) Review: previous methods
- (2) Problem reformulation through functional Liouville flow

  Shortcomings of existing methods
- (3) Physics-Informed Neural Operator(PINO)
- (4) Results and Discussion

Basic Assumptions in this work: existence and uniqueness of the attractor.

Numerical Methods

- LES (Large eddy simulation)

Data-Driven Methods

- Learning (implicit) closure term/error of model
- Via conditional sampling (and Diffusion Model)
- FNO

Preview

Numerical Methods

- LES (Large eddy simulation)

Data-Driven Methods

- Learning (implicit) closure term/error of model
- Via conditional sampling (and Diffusion Model)
- FNO

Evolve with small Δt

Need a large number of DNS traj for training

Our Method (PINO)

Could never achieve optimal estimation even with perfect training/optimization.

LES

D: num of DNS grids   d: num of LES grids

- The nonlinear dynamics $\partial_t \boldsymbol{u} = L\boldsymbol{u}, \boldsymbol{u} \in H$ (In following discussions, we ignore the difference between DNS and true trajectory)
- Evolve on low-res grid: outcome of a linear filter $F: H(\text{or } R^D) \to R^d: \boldsymbol{u} \to \bar{u}$
- Alternative understanding: Filter operator in function space: $H \to H$. (Intuitive example: spectral method and Fourier truncation)

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

  (F and L does not commute due to nonlinearity)
- Since we are simulating on low-res grids (the space of $F(H)$), we don't have access to $\boldsymbol{u}$.
- [**LES**]: Evolve $\partial_t u = Lu + clos(u;\theta)$

$$'Closure\ model'$$

Closure model

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

Handcraft Model:

Sub-grid Stress (SGS) model (at most 11parameter),

e.g. Smagorinsky model for Navier Stokes

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:

Closure model

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

$clos: H(R^3) \to H(R^3)$

Handcraft Model:

Sub-grid Stress (SGS) model (at most 11parameter)

Learned Model:

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

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

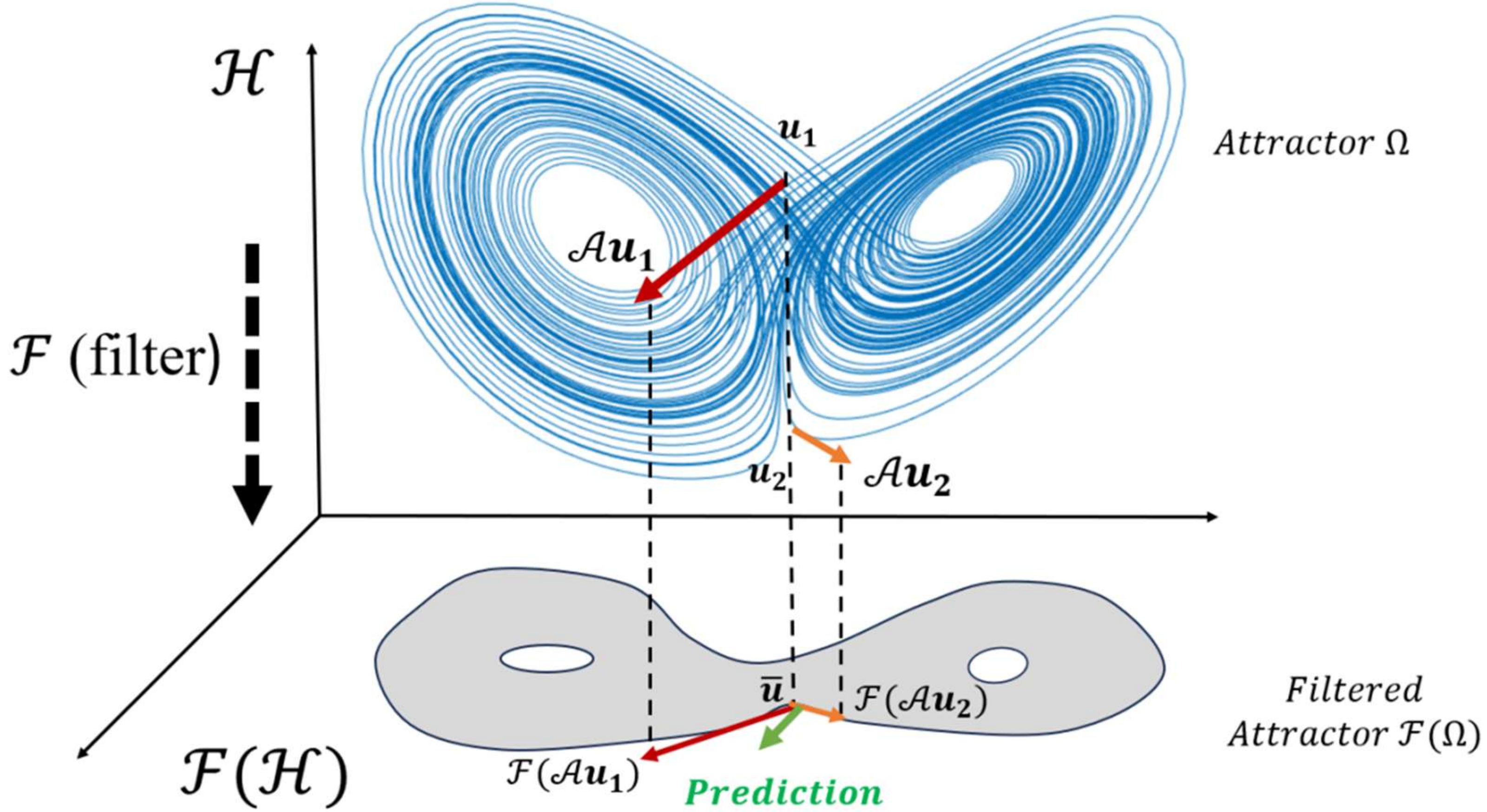
(3*) More complex version: (history information)

