Complex Fluids and Complex Flows Group
Dept. Physics & INFN - University of Rome 'Tor Vergata'
biferale@roma2.infn.it
https://biferale.web.roma2.infn.it/


Tianyi Li


Chiara Calascibetta


Michele Buzzicotti


Robin Heinonen

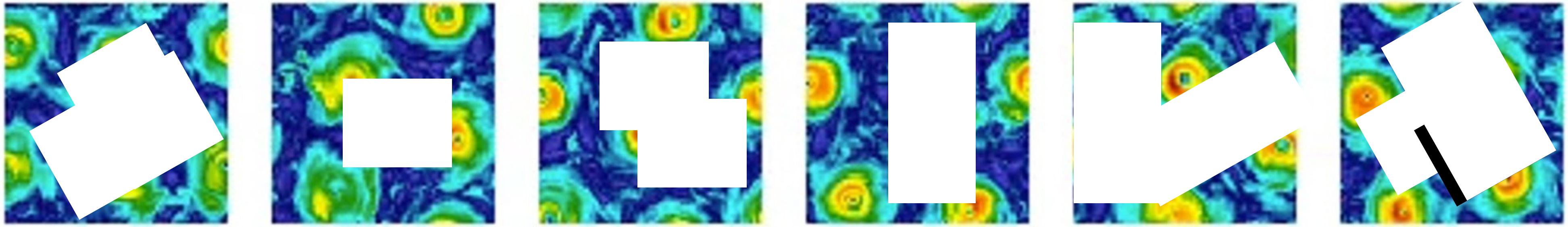

Damiano Capocci


Fabio Bonaccorso

CFDParSchool GSSI 2023

Equations-**Informed** and **Dat**a-Driven Tools For Turbulent Flows

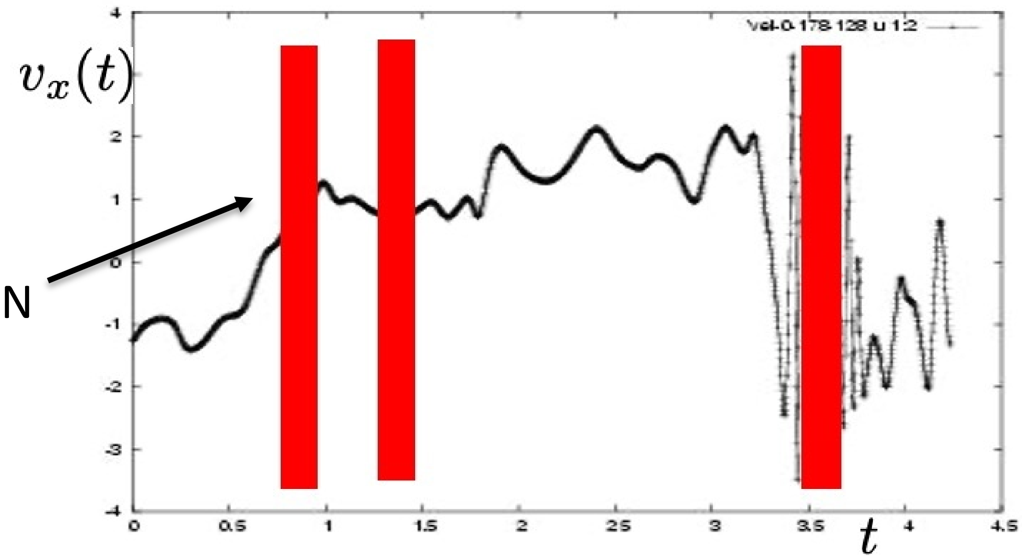
Complex Fluids and Complex Flows Group
Dept. Physics & INFN - University of Rome 'Tor Vergata'
biferale@roma2.infn.it
https://biferale.web.roma2.infn.it/



Rotation rate? Viscosity? Reynolds number? Boundary comditions? Forcing properties? Aspect ratio?

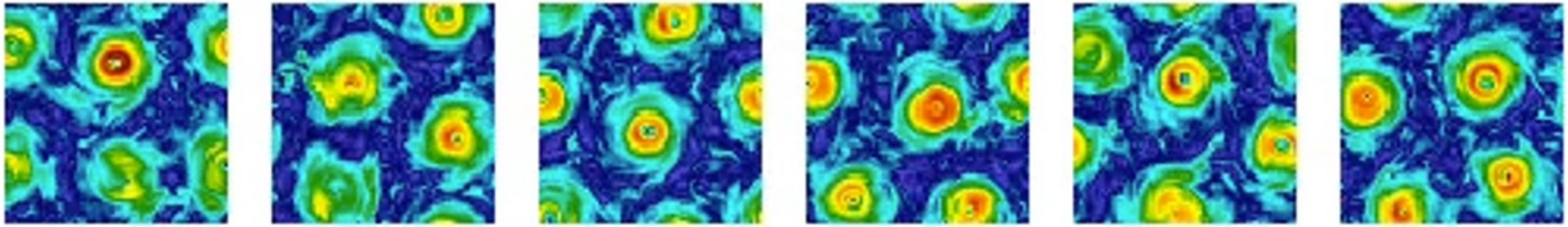
EULERIAN

LAGRANGIAN



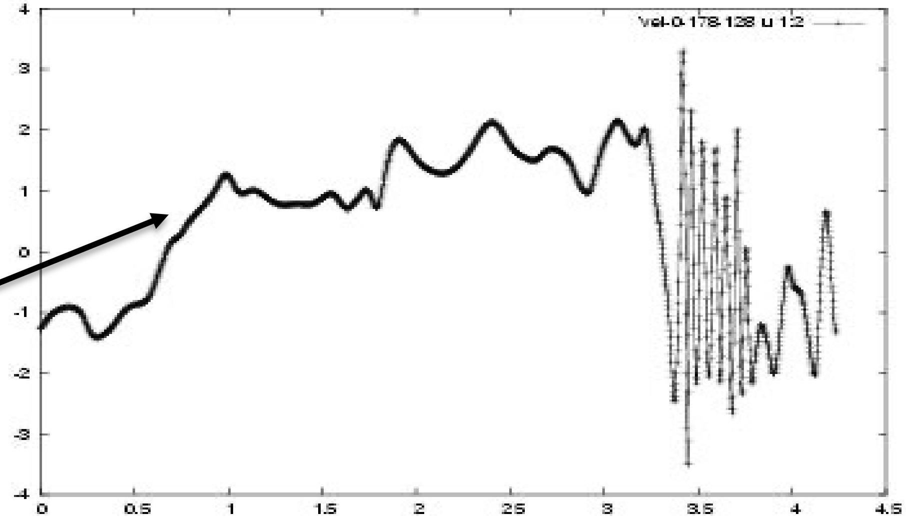
CREDITS: T. LI, M. BUZZICOTTI, S. CHEN. M. WAN

Complex Fluids and Complex Flows Group
Dept. Physics & INFN - University of Rome 'Tor Vergata'
biferale@roma2.infn.it
https://biferale.web.roma2.infn.it/



Rotation rate? Viscosity? Reynolds number? Boundary comditions? Forcing properties? Aspect ratio?

EULERIAN

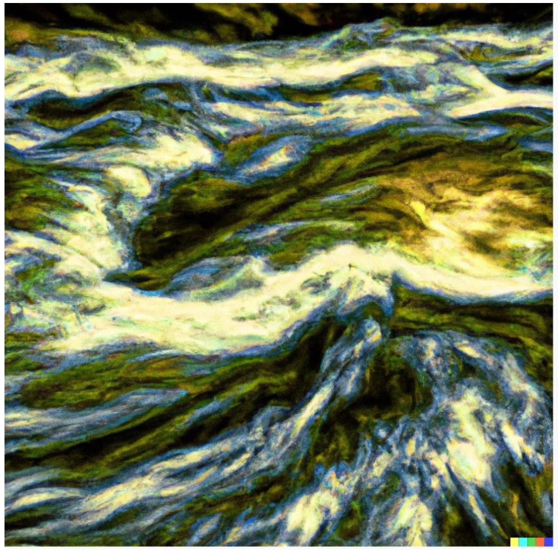


LAGRANGIAN

CREDITS: T. LI, M. BUZZICOTTI, S. CHEN. M. WAN

1. Short introduction to Eulerian and Lagrangian Turbulence in 2D, 3D and in between
2. Data-driven and Equation-Informed tools for Eulerian Turbulence
3. Data-driven and Equation-Informed tools for Lagrangian Turbulence

A WATER COLOR STYLE IMAGE OF A TURBULENT FLOW



MADE BY DALL-E OpenAI

A MIRO' STYLE IMAGE OF A TURBULENT FLOW

MADE BY DALL-E OpenAI

A CUBIST STYLE IMAGE OF A TURBULENT FLOW

MADE BY DALL-E OpenAI

COMPLEX FLUIDS & COMPLEX FLOWS

EULERIAN

$$\begin{cases} \partial_t v + v \cdot \partial_x v = -\partial_x p + \nu \partial^2 v + g\theta + b \cdot \partial_x b + f + \sum_i \delta(x - X_i)\mathcal{F} \\ \partial_x \cdot v = 0 \\ +b.c. \\ \partial_t \theta + v \cdot \partial_x \theta = \chi_\theta \partial^2 \theta \\ \partial_t b + v \cdot \partial_x b - b \cdot \partial_x v = \chi_b \partial^2 b \end{cases}$$

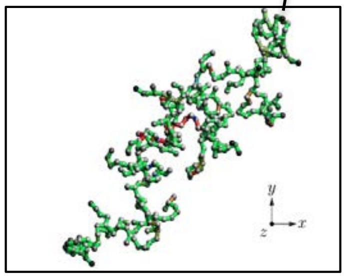
temperature

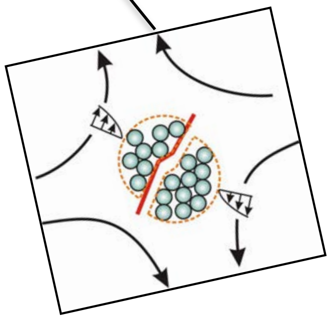
magnetic field

LAGRANGIAN

$$\begin{cases} \dot{X}_i = U_i \\ \dot{U}_i = \frac{v(X_i) - U_i}{\tau} + \beta D_t v(X_i) + U_i^C \end{cases}$$

small active/passive
particles/droplets/bubbles/colloidal aggregates





COMPLEX FLUIDS & COMPLEX FLOWS

EULERIAN

$$\begin{cases} \partial_t v + v \cdot \partial_x v = -\partial_x p + \nu \partial^2 v \quad + f \\ \partial_x \cdot v = 0 \\ +b.c. \end{cases}$$

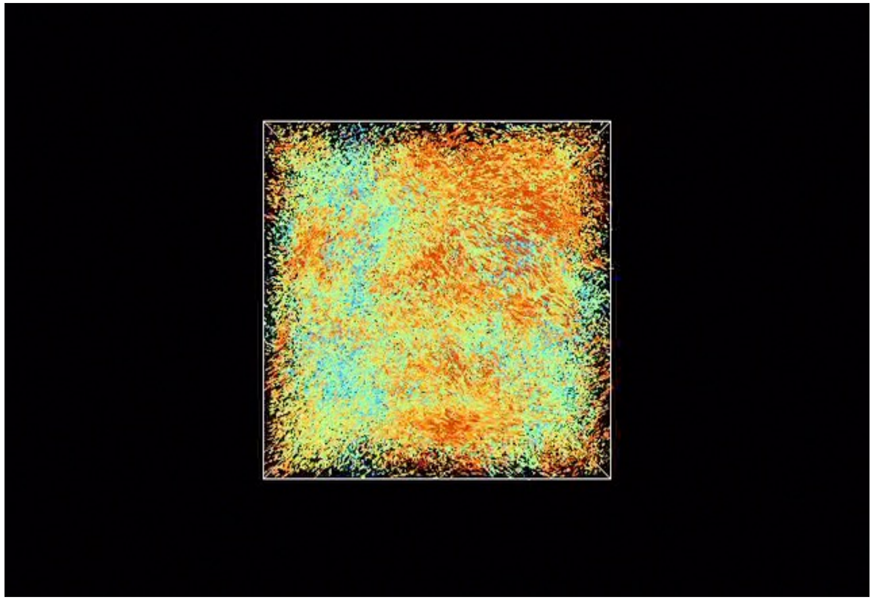
temperature

magnetic field

LAGRANGIAN

$$\left\{ \dot{X}_i = v(X_i(t), t) \right.$$

passive fluid tracers

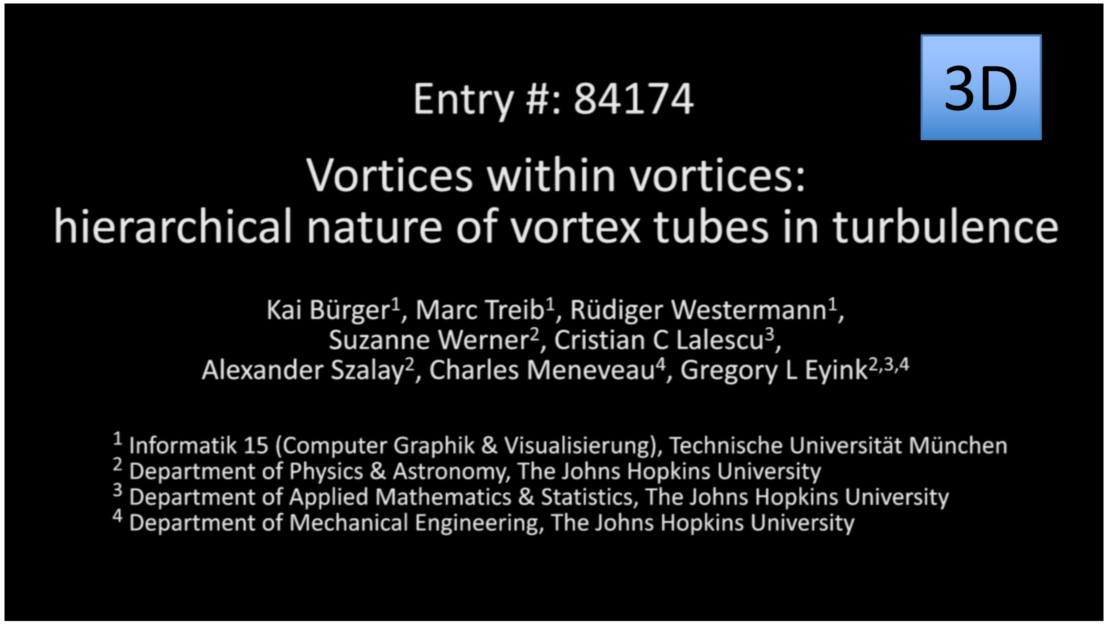


CONTROL PARAMETER:

$$Re \sim \frac{v \partial_x v}{\nu \partial^2 v} \sim \frac{v_0 L_0}{\nu}$$

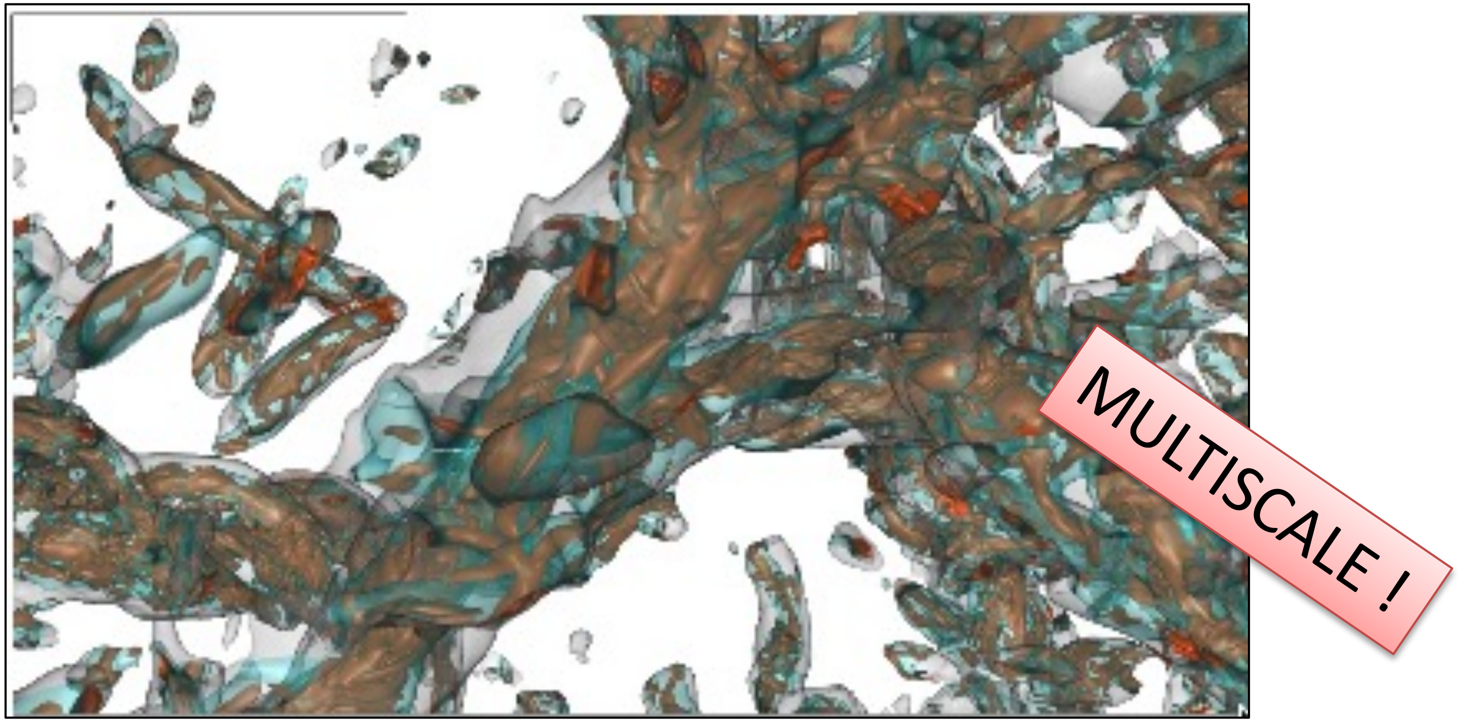
(NASA/Goddard Space Flight Center Scientific Visualization Studio)