Better expressive power;

Relies on solver to start (for$[0,t']$)

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

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



(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)*.

If restricted to the subspace, deterministic vector filed(closure model) is weird!

(3*) More complex version: (history information)

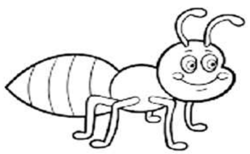
$$\boldsymbol{clos}: H(R^3 \times [0, t')) \rightarrow H(R^3)$$
$$u(x,t), x \in D, t \in [t_0 - t', t_0) \rightarrow clos(u)(x, t_0)$$

Would historical information helps?

[Thm] For any finite $\mathrm{t}_0, \mathrm{T} > 0$, any initialization $u \in H$, there exists infinite instances of $v \in H$, such that $F(S_t v) = F(S_t u)$, $\forall t \in [t_0, T]$.

S_t: semigroup of the dynamics, $u_0(x) \rightarrow u(t, x)$

$\{\boldsymbol{F}(\boldsymbol{S}_t \boldsymbol{u})\}_{t<t'}$ are all the information I could use to decide next step's direction at moment t' !



Optimal LES

From an probabilistic viewpoint:

- Decompose the fluid field into resolved part(in coarse grid) and unresolved part:

  e.g. $R^D = R^d \times R^{D-d}$ or $H = F(H) \bigoplus F(H)^{\perp}$: $\boldsymbol{u} = u + v$
- The distribution of $\boldsymbol{u}$ a joint distribution of (u, v).
- In LES setting, given u, what is the best choice for vector field/closure term?
- ***Conditional Expectation!***
- **The optimal closure model is** $E[L(\boldsymbol{u})|u]$

(**) Expectation w.r.t. which distribution?

Optimal LES formulations for isotropic turbulence, 1998

Neural Ideal Large Eddy Simulation: Modeling Turbulence with Neural Stochastic Differential Equations

Neurips 2023

The evolution of the ideal LES field v is obtained from the time derivatives of the set of unfiltered turbulent fields whose large scale features are the same as v [48]:

$$\frac{\partial v}{\partial t} = \mathbb{E}_{\pi_t}\left[\overline{\frac{\partial u}{\partial t}} \,\middle|\, \overline{u} = v \right] \tag{7}$$

Idea: Train a generative model (via an SDE in latent space)

1. Given u, sample $v, or\ equivalently,\ \boldsymbol{u}$.
2. Compute conditional expectation.

FNO

Fourier neural operator for large eddy simulation of compressible Rayleigh-Taylor turbulence, 2024

- Leverage the resolution-invariant property of FNO to capture motions in unresolved space (high Fourier modes)
- **[Drawbacks]**
- FNO is trained through supervised learning and is thus vulnerable to distribution shift.
- Training material in practice: solver data evolved on coarse grid.
- Training data in the paper: down sampling of training data from fine grid.