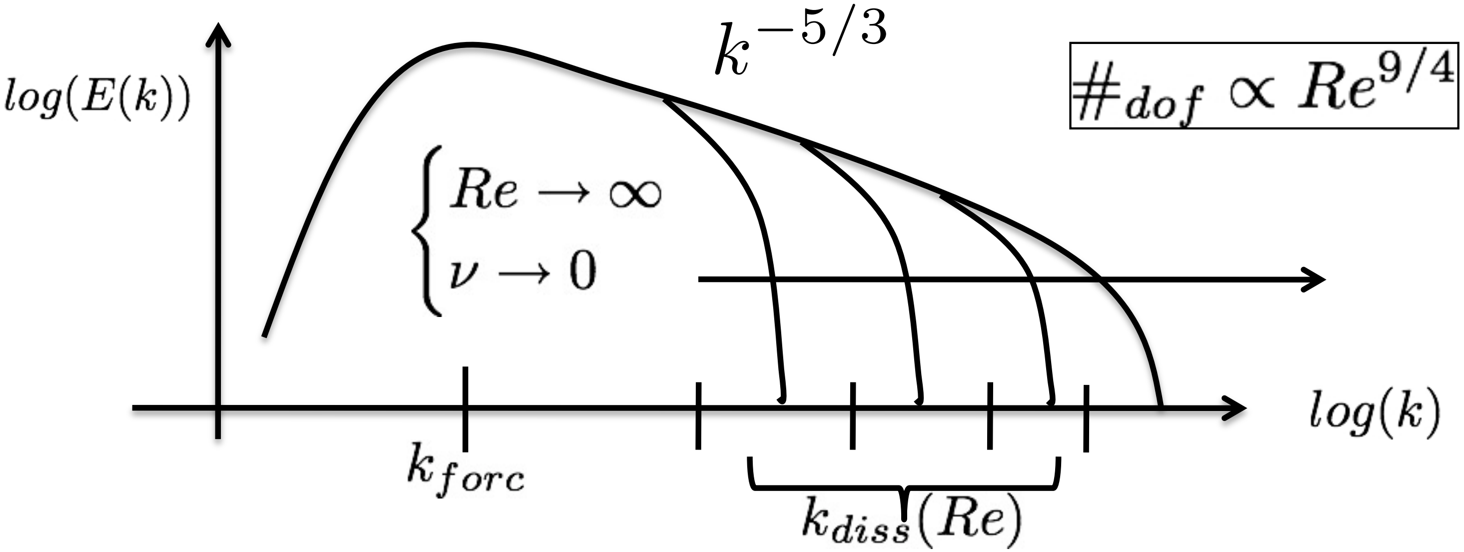
NAVIER-STOKES 3D-2D:
DIMENSIONS MATTER!



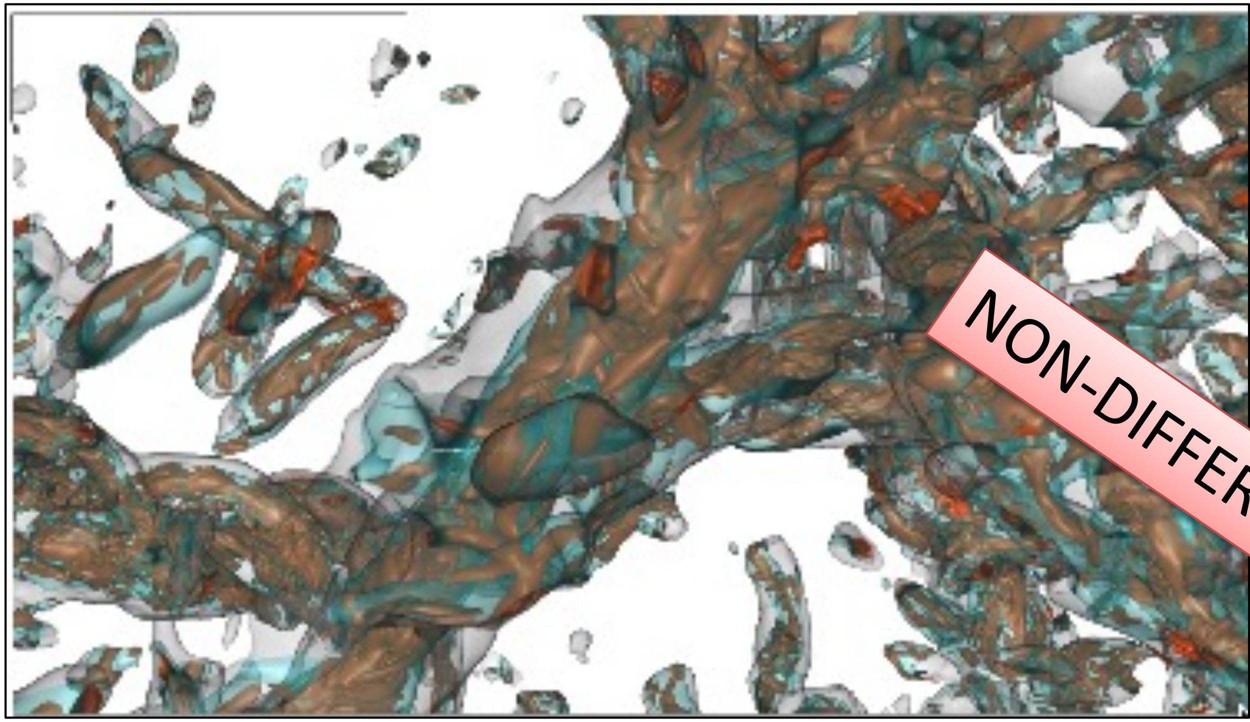
ULTRAVIOLET (3D)
AND INFRARED (2D)
PHYSICS

NAVIER-STOKES 3D

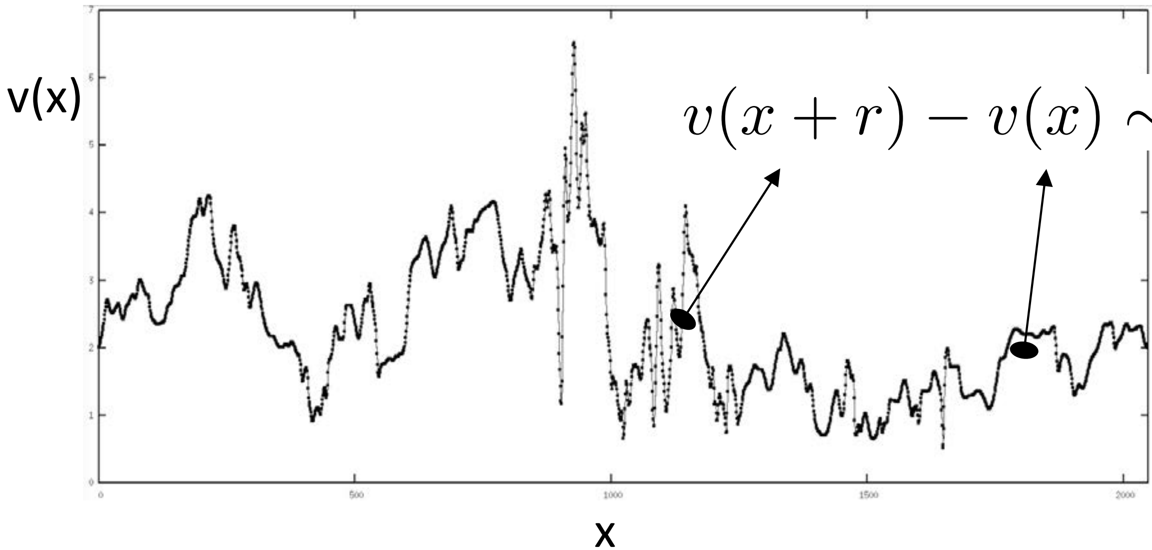




NAVIER-STOKES 3D



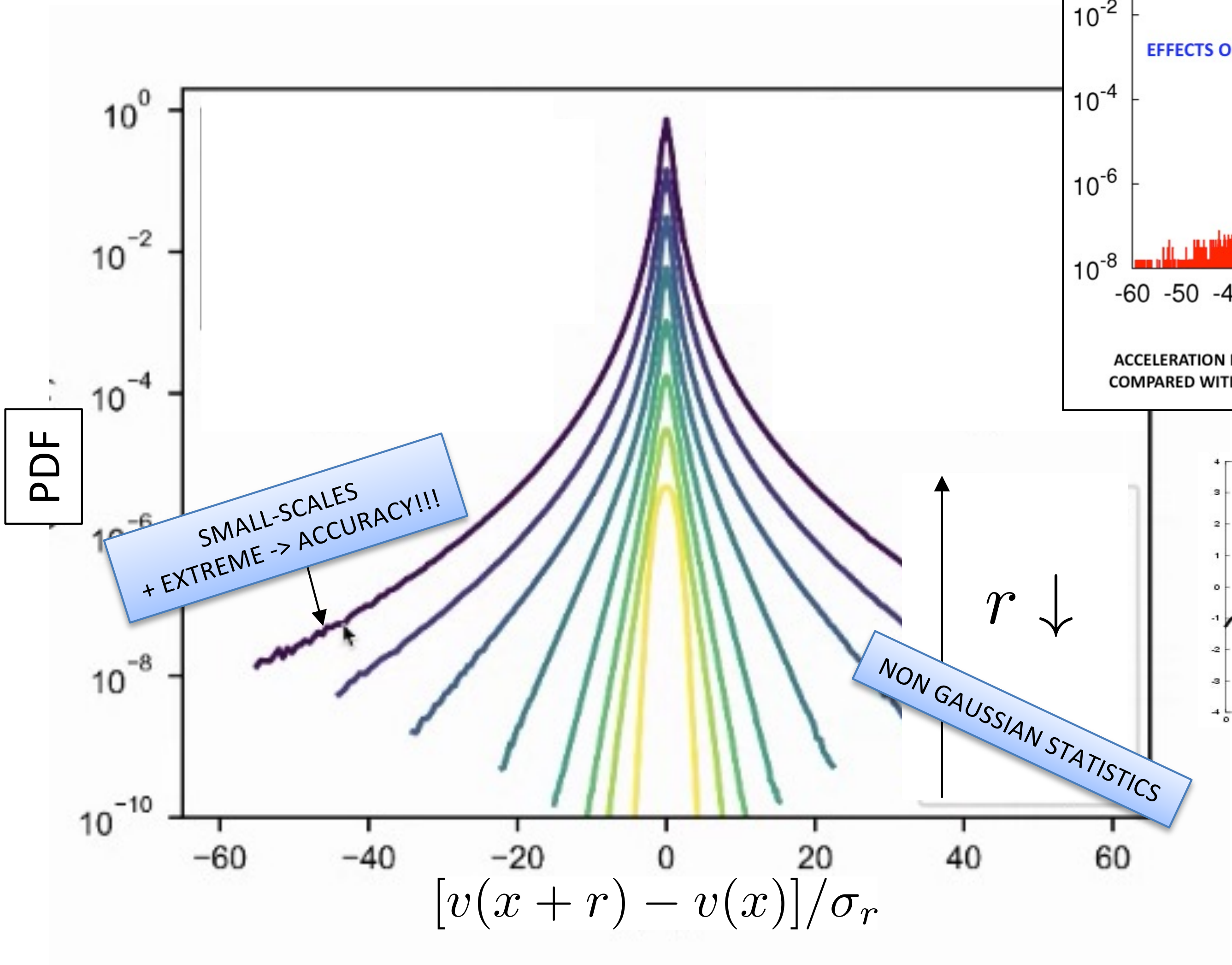
NON-DIFFERENTIABLE !

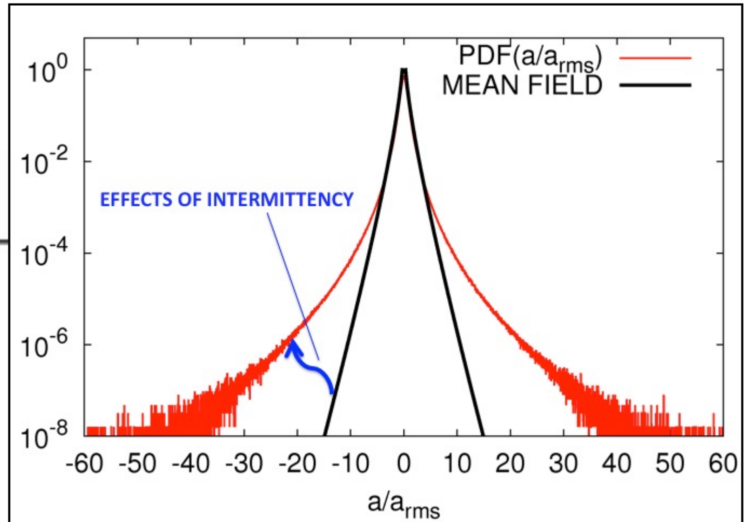


$$v(x+r) - v(x) \sim r^{1/3}$$

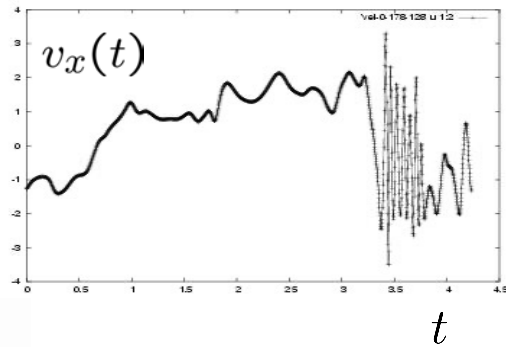
Holder continuos 1/3
non-differentiable!

NAVIER-STOKES 3D





ACCELERATION PROBABILITY DISTRIBUTION FUNCTION (PDF) AT RE ~ 10^5 [Bi04]
COMPARED WITH THE PREDICTION FROM MEAN FIELD (KOLMOGOROV THEORY)



Bentkamp, L, Cr C. Lalescu, and M. Wilczek. "*Nature communications* 10.1 (2019): 1-8.

COMPLEX FLUIDS & COMPLEX FLOWS

EULERIAN

$$\begin{cases} \partial_t v + v\cdot\partial_x v = -\partial_x p + \nu\partial^2 v + g\theta + b\cdot\partial_x b + f + \sum_i \delta(x-X_i)\mathcal{F} \\ \partial_x \cdot v = 0 \\ +b.c. \\ \partial_t\theta + v\cdot\partial_x\theta = \chi_\theta\partial^2\theta \\ \partial_t b + v\cdot\partial_x b - b\cdot\partial_x v = \chi_b\partial^2 b \end{cases}$$

temperature

m

ve

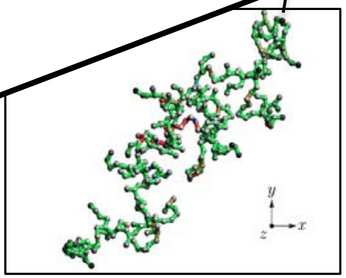
roplets/bubbles/colloidal aggregates

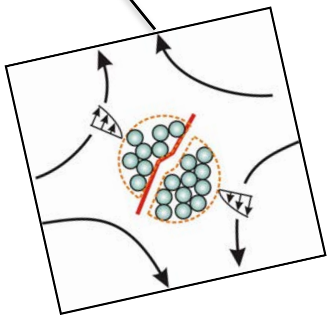
LAGRANGIAN

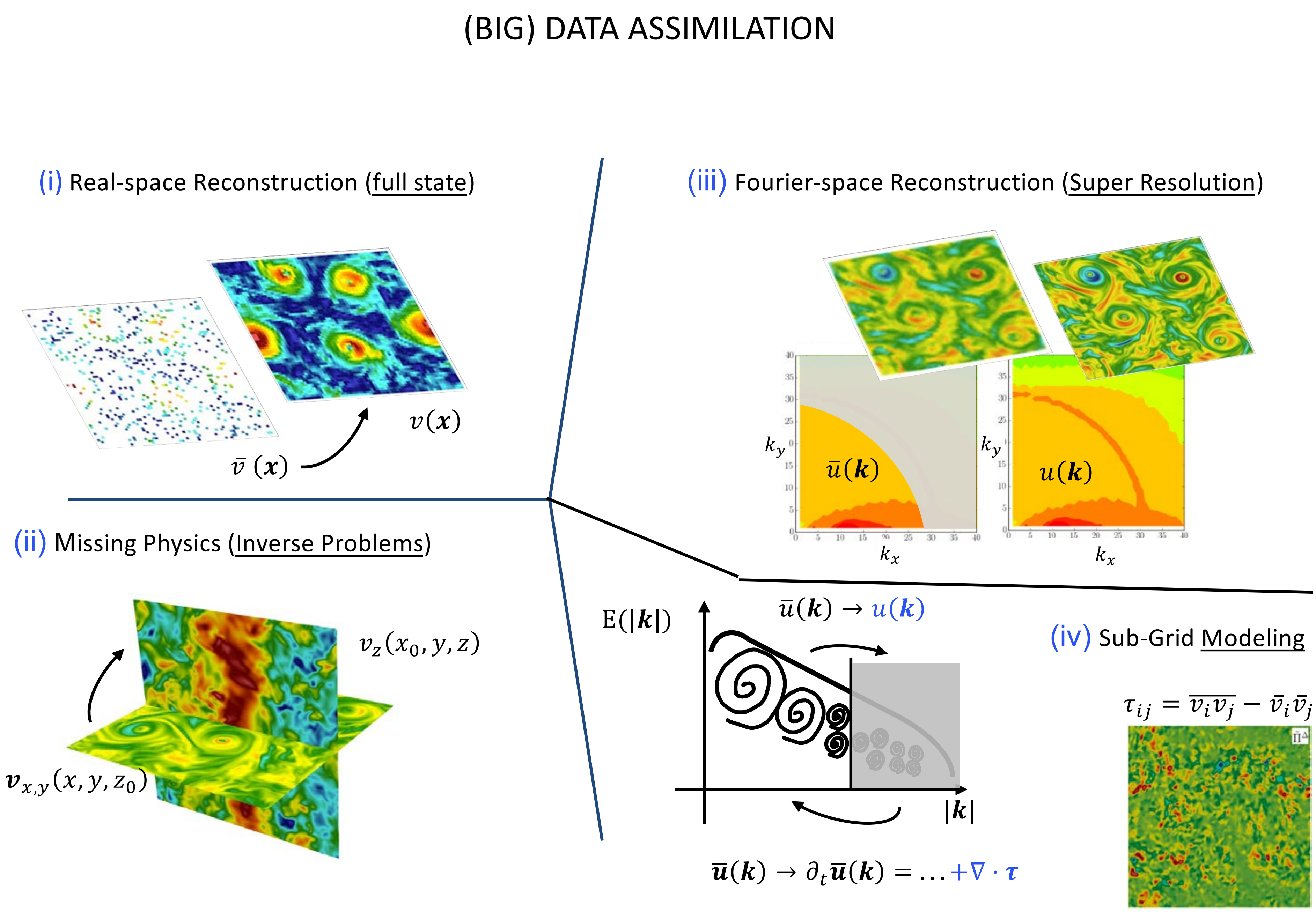
$$\begin{cases} \dot{X}_i = U_i \\ \dot{U}_i = \frac{v(X_i)-U_i}{} \end{cases}$$

1. NO WAY TO PREDICT STATISTICS FOR MEAN PROFILES OR EXTREME EVENTS FROM EoM
2. NO WAY TO PERFORM DIRECT NUMERICAL SIMULATIONS FOR REALISTIC PROBLEMS

WE NEED QUANTITATIVE MODELS AND TOOLS TO MODEL!





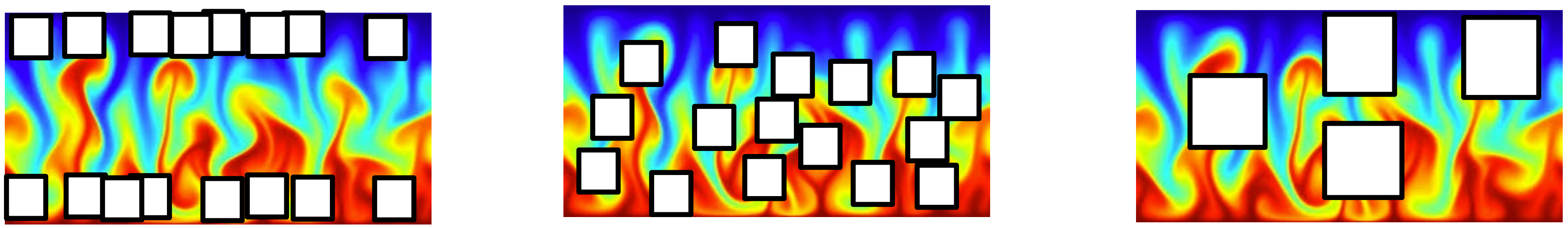


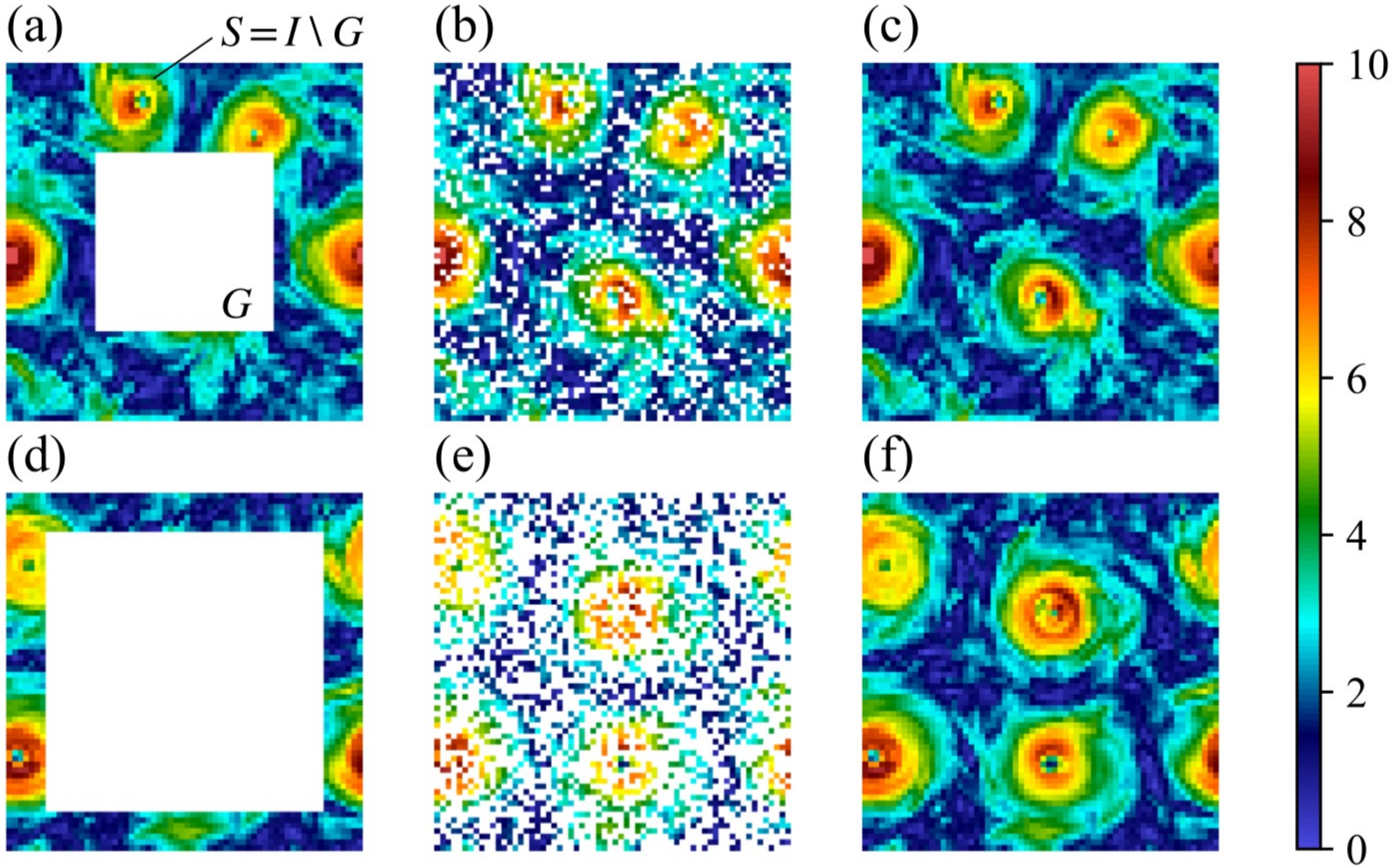
(BIG) DATA ASSIMILATION
(i) Real-space Reconstruction (full state)
$v(\boldsymbol{x})$
$\bar{v}\ (\boldsymbol{x})$
(ii) Missing Physics (Inverse Problems)
$v_z(x_0, y, z)$
$\boldsymbol{v}_{x,y}(x, y, z_0)$
(iii) Fourier-space Reconstruction (Super Resolution)
k_y
k_x
$\bar{u}(\boldsymbol{k})$
$u(\boldsymbol{k})$
E(|k|)
$\bar{u}(\boldsymbol{k}) \to u(\boldsymbol{k})$
$|\boldsymbol{k}|$
$\bar{\boldsymbol{u}}(\boldsymbol{k}) \to \partial_t \bar{\boldsymbol{u}}(\boldsymbol{k}) = \ldots + \nabla \cdot \boldsymbol{\tau}$
(iv) Sub-Grid Modeling
$\tau_{ij} = \overline{v_i v_j} - \bar{v}_i \bar{v}_j$

1. RECONSTRUCTION OF MISSING INFORMATION (INPAINTING – SUPER RESOLUTION)
2. FEATURES RANKING: QUALITY AND QUANTITY OF DATA

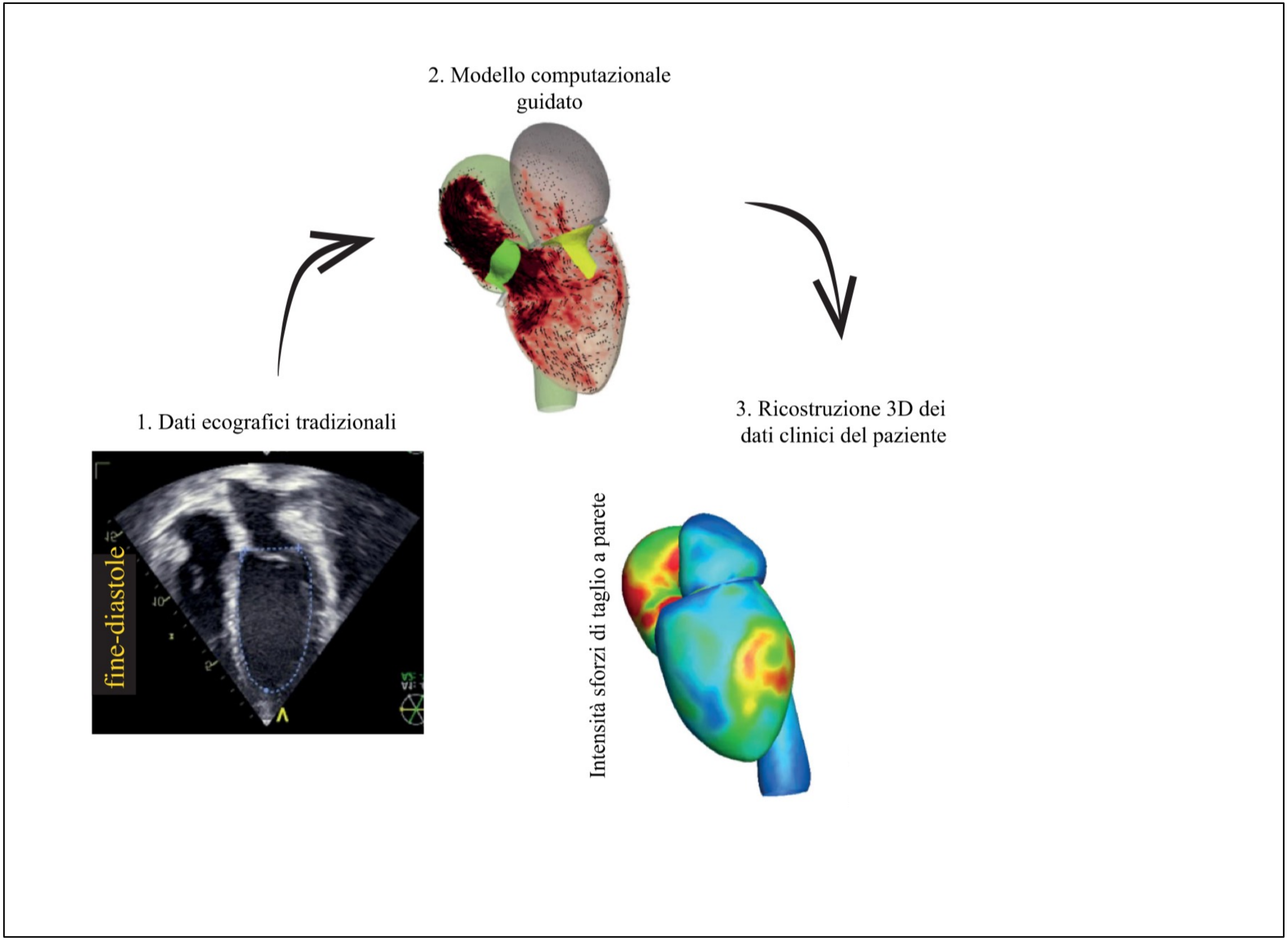
- IS IT BETTER TO INPUT SPATIAL OR TEMPORAL DATA?
- HOW MANY DATA/VARIABLES YOU NEED TO SUPPLY FOR PERFECT RECONSTRUCTION (SYNCHRONIZATION-TO-DATA)?
- CAN YOU GUESS VELOCITY FIELDS BY MEASURING ONLY TEMPERATURE AND/OR VICEVERSA?
- IS IT BETTER TO PROVIDE INFORMATION FROM BOUNDARIES OR BULK?
- FROM LARGE OR SMALL SCALES?
- DO WE NEED TO KNOW THE EQUATIONS?
- HOW TO COMPARE EQUATIONS-BASED AND EQUATIONS-FREE MODELS?