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.

Outline

- (0) Overview
- (1) Review: previous methods
- (2) Problem reformulation through functional Liouville flow

  Shortcomings of existing methods
- (3) Physics-Informed Neural Operator(PINO)
- (4) Results and Discussion

What is our goal?

- Obtain good estimations of long-term statistics with simulations only conducted on coarse grids.

Let's make it precise!

- 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 $\langle O \rangle \coloneqq E_{u \sim \mu^*} O(u) = \int O(u) \mu^*(du)$

What is invariant measure μ^*?

Stationary solution of a Liouville equation (in function space).

What is the best approximation of μ^* with finite grid?

Pushforward of μ^* of the filtering (informal).

What are previous methods doing?

Assign a vector field in the reduced space F(H), which always have the form of a conditional expectation.

Previous method can not give good result in good estimations of statistics unless the model is trained with a large number of DNS data with which one can actually directly estimate the statistics. (informal)

Functional Liouville (Measure) Flow

- $H = F(H) \oplus F(H)^{\perp}$: $\boldsymbol{u} = u + v$
- ONB $\{\psi_i\}$: $F(H) = span\{\psi_i : i \leq d\}$
- Canonical isometric isomorphism T: $\boldsymbol{u} \leftrightarrow c \in R^{\infty} \cap \ell^2$: $\boldsymbol{u} = \sum_i c_i \psi_i$.
- Denote $\hat{u} \coloneqq (c_1, c_2, \dots c_d), \hat{v} \coloneqq (c_{d+1}, \dots .)$.
- The nonlinear dynamics becomes: $\frac{dc}{dt} = b(c), b$ is an inf-dim vector field (function).
- **Example**: ψ_i: Fourier basis (or sin & cos to avoid complex number)

KS: $\partial_t u + uDu + D^2 u + D^4 u = 0,\ D \coloneqq \partial_x, b(c)_k = (-k^4 + k^2)c_k - ik \sum_{j+l=k} c_j c_l$.

- We start with 'particles' $\boldsymbol{u} \sim T^{\#} p_0(c)$.
- Each particle evolves and generates a traj $\boldsymbol{u}(t)$.
- $p(c, t)$: the prob density of $\boldsymbol{u}(t)$
- [**Liouville Eqn**] $\partial_t p(c, t) = -\nabla_c \cdot \big(b(c) p(c, t)\big)$; $p(c, 0) = p_0(c)$
- (Denoted as $\partial_t p = \mathcal{L} p$)

Invariant Measure

- Invariant measure: (Cesaro) limit of $p(c,t)$: $\boldsymbol{p}(c,t) := \frac{1}{t}\int_0^t p(c,s)ds$.
- Eqn for $\boldsymbol{p}$: $\partial_t \boldsymbol{p} = \mathcal{L}\boldsymbol{p} + \frac{1}{t}(p(t) - p_0)$
- The density of μ^* is the solution to $\mathcal{L}\boldsymbol{p} = 0$! (Denoted as p^*)

  (usually in weak sense)

Best Approximation in the Filtered Space

Def $P: H \to F(H)$ the orthogonal projection. (might be different from F).

[Thm] $P_{\#}\mu^* = \text{argmin}_{\gamma \in \mathcal{P}(K)} \mathcal{W}_2(\gamma, \mu^*)$.

The functional Wasserstein distance is defined with cost function $||u_1 - u_2||_H^2$.

Note: $P_{\#}\mu^*$ can also be viewed as the marginal distribution of μ^* in first d- dim.

Let's write out the Eqn! (for $P^{\#}p(c, t)$)

- $p_1(\hat{u}, t) := P_{\#}p(c, t)$
- $\partial_t p_1(\hat{u}, t) = -\nabla_{\hat{u}} \cdot \left\{ p_1(\hat{u}, t) \int \left[b_1(\hat{u}, \hat{v}) \frac{p(\hat{u}, \hat{v}, t)}{\int p(\hat{u}, \hat{v}, t) d\hat{v}} \right] d\hat{v} \right\}$

  $= -\nabla_{\hat{u}} \cdot \left(p_1(t) \boldsymbol{E}_{\hat{v} \sim p(\hat{v}|\hat{u}, t)} [b_1(\hat{u}, \hat{v}) | \hat{u}] \right)$
- $\frac{d\hat{u}}{dt} = \boldsymbol{E}_{\hat{v} \sim p(\hat{v}|\hat{u}, t)} [b_1(\hat{u}, \hat{v}) | \hat{u}]$ (The 'optimal' dynamics in K).