A WAY TO LEARN ABOUT THE UNDERLYING PHYSICS



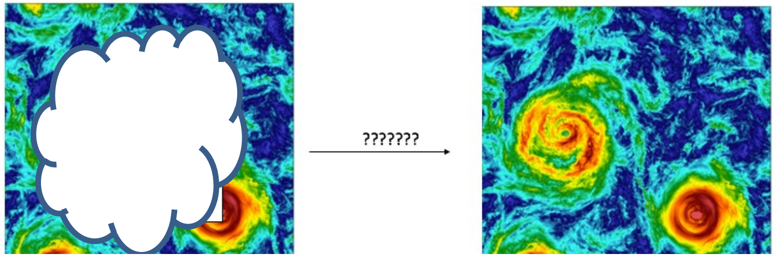


(a)
S = I \ G
G
(b)
(c)
(d)
(e)
(f)
10
8
6
4
2
0



with F. Viola (GSSI) and R. Verzicco (U Tor Vergata)

DATA ASSIMILATION, FLOW RECONSTRUCTION, INPAINTING, SUPER-RESOLUTION, PHYSICS-INFERRING (CLASSIFICATION)

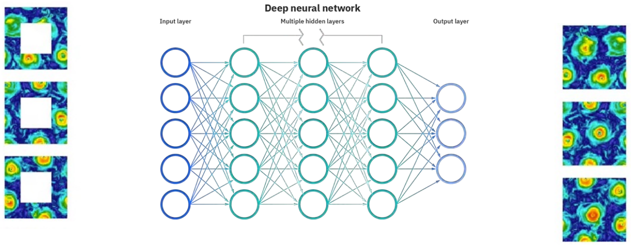


DATA DRIVEN

$$u(\boldsymbol{x}) = \textstyle\sum_{n=1}^{N} a_n \psi_n(\boldsymbol{x}) = \sum_{n=1}^{N'} a_n \psi_n(\boldsymbol{x}) + \sum_{n=N'+1}^{N} a_n \psi_n(\boldsymbol{x})$$

$$\int_S \left[u(\boldsymbol{x}) - \textstyle\sum_{n=1}^{N'} a_n \psi_n(\boldsymbol{x})\right]^2 \mathrm{d}\boldsymbol{x}$$

1. EQUATION FREE
PRINCIPAL ORTHOGONAL DECOMPOSITION
GAPPY-POD & EXTENDED POD



2. EQUATION FREE
GENERATIVE-ADVERSARIAL-NETWORK

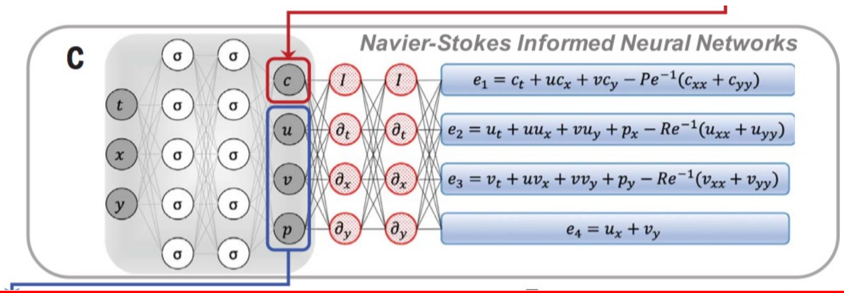
EQ. BASED

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{v} + 2\Omega \hat{x}_3 \times \boldsymbol{v} = -\boldsymbol{\nabla} p + \nu \nabla^2 \boldsymbol{v} + \gamma (1 - \hat{M})_{x_3} \odot (\boldsymbol{v} - \boldsymbol{v}_{\mathrm{ref}})$$

3. NUDGING

DATA DRIVEN

EQ. BASED

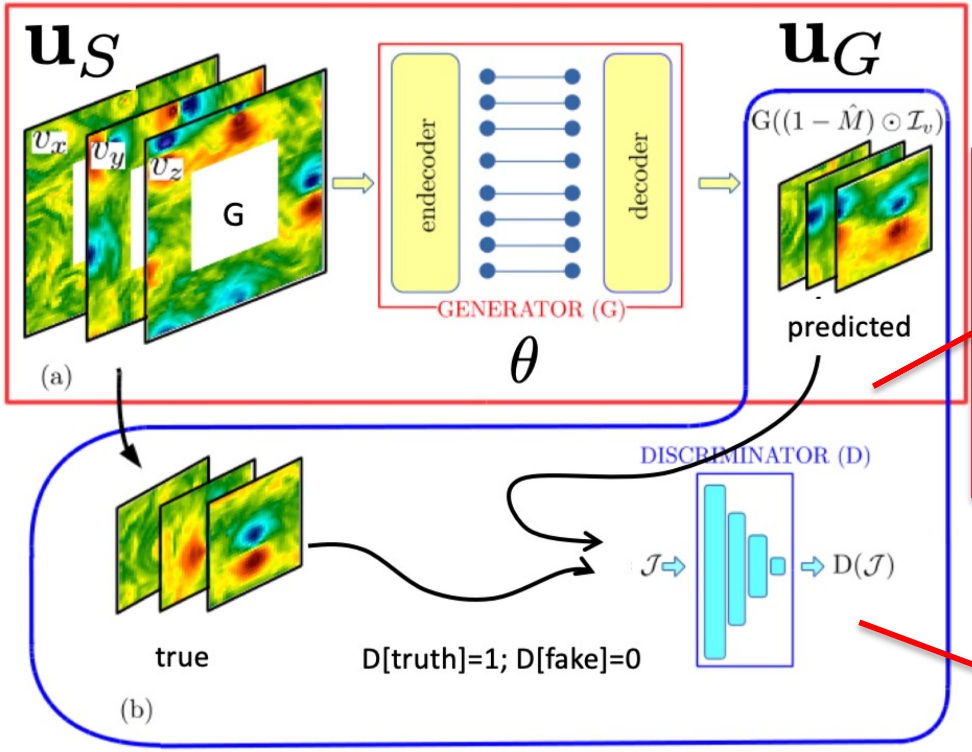


$$MSE = \frac{1}{N}\sum_{n=1}^{N} \left| c(t^n, x^n, y^n, z^n) - c^n \right|^2 + \sum_{i=1}^{5} \frac{1}{M} \sum_{m=1}^{M} \left| e_i(t^m, x^m, y^m, z^m) \right|^2$$

4. PINN
Physics-Informed NN

Reconstruction of turbulent data with deep generative models for semantic inpainting from TURB-Rot database
M. Buzzicotti, F. Bonaccorso, P. Clark Di Leoni, and L. B.
Phys. Rev. Fluids **6**, 050503 , May 2021

GENERATIVE ADVERSARIAL NETWORK: CONTEXT ENCODER (ACTOR-CRITIC)



MINIMIZE:

$$\mathcal{L}_{GEN} = (1-\lambda_{adv})\mathcal{L}_{\mathrm{MSE}} + \lambda_{adv}\mathcal{L}_{adv},$$

$$\mathcal{L}_{\mathrm{MSE}} = \langle \frac{1}{A(I)} \int_I \|\boldsymbol{u}_G^{(p)}(\boldsymbol{x}) - \boldsymbol{u}_G^{(t)}(\boldsymbol{x})\|^2 \, d\boldsymbol{x} \rangle$$

$$\mathcal{L}_{adv} = \langle \log(1 - D(\boldsymbol{u}_G^{(p)})) \rangle$$
$$= \int p(\boldsymbol{u}_S) \log[1 - D(GEN(\boldsymbol{u}_S))] \, d\boldsymbol{u}_S$$

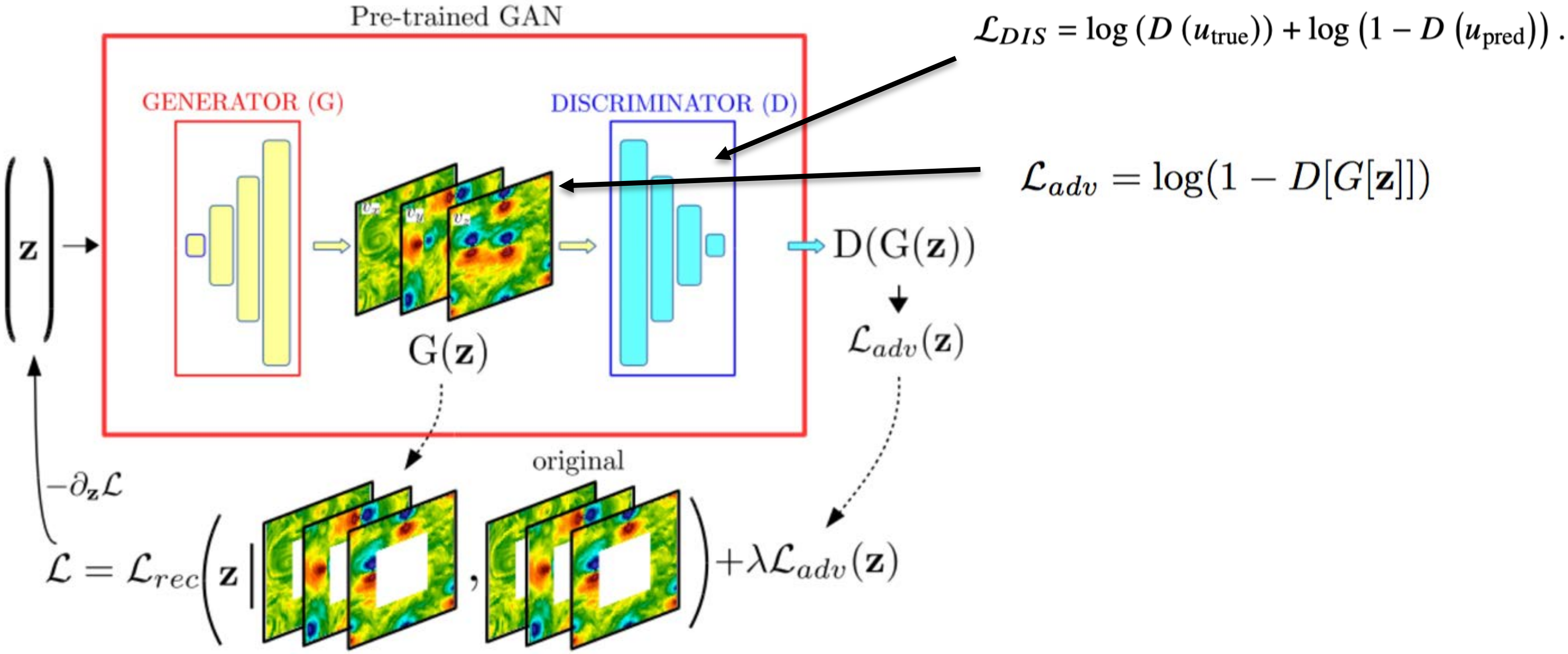
MAXIMIZE:

$$\mathcal{L}_{DIS} = \langle \log(D(\boldsymbol{u}_G^{(t)})) \rangle + \langle \log(1 - D(\boldsymbol{u}_G^{(p)})) \rangle$$
$$= \int [p_t(\boldsymbol{u}_G) \log(D(\boldsymbol{u}_G)) +$$
$$p_p(\boldsymbol{u}_G) \log(1 - D(\boldsymbol{u}_G))] \, d\boldsymbol{u}_G.$$

$$\mathcal{D}^*(\mathbf{u}) = \frac{p_t(\mathbf{u})}{p_t(\mathbf{u}) + p_p(\mathbf{u})}$$

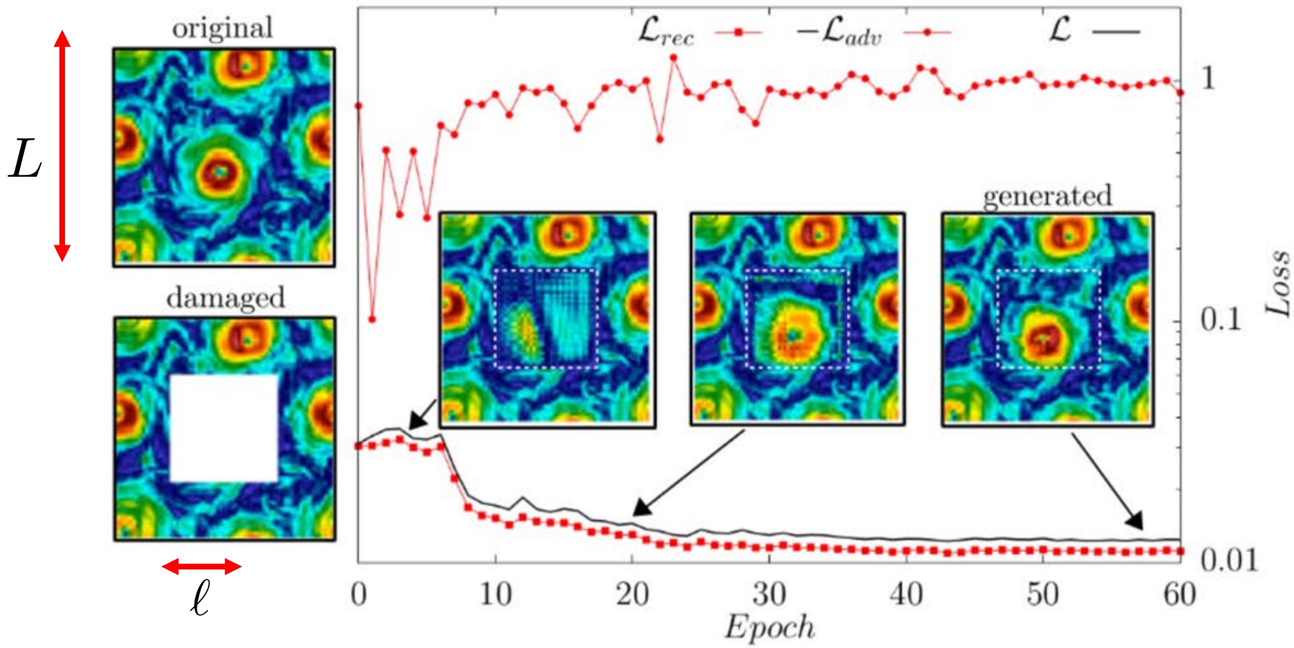
$$KL(p_t \parallel p_p) = \int_{-\infty}^{\infty} p_t(x) \log\left(\frac{p_t(x)}{p_p(x)}\right) dx$$

CONTEXT ENCODER 2 (CE2)
CONSTRAINED IMAGE GENERATION



Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.

DURING TRAINING
80K 64x64 images of velocity amplitude for training
20K 64x64 images of velocity amplitude for validation



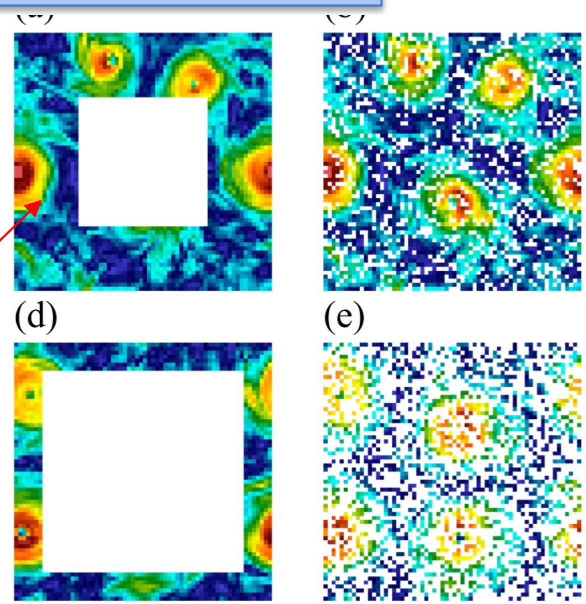
ℓ :much larger than differentiable scale, i.e. velocity fields are rough (no linear interpolation here)

GAPPY-POD (PRINCIPAL ORTHOGONAL DECOMPOSITION)

$$u_{pred}(\mathbf{x}) = \sum_{n=1}^{N} a_n \psi_n(\boldsymbol{x}) = \sum_{n=1}^{N'} a_n \psi_n(\boldsymbol{x}) + \sum_{n=N'+1}^{N} a_n \psi_n(\boldsymbol{x}),$$

$$K(\mathbf{x},\mathbf{y}) = \langle u_{true}(\mathbf{x}) u_{true}(\mathbf{y}) \rangle$$

$$\int K(\mathbf{x},\mathbf{y}) \psi_n(\mathbf{y}) d\mathbf{y} = \lambda_n \psi_n(\mathbf{x})$$



$$X' = \psi \oplus \psi$$

$$\tilde{\boldsymbol{E}} = \|\tilde{\boldsymbol{u}} - \tilde{\boldsymbol{X}}'\boldsymbol{a}'\| = \int_S d\mathbf{x} \left(u_{true}(\mathbf{x}) - \sum_{n=1}^{N'} a_n \psi_n(\mathbf{x})\right)^2$$

LINEAR OPTIMAL REGERESSION

T. Li, L. Buzzicotti, F. Bonaccorso, L.B., S. Chen, M. Wan. Data reconstruction of turbulent flows with Gappy-POD and Generative Adversarial Networks. Submitted to JFM 2022.

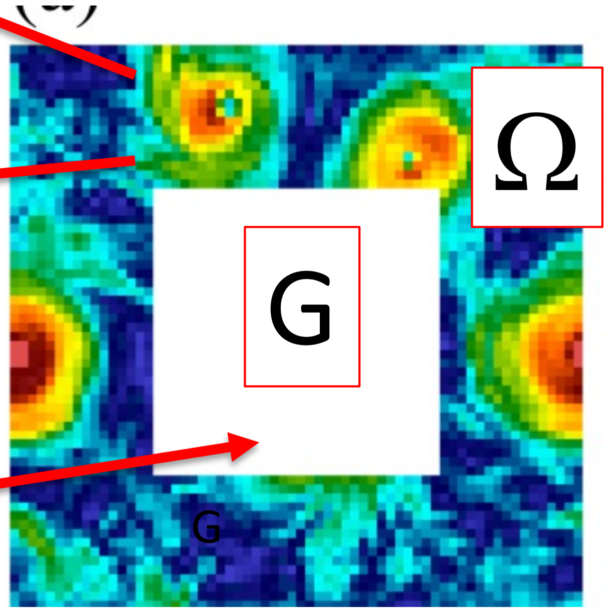
EXTENDED-POD (PRINCIPAL ORTHOGONAL DECOMPOSITION)

$$\int_\Omega R_S(x,y)\phi_S^{(n)}(y)\,dy = \sigma_n \phi_S^{(n)}(x),$$

$$u_S(x) = \sum_{n=1}^{N_S} b_S^{(n)} \phi_S^{(n)}(x),$$

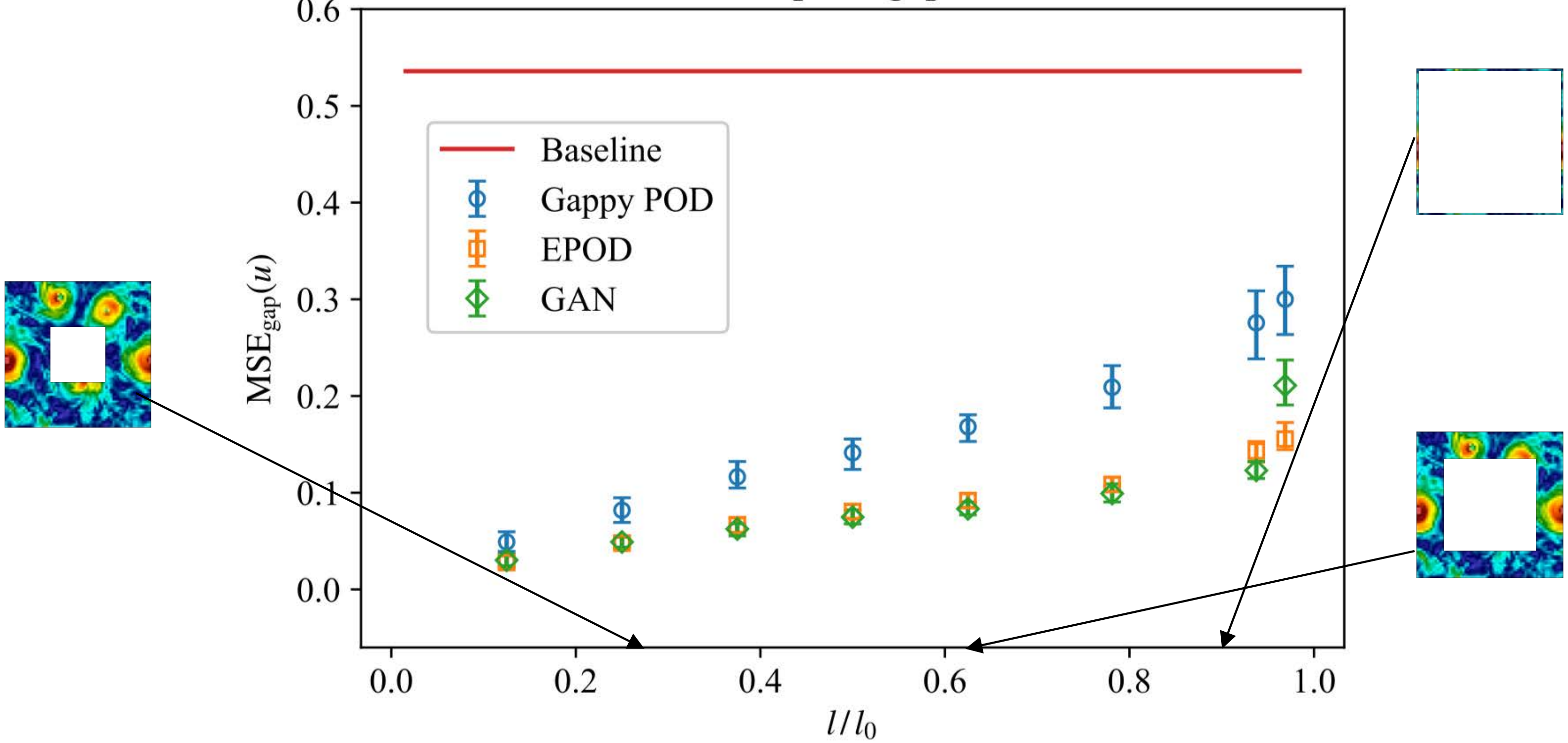
$$\phi_S^{(n)}(x) = \langle b_S^{(n)} u_S(x)\rangle / \sigma_n.$$

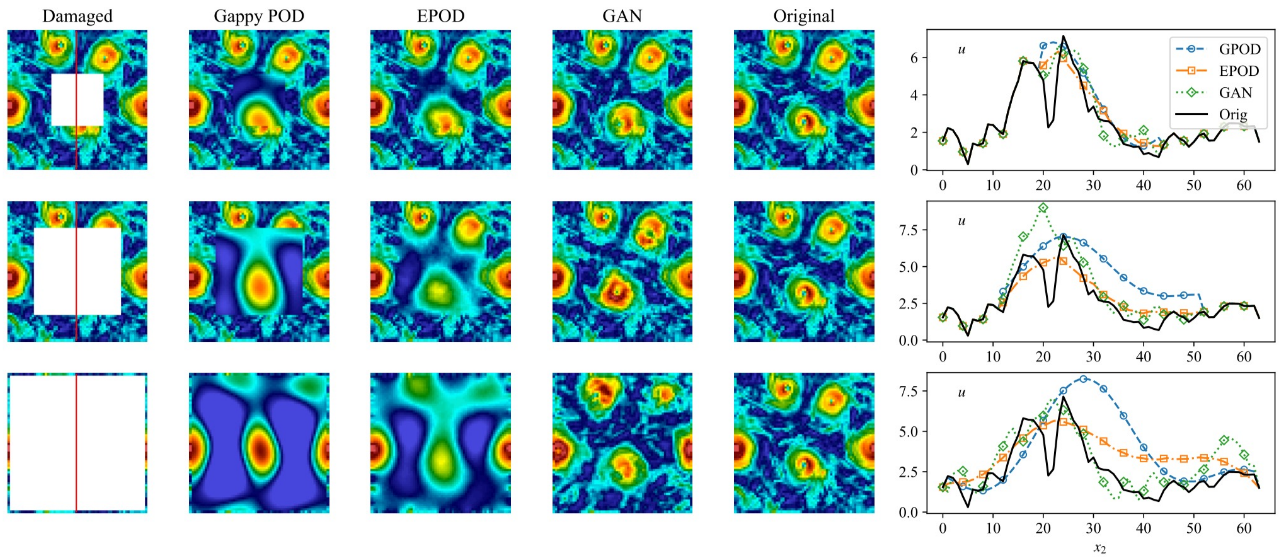
$$\phi_E^{(n)}(x) = \langle b_S^{(n)} u_G(x)\rangle / \sigma_n.$$

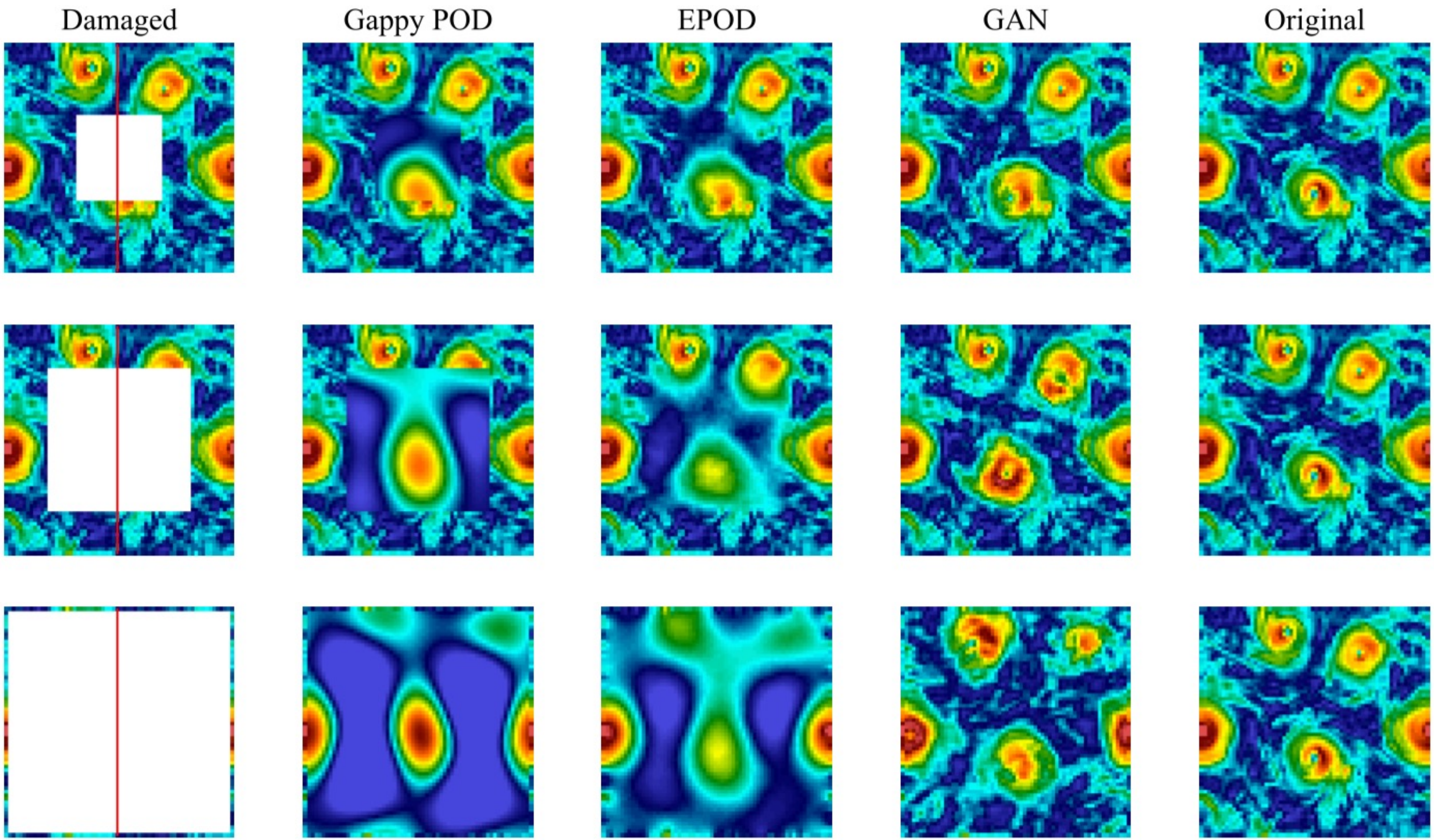


$$u_G^{(p)}(x) = \sum_{n=1}^{N_S} b_S^{(n)} \phi_E^{(n)}(x).$$

$$\mathcal{L}_G = \langle \int_S d\mathbf{x}(u_{true}(\mathbf{x}) - u_{pred}(\mathbf{x}, \theta))^2 \rangle$$






Damaged
Gappy POD
EPOD
GAN
Original

$$D(P \parallel Q) = \int_{-\infty}^{\infty} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \mathrm{d}x$$

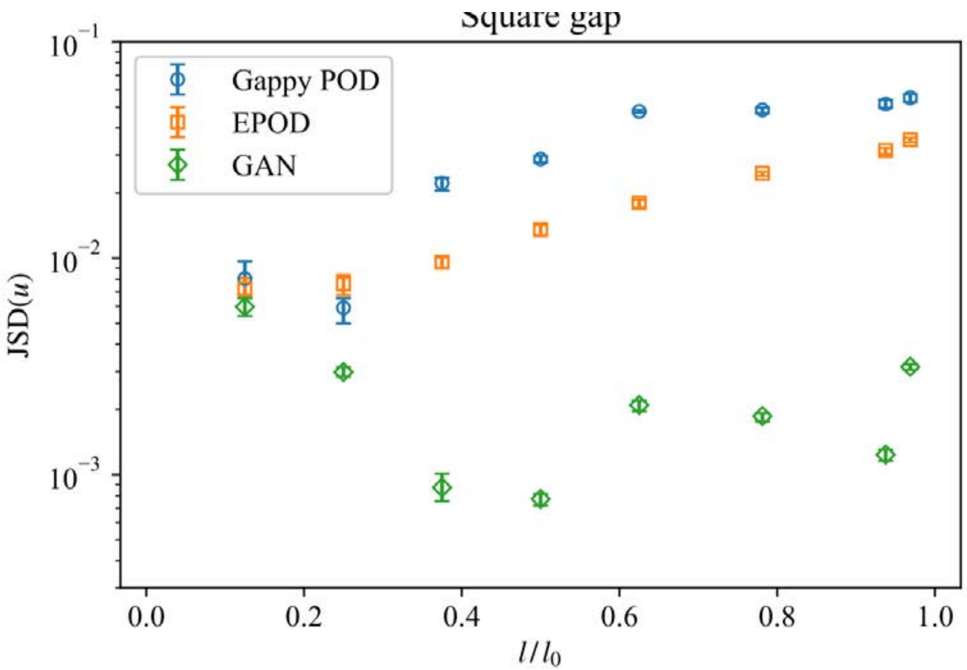
$$M = \tfrac{1}{2}(P + Q)$$

KULLBLACK-LEIBLER

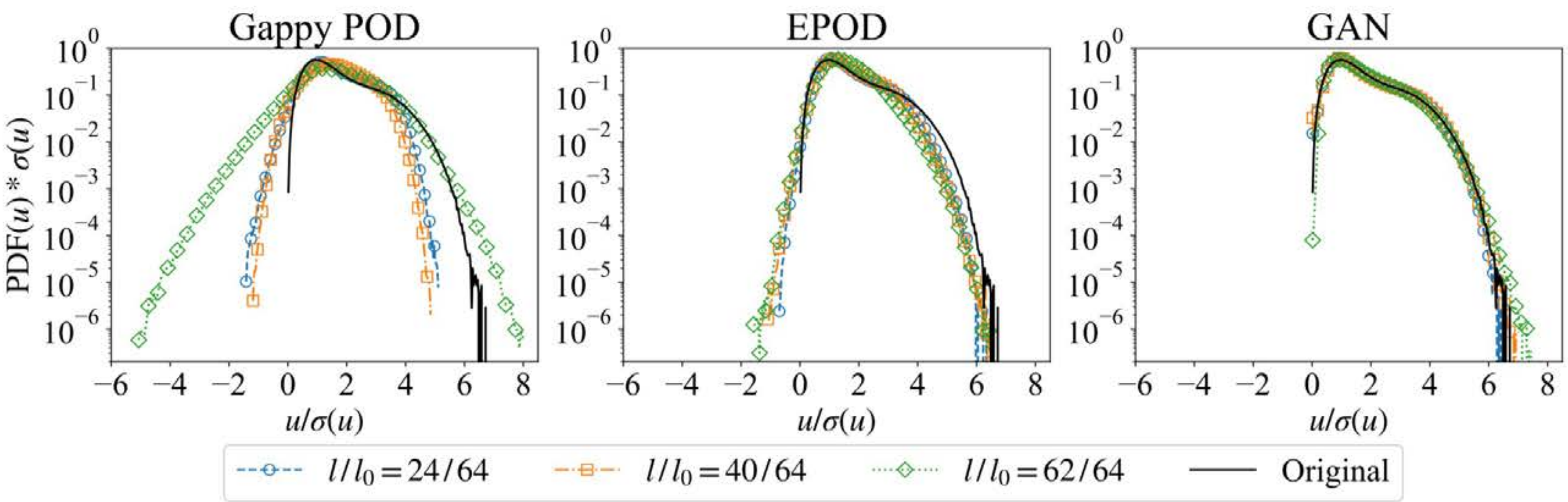
→

JENSEN-SHANNON

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}D(P \parallel M) + \frac{1}{2}D(Q \parallel M),$$