$b_1(\hat{u}, \hat{v})$: the first d-dim of b.

Optimal LES

From an probabilistic viewpoint:

- Decompose the fluid field into resolved part(in coarse grid) and unresolved part:

  e.g. $R^D = R^d \times R^{D-d}$ or $H = F(H) \oplus F(H)^{\perp}$: $\boldsymbol{u} = u + v$
- The distribution of $\boldsymbol{u}$ a joint distribution of (u, v).
- In LES setting, given u, what is the best choice for vector field/closure term?
- ***Conditional Expectation!***
- **The optimal closure model is** $E[L(\boldsymbol{u})|u]$

(**) Expectation w.r.t. which distribution?

Let's write out the Eqn! (for $P^{\#}p(c,t)$)

- $p_1(\hat{u},t) \coloneqq P^{\#}p(c,t)$
- $\partial_t p_1(\hat{u},t) = -\nabla_{\hat{u}} \cdot \left\{ p_1(\hat{u},t) \int \left[b_1(\hat{u},\hat{v}) \frac{p(\hat{u},\hat{v},t)}{\int p(\hat{u},\hat{v},t)\, d\hat{v}} \right] d\hat{v} \right\}$

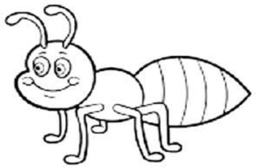
$$= -\nabla_{\hat{u}} \cdot \left(p_1(t) \boldsymbol{E}_{\hat{v} \sim p(\hat{v}|\hat{u},\boldsymbol{t})} [b_1(\hat{u},\hat{v}) | \hat{u}] \right)$$

- $\frac{d\hat{u}}{dt} = \boldsymbol{E}_{\hat{v} \sim p(\hat{v}|\hat{u},\boldsymbol{t})} [b_1(\hat{u},\hat{v}) | \hat{u}]$ (The 'optimal' dynamics in K).

It is irresolvable!

The $p(\hat{v}|\hat{u},\boldsymbol{t})$ depends on t and initial condition/density, and an (evolving) distribution in $F(H)^{\perp}$, but at the point $T\hat{u} \in F(H)$, or more generally $(T\hat{u}, t_0) \in K \times [0,\infty)$, I don't know what is happening in $K^{\perp}$ and what happened t' seconds ago!

The only performable action is to compute conditional expectation w.r.t an **fixed** distribution $q(c) \in \mathcal{P}(H)$.



Let's write out the Eqn! (for $\boldsymbol{P^{\#}p(c,t)}$)

The only performable action is to compute conditional expectation w.r.t an **fixed** distribution $q(c) \in \mathcal{P}(H)$.

$\frac{d\hat{u}}{dt} = \boldsymbol{E}_{\hat{v}\sim \mathrm{q}(\hat{v}|\hat{u})}[b_1(\hat{u},\hat{v})|\hat{u}]$ (The surrogate dynamics in K).

Write out the Liouville equation in reduced space K: $\partial_t p_1(\hat{u},t) = \hat{\mathcal{L}}_q p_1(\hat{u},t)$

Limit distribution: $\hat{\mathcal{L}}_q p_1 = 0$

[**Basic Requirement: Consistency**]: $\hat{\mathcal{L}}_q p_1^* = 0$. (Faithfully recover the optimal measure)

The choice for $q(c)$: $p^*(c)$.

- **The ideal and realistic closure** model**:** $\frac{d\hat{u}}{dt} = \boldsymbol{E}_{\hat{v}\sim p^*(\hat{v}|\hat{u})}[b_1(\hat{u},\hat{v})|\hat{u}]$

Previous Methods Revisit

Ideal: $\frac{d\hat{u}}{dt} = \boldsymbol{E}_{\hat{v}\sim p^*(\hat{v}|\hat{u})}[b_1(\hat{u},\hat{v})|\hat{u}]$

$\partial_t u = \boldsymbol{E}_{v\sim T^{\#}p^*(v|u)}[L(u+v)|u]$

Closure model

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

Handcraft Model:
Sub-grid Stress (SGS) model (at most 11parameter) → *Not Expressive Enough?*

Learned Model:
(1)Fix an ansatz $clos(u;\theta)$
(2) Supervised loss: $||(FL - LF)\boldsymbol{u} - clos(F\boldsymbol{u};\theta)||$

→ [**Fact**] Minimizing $||\cdot||_H$-*loss* is equivalent to choosing $q = p_{data}(u,v)$

[**Paradox**] p^* is inf-dim in nature, so approximating it with sampling is very costly!