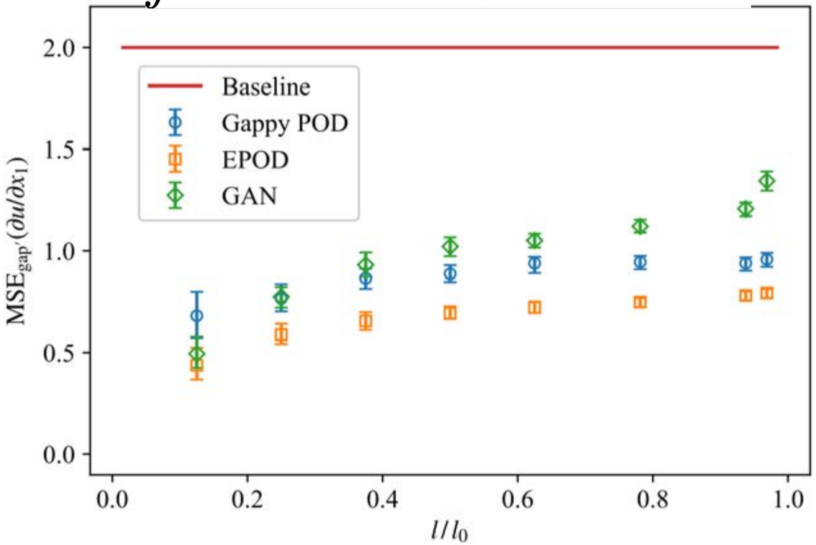


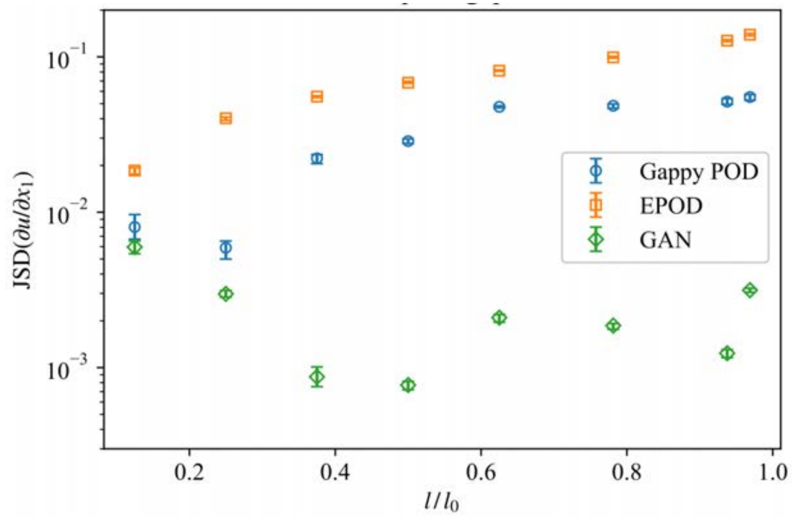
11



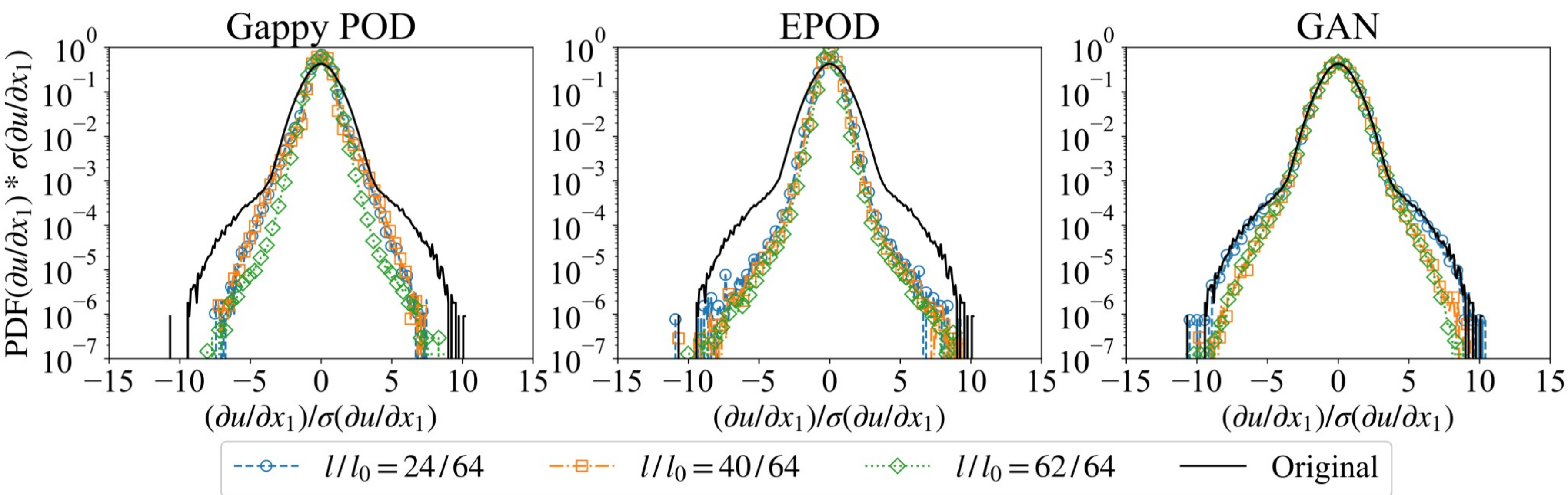
$$\int d\mathbf{x}[\partial_x u^p(\mathbf{x},\theta) - \partial_x u^t(\mathbf{x})]^2$$

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}D(P \parallel M) + \frac{1}{2}D(Q \parallel M),$$





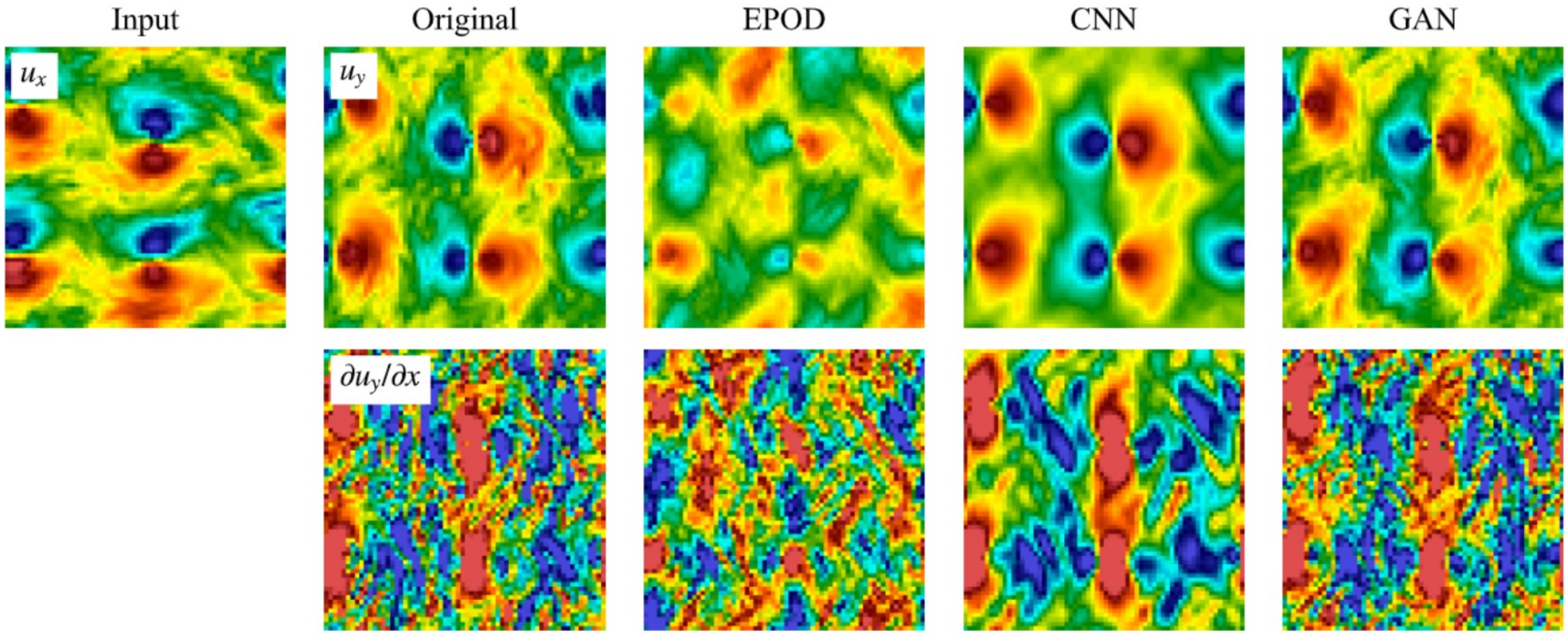
15



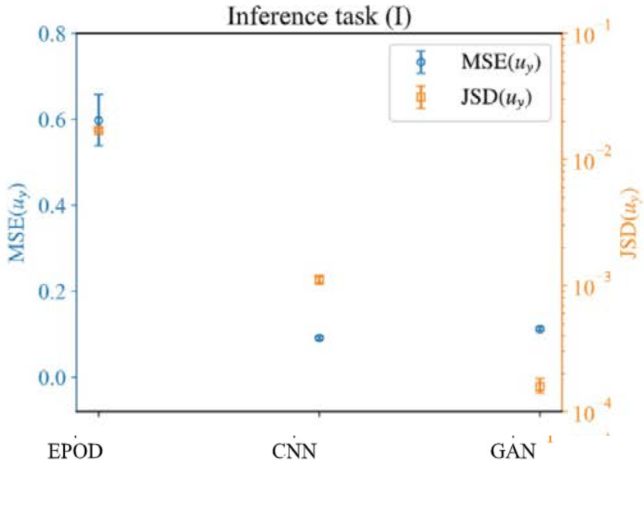
Generative adversarial networks to infer velocity components in rotating turbulent flows

Tianyi Li ✉, Michele Buzzicotti, Luca Biferale & Fabio Bonaccorso

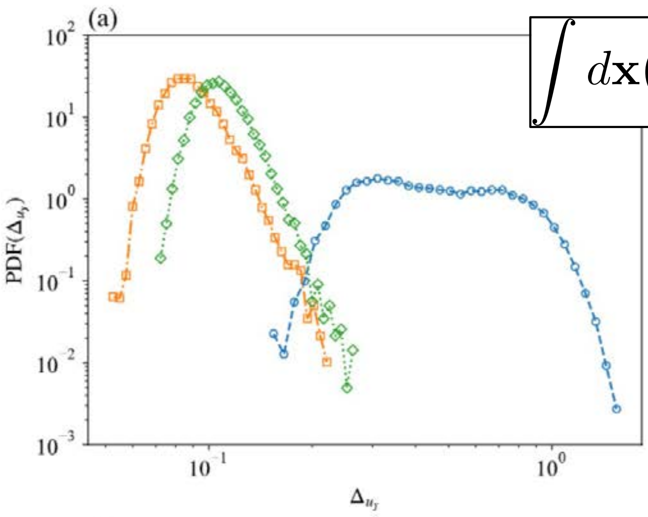
The European Physical Journal E **46**, Article number: 31 (2023)

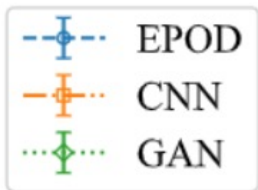


$$\langle \int d\mathbf{x}(u^p - u^t)^2 \rangle$$

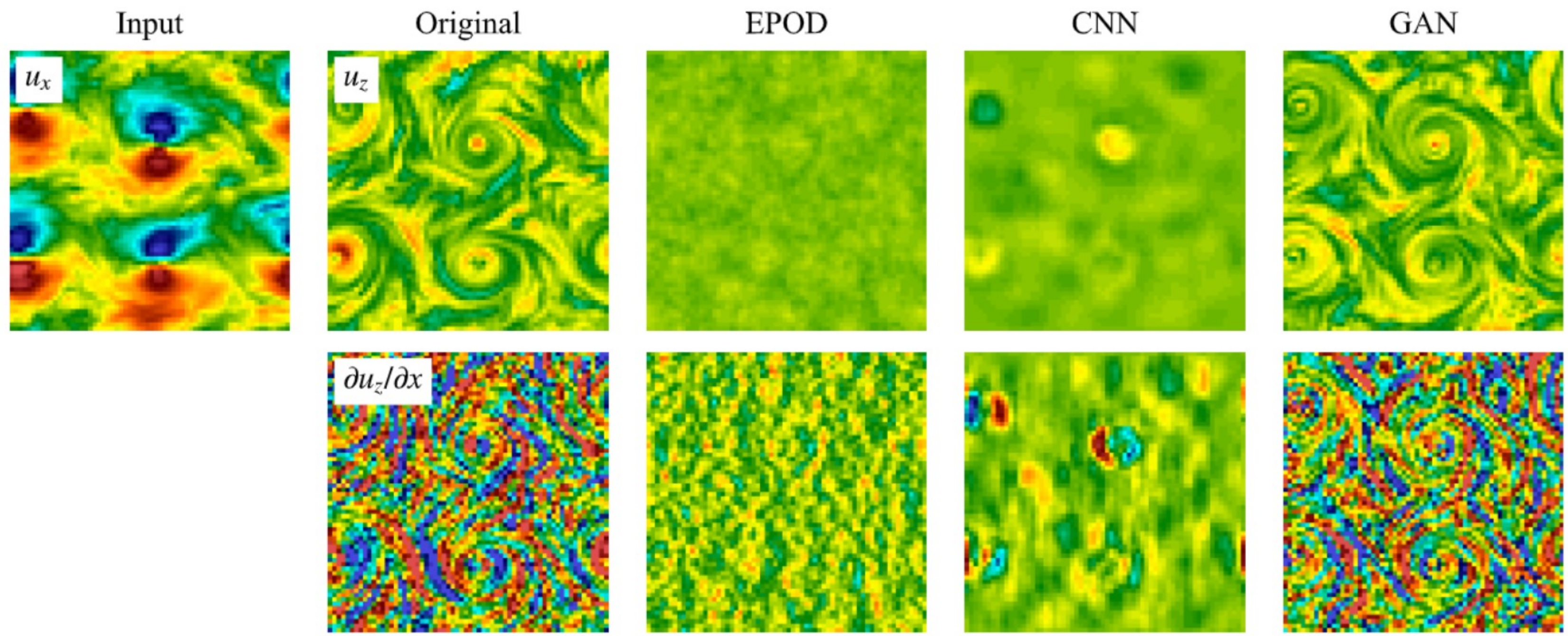


$$\int d\mathbf{x}(u^p - u^t)^2$$

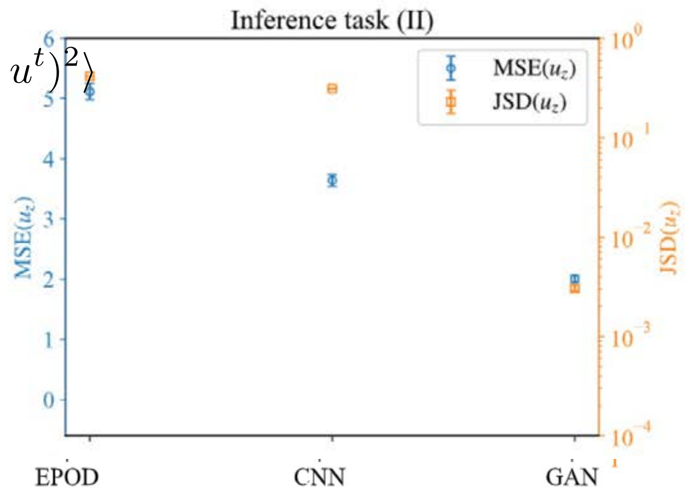




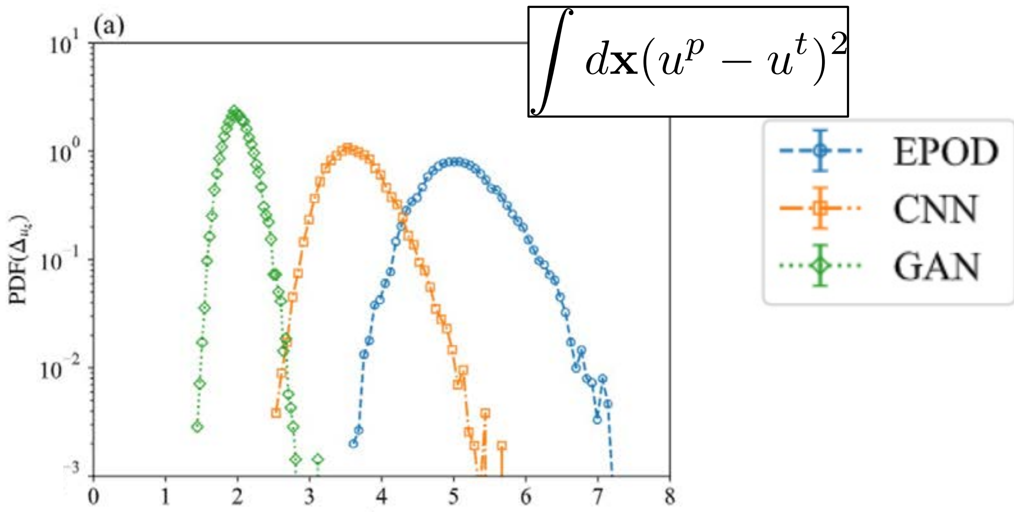
Generative Adversarial Networks to infer velocity components in rotating turbulent flows



$$\langle \int d\mathbf{x}(u^p - u^t)^2 \rangle$$



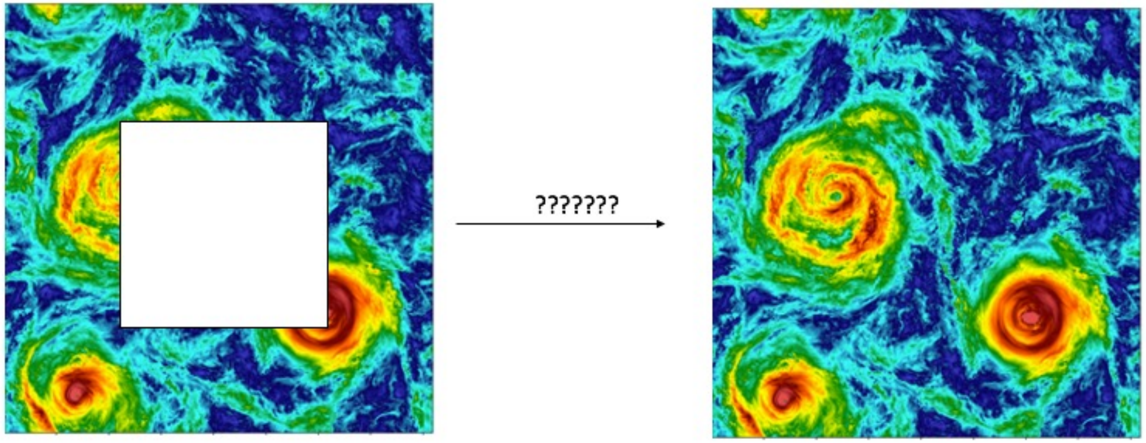
$$\int d\mathbf{x}(u^p - u^t)^2$$



NUDGING: AN **EQUATION-INFORMED UNBIASED** TOOL TO ASSIMILATE AND RECONSTRUCT TURBULENCE DATA/PHYISICS BY ADDING A DRAG TERM AGAINST PARTIAL FIELD MEASUREMENTS

EQUATIONS BASED

C.C. Lalescu, C. Meneveau and G.L. Eyink. Synchronization of Chaos in Fully Developed Turbulence. Phys. Rev. Lett. 110, 084102 (2013)
A.Farhat, E. Lunasin, and E.S. Titi. Abridged Continuous Data Assimilation for the 2d Navier-Stokes Equations Utilizing Measurements of Only One Component of the Velocity Field. J. Math. Fluid Mech. 18(1), 1 (2016)
Patricio Clark Di Leoni, Andrea Mazzino, and L.B. Synchronization to Big Data: Nudging the Navier-Stokes Equations for Data Assimilation of Turbulent Flows Phys. Rev. X 10, 011023 (2020)



$$\mathbf{v}_N = G[\mathbf{v}_{true}] \qquad \mathbf{v}_{true}$$

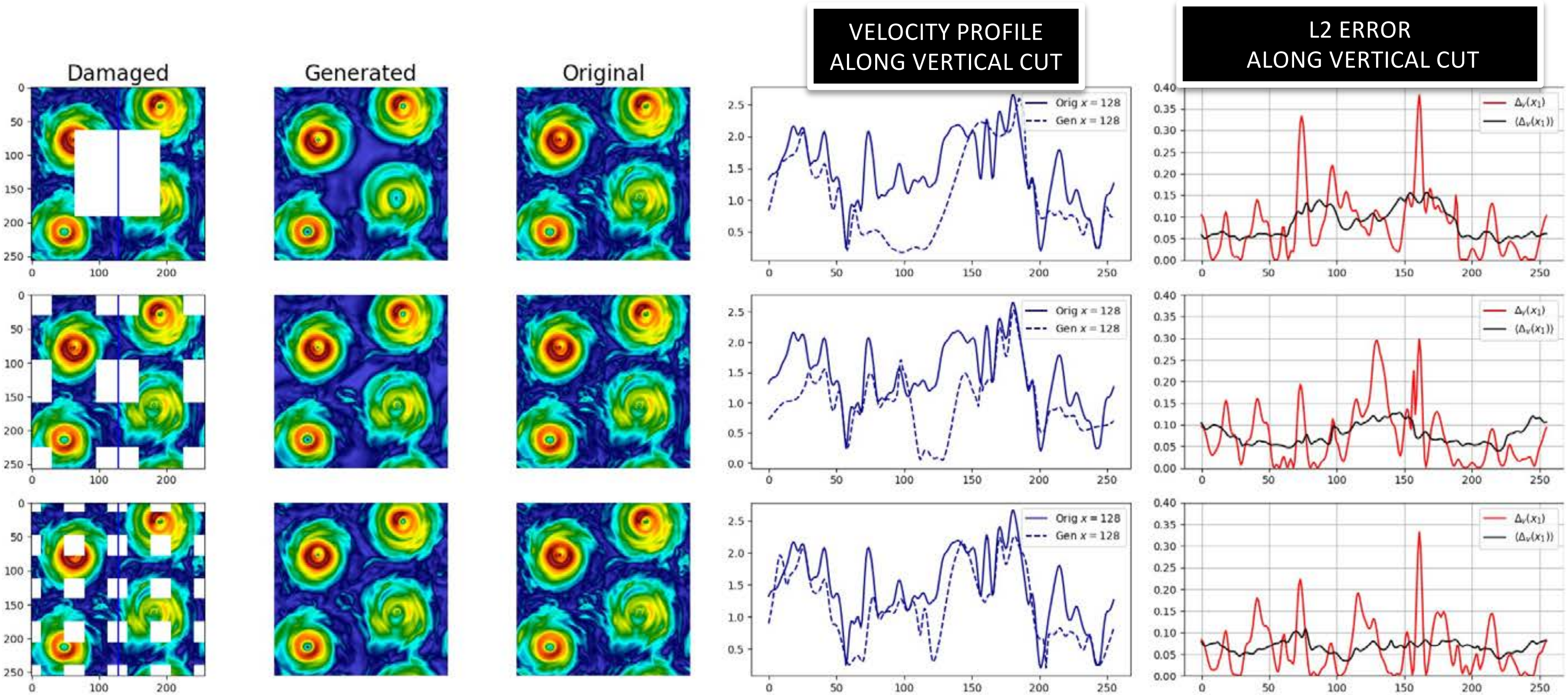
FULLY PHYSICS COMPLIANT

NEED HUGE COMPUTATIONAL RESOURCES

$$\begin{cases} \partial_t \mathbf{v} + \mathbf{v}\cdot\partial_{\mathbf{x}}\mathbf{v} + \partial_{\mathbf{x}} P - \nu\Delta\mathbf{v} = 2\mathbf{v}\times\mathbf{\Omega} + \mathcal{S}\mathbf{v} + \alpha g\hat{\mathbf{z}}T + \mathcal{F} - N(\mathbf{v}_N - \mathbf{v}) \\ \partial_x \mathbf{v} = 0 \end{cases}$$

NO NEED TO TRAIN!! NAVIER AND STOKES DID THE JOB FOR YOU: ONE CONF IS ENOUGH

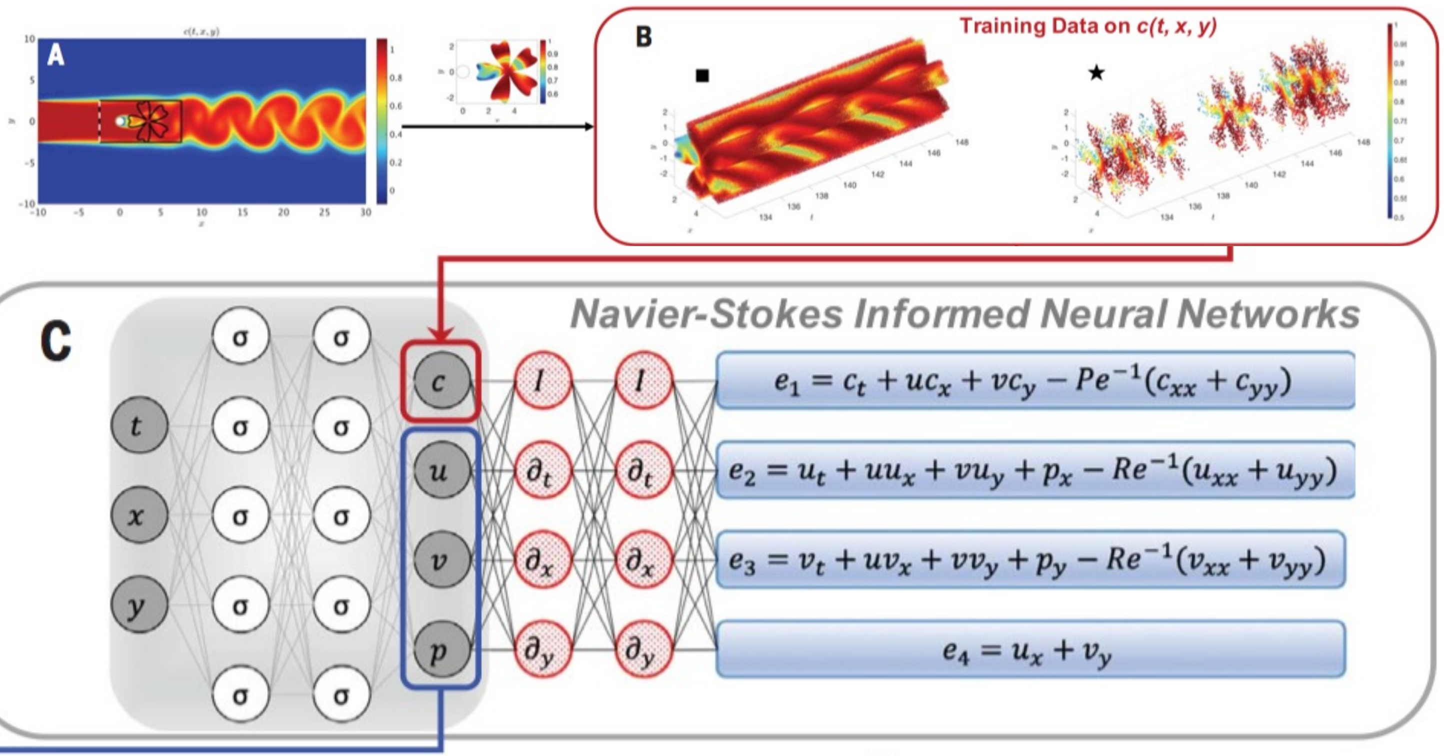
$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} \boldsymbol{v} + 2\Omega \hat{x}_3 \times \boldsymbol{v} = -\boldsymbol{\nabla} p + \nu \nabla^2 \boldsymbol{v} + \gamma(1 - \hat{M})_{x_3} \odot (\boldsymbol{v} - \boldsymbol{v}_{\mathrm{ref}})$$



Synchronization to Big Data: Nudging the Navier-Stokes Equations for Data Assimilation of Turbulent Flows
Patricio Clark Di Leoni, Andrea Mazzino, and L. B. Phys. Rev. X **10**, 011023 (2020)

Reconstruction of turbulent data with deep generative models for semantic inpainting from TURB-Rot database
M. Buzzicotti, F. Bonaccorso, P. Clark Di Leoni, and L. B. Phys. Rev. Fluids **6**, 050503 (2021)

PHYSICS INFORMED

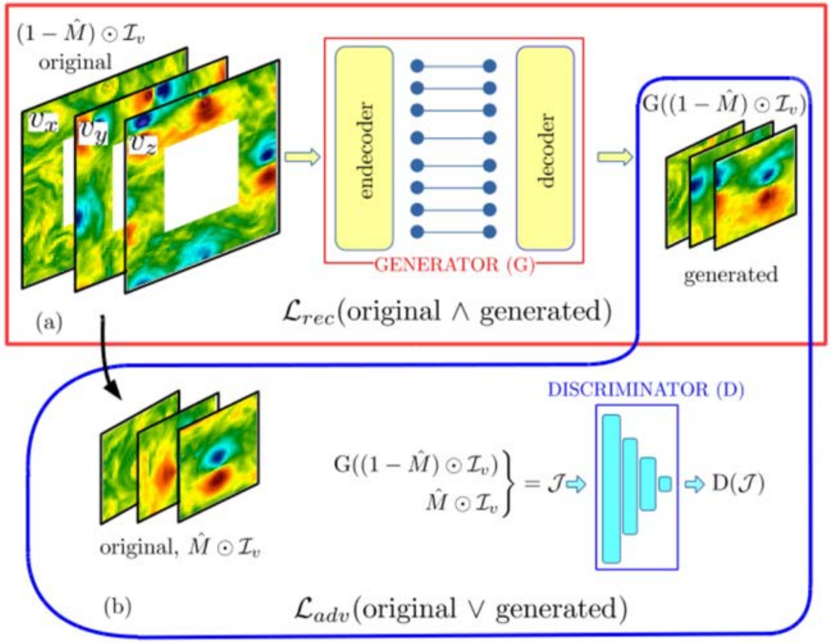


$$MSE = \frac{1}{N}\sum_{n=1}^{N}\left|c(t^n, x^n, y^n, z^n) - c^n\right|^2 + \sum_{i=1}^{5} 4 \frac{1}{M}\sum_{m=1}^{M}\left|e_i(t^m, x^m, y^m, z^m)\right|^2$$

ML-TRAINED ON A SPARSE SPATIO+TEMPORAL DATASET FOR CONCENTRATION -> INFER VELOCITY + PRESSURE -> BACK PROPAGATE FOR GRADIENTS **(AUTOMATIC DIFFERENTIATION)**-> NAVIER-STOKES

Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. M. Raissi A. Yazdani , G. E. Karniadakis , Science 367, 1026–1030 (2020)
Physics-Informed Neural Network for Ultrasound Nondestructive Quantification of Surface Breaking Cracks K. Shukla, P. Clark Di Leoni, J. Blackshire, D. Sparkman & G. E. Karniadakis *Journal of Nondestructive Evaluation* 39 (2020)

CONTEXT ENCODER



CNN-GAN

-EQUATION-FREE

GENERATION OF MISSING DATA ONLY

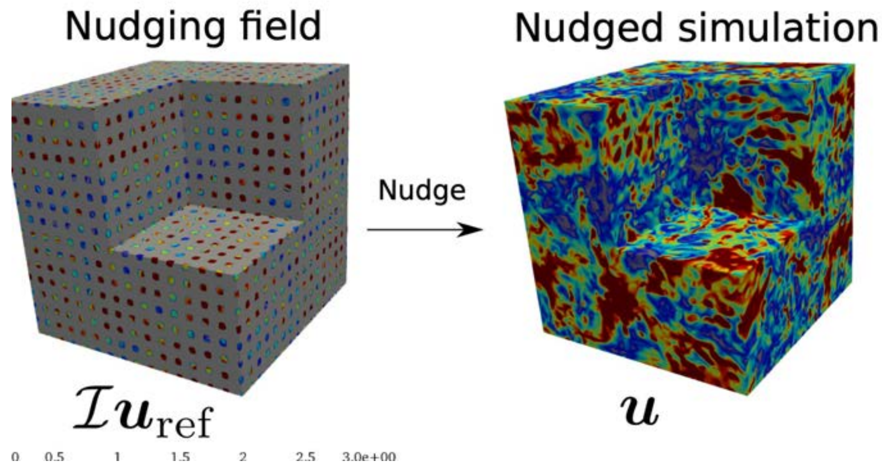
+ONCE TRAINED -> INSTANTANEOUS

+MIXED INPUT FEATURES

LINEAR CASE: EXTENDED-POD

NUDGING

$$\begin{cases} \partial_t \mathbf{v} + \mathbf{v}\cdot\partial_{\mathbf{x}}\mathbf{v} + \partial_{\mathbf{x}}P - \nu\Delta\mathbf{v} = 2\mathbf{v}\times\boldsymbol{\Omega} + \mathcal{S}\mathbf{v} + \alpha g\hat{\mathbf{z}}T + \mathcal{F} - N(\mathbf{v}_N - \mathbf{v}) \\ \partial_t T + \mathbf{v}\cdot\partial_{\mathbf{x}}T - \chi\Delta T = \mathcal{G}v_z + \mathcal{L} - N_T(T_N - T) \end{cases}$$