(1) The DNS data suffices to approximate p^* well: no longer need LES!

(2) The DNS data is not enough: p_{data} is far from p^*, thus the limit distribution could be rather fallacious!

Previous Methods Revisit

What about random sampling method?

Neural Ideal Large Eddy Simulation: Modeling Turbulence with Neural Stochastic Differential Equations

The evolution of the ideal LES field v is obtained from the time derivatives of the set of unfiltered turbulent fields whose large scale features are the same as v [48]:

$$\frac{\partial v}{\partial t} = \mathbb{E}_{\pi_t}\left[\overline{\frac{\partial u}{\partial t}} \,\middle|\, \overline{u} = v\right] \quad (7)$$

Idea: Train a generative model (via an SDE in latent space)

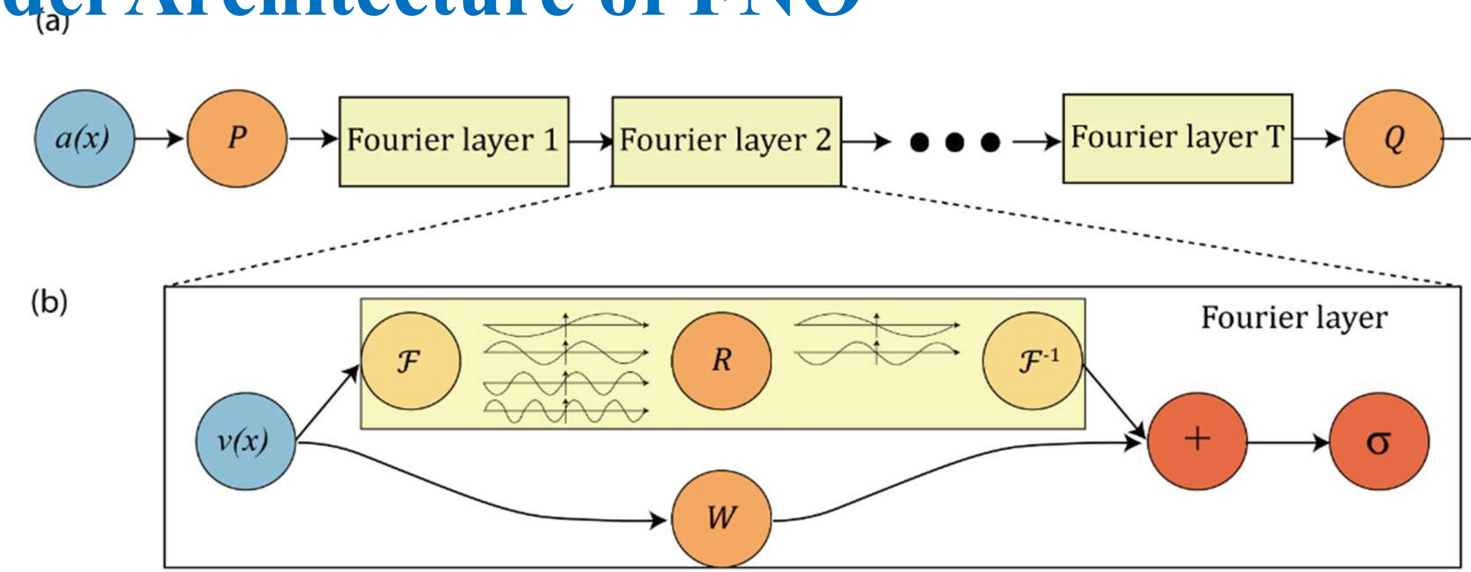
1. Given u, sample v, *or equivalently*, $\boldsymbol{u}$.
2. Compute conditional expectation.

- Write out the PDE of density evolution (Fokker-Planck this time)
- Unfortunately…
- [**Thm**] It could **never** recover p^* or p_1^* unless the stochastic term in SDE is **0**! (i.e. It's a deterministic method).

Outline

- (0) Overview
- (1) Review: previous methods
- (2) Problem reformulation through functional Liouville flow
  Shortcomings of existing methods
- (3) Physics-Informed Neural Operator(PINO)
- (4) Results and Discussion

Model Architecture of FNO



(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.

$$\mathcal{G}_{FNO} := \mathcal{Q} \circ (W_L + \mathcal{K}_L) \circ \cdots \circ \sigma(W_1 + \mathcal{K}_1) \circ \mathcal{P}, \quad (7)$$

where $\mathcal{P}$ and $\mathcal{Q}$ are pointwise lifting and projection operators. The intermediate layers consist of an activation function σ, pointwise operators W_ℓ and integral kernel operators K_ℓ : $u \rightarrow \mathscr{F}^{-1}(R_\ell \cdot \mathscr{F}(u))$, where R_ℓ are weighted matrices and $\mathscr{F}$ denotes Fourier transform.

What makes FNO different?

- (!!!!)Resolution Invariant!
- Long time-ste[illegible]iction.

Notation:

$$I_n: H \to R^n$$

$$u(x) \to \big(u(x_1), u(x_2), \dots u(x_n)\big)$$

Function values on grids.

- The input could be either on coarse grids or fine grids (≈ function)
- [**Thm**] Suppose we have trained a perfect neural operator, then it could recover the optimal approximation of invariant measure, i.e. $p_1^* = P^{\#} p^*$

Key Insights:

- Neural operator G(u, θ): $u(x, t_0) \to u(x, t_0 + h)$ $(or\ \{u(x, t)\}_{t \in [t_0, t_0+h]}$)
- ***An perfect NO***: If input is a function $u \in H$, then $G(u) = S_h u$
- What happens if the input is U $= I_n u$ (functions represented on coarse grid)?
- There exists $u' \in H, s.t.\ I_n u' =$ U , $I_n(S_h u') = G($U). (From FNO architecture)
- μ^* does not depends on trajectories!

How to obtain a perfect FNO?

[Fact] For any non-vanishing $\gamma \in \mathcal{P}(H)$, if $E_{u\sim\gamma}\ell_{PDE}\big(G(u)\big) = 0$, then G is a perfect FNO, where the PDE loss $\ell(u) = ||(\partial_t - L)u||_H^2$

Remarks:

- Does not depend on data distribution
- Does not rely on DNS(ground truth) data.

In practice, it is hard to achieve 0-loss.

[Thm] For $\forall\epsilon > 0$, $\exists\delta > 0$, s.t. if $|G(u) - S_h u| < \delta, \forall u \in H$,

then $\mathcal{W}_2(p_1^*, \hat{p}_1) < \epsilon$. $\hat{p}_1 \coloneqq \lim_{M\to\infty} \frac{1}{M}\sum_{m=1}^{M} \delta_{G^m(I_N u)}$.

Practical Algorithm

[Fact] For any non-vanishing $\gamma \in \mathcal{P}(H)$, if $E_{u\sim\gamma}\ell_{PDE}\big(G(u)\big) = 0$, then G is a perfect FNO, where the PDE loss $\ell(u) = ||(\partial_t - L)u||_H^2$

Remarks:

- PDE-loss is super hard to optimize!
- Solution: find a good initialization with the help of data.

[Algorithm]

➢Supervised learning with LES data.

➢Supervised learning with DNS data (very few).

➢Minimizing the PDE loss function.

Outline

- (0) Overview
- (1) Review: previous methods
- (2) Problem reformulation through functional Liouville flow
  Shortcomings of existing methods
- (3) Physics-Informed Neural Operator(PINO)
- (4) Results and Discussion

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

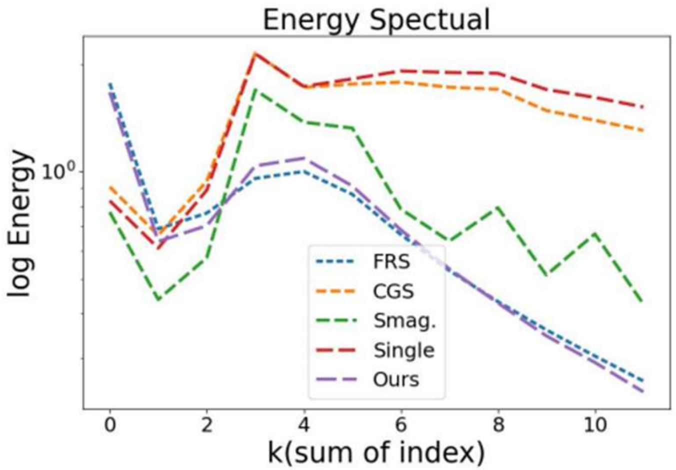
2D Kolmogorov Flow (toy test example)

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

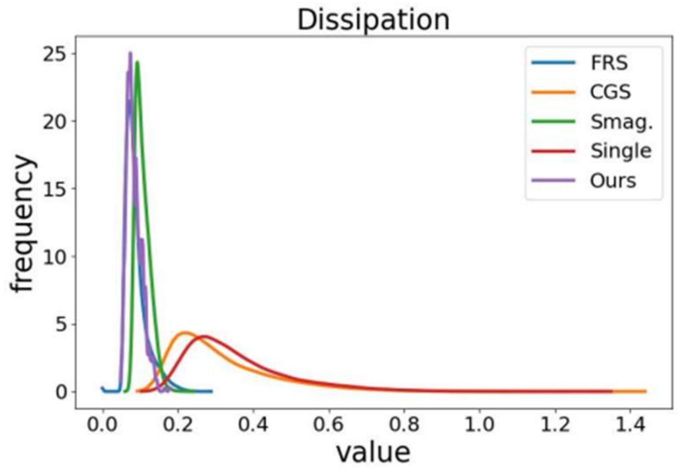
- Reynolds number: 100 L: 2π
- DNS: 128*128 spatial grid
- LES: 16*16 spatial grid
- Note: baseline methods are trained with the same number of DNS data as ours.

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%**

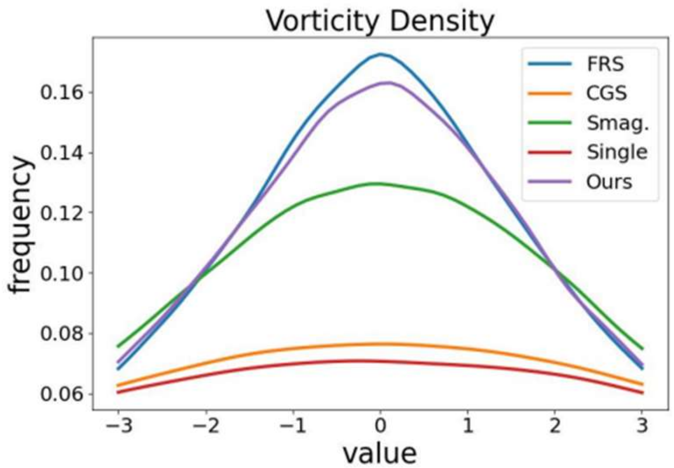
Energy Spectrum Vorticity Total Variation of each mode



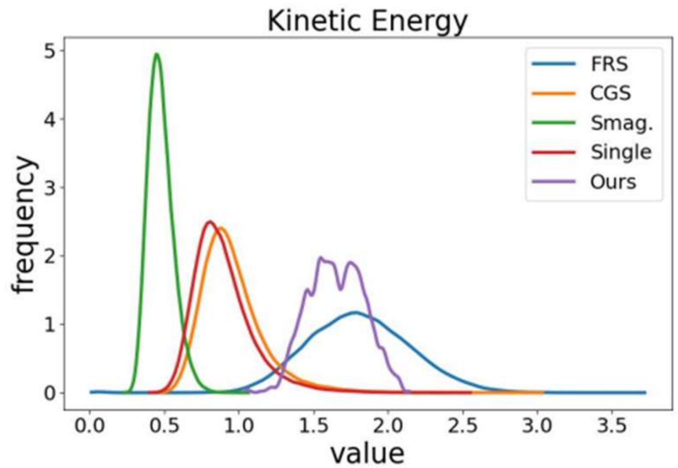
(a) Energy Spectrum



(b) Dissipation Distribution



(c) Vorticity Distribution



(d) Kinetic Energy Distribution

Summary

- ➢[**Closure Model**]: Evolve $\partial_t \bar{u} = L\bar{u} + clos(\bar{u};\theta)$
- [**Operator Learning**]
- New scheme: Evolve $\partial_t \bar{u} = \boldsymbol{L(\theta)}\bar{u}$
- Where is the missing information (error correcting term)?

  -They (their effects) are incorporated in the model.
- Intuition: the error-correcting term is implicitly involved in the simulation.

Thanks!