+EQUATION-INFORMED

GENERATION OF FRAME & MISSING DATA

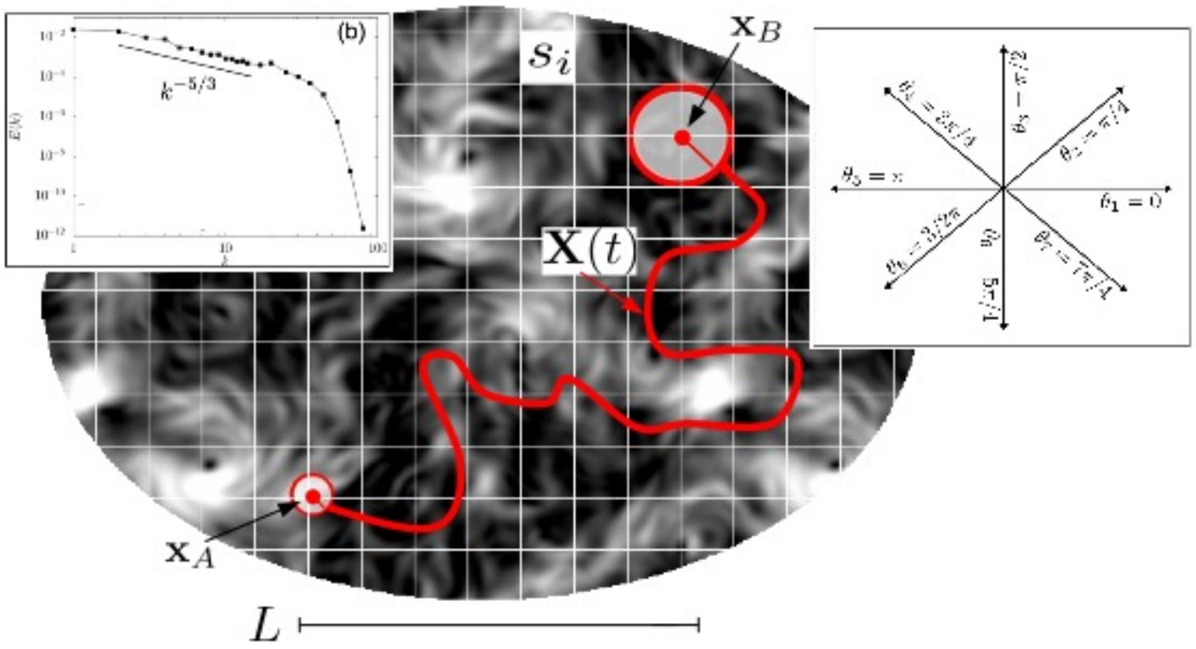
-NEW 3D DNS FOR EACH DA

-RESTRICTED INPUT FEATURES

Zermelo's problem: Optimal point-to-point navigation in 2D turbulent flows using Reinforcement Learning

L. Biferale,[1] F. Bonaccorso,[1,2] M. Buzzicotti,[1] P. Clark Di Leoni,[1,3] and K. Gustavsson[4]

Chaos: An Interdisciplinary Journal of Nonlinear Science
29.10 (2019): 103138.
arXiv preprint:1907.08591



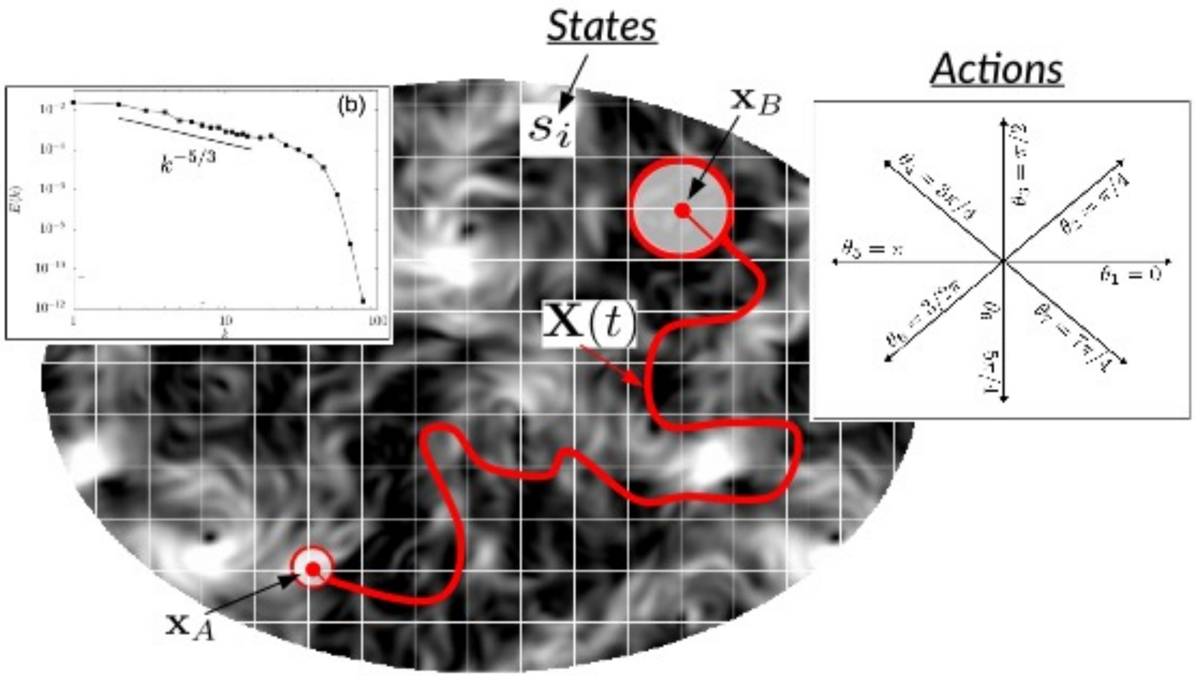
$$\begin{cases} \dot{\boldsymbol{X}}_t = \boldsymbol{u}(\boldsymbol{X}_t) + \boldsymbol{U}^{ctrl}(\boldsymbol{X}_t) \\ \boldsymbol{U}^{ctrl}(\boldsymbol{X}_t) = V_s \boldsymbol{n}(\boldsymbol{X}_t) \end{cases}$$

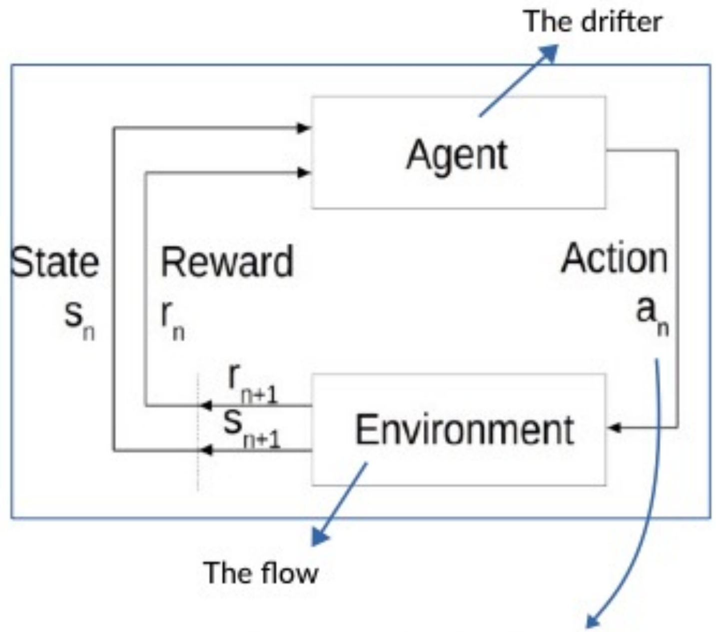
$$\boldsymbol{n}(\boldsymbol{X}_t) = (\cos[\theta_t], \sin[\theta_t]),$$

V_s → <u>Navigation speed is small compared to the velocity of the underling flow!</u>

E. Zermelo, "Über das navigationsproblem bei ruhender oder veränderlicher windverteilung," ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik **11**, 114–124 (1931).
A. E. Bryson and Y. Ho, *Applied optimal control: optimization, estimation and control* (New York: Routledge, 1975).

Reinforcement Learning; *Policy Gradient Methods*





Parameterized policy:

$$\pi(a_j|s_i,\boldsymbol{q}) = \frac{\exp h(s_i,a_j,\boldsymbol{q})}{\sum_{k=1}^{N_a}\exp h(s_i,a_k,\boldsymbol{q})}$$

Parameterized state value function:

$$\hat{v}(s_i,\boldsymbol{w}) = \sum_{j=1}^{N_s} w_j\delta_{j,i}$$

Reward

$$r_t = -\Delta t$$

$$r_{tot} = -T_{A\to B}$$

Actor-Critic algorithm

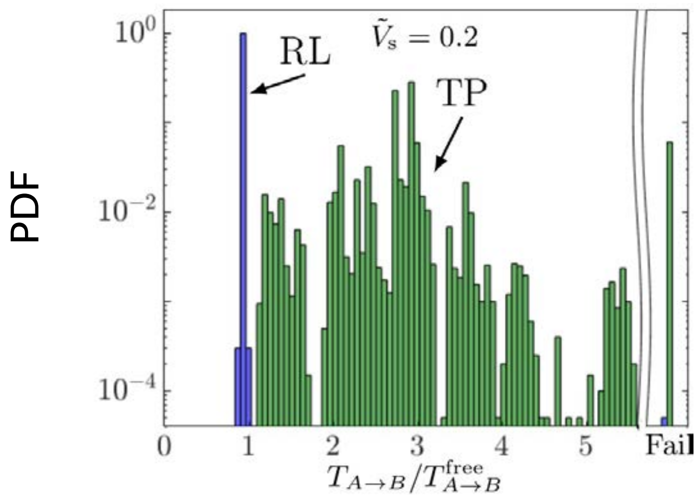
$$\begin{cases}\boldsymbol{q}_{t+\Delta t} = \boldsymbol{q}_t + \alpha_t\beta_t\nabla_{\boldsymbol{q}}\ln(\pi(a_t|s_t,\boldsymbol{q}_t)) \\ \boldsymbol{w}_{t+\Delta t} = \boldsymbol{w}_t + \alpha'_t\beta_t\nabla_{\boldsymbol{w}}\hat{v}(s_t,\boldsymbol{w}_t)\end{cases}$$

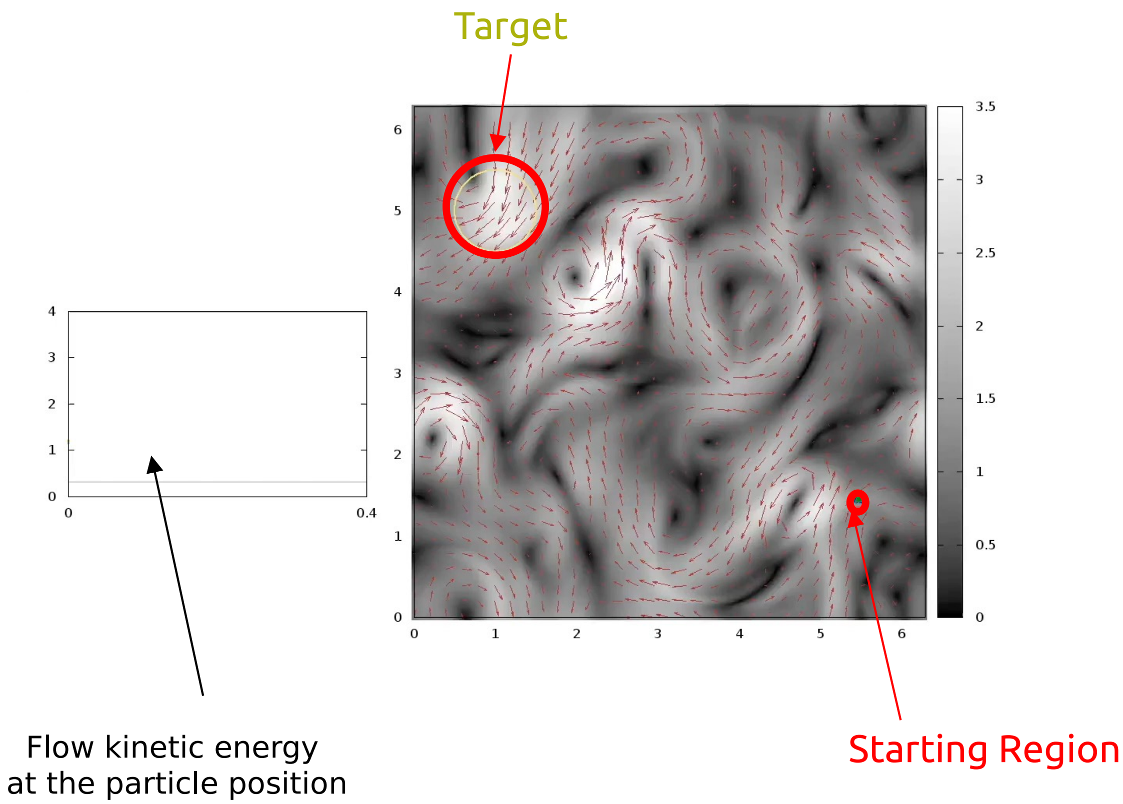
$$\beta_t = [\hat{r}_{t+\Delta_t} - \hat{v}(s_t,\boldsymbol{w}_t)] \to \text{baseline}$$

TIME-DEPENDENT 2D TURBULENT FLOWS

REINFORCEMENT LEARNING (BLUE) VS TRIVIAL POLICY (GREEN) $\tilde{V}_s = 0.2$

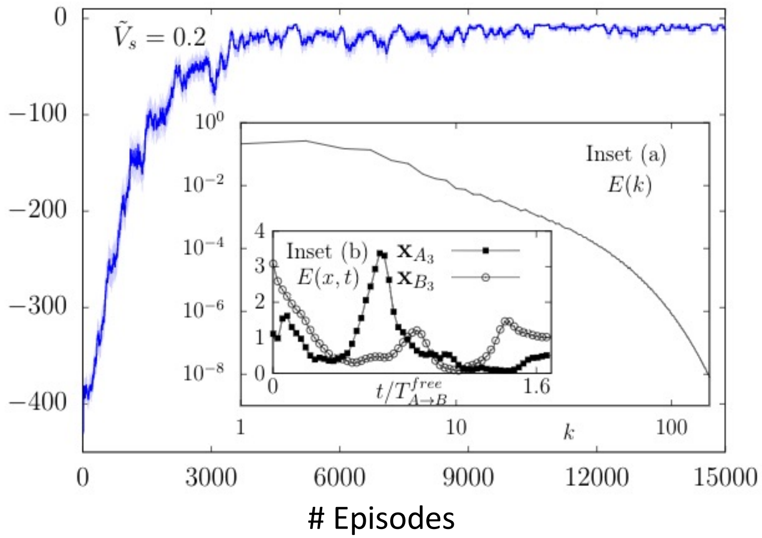
Comparison with a Trivial Policy (TP)

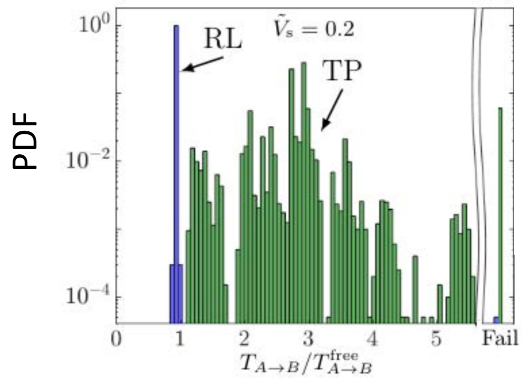




Time-Dependent 2D Turbulent Flow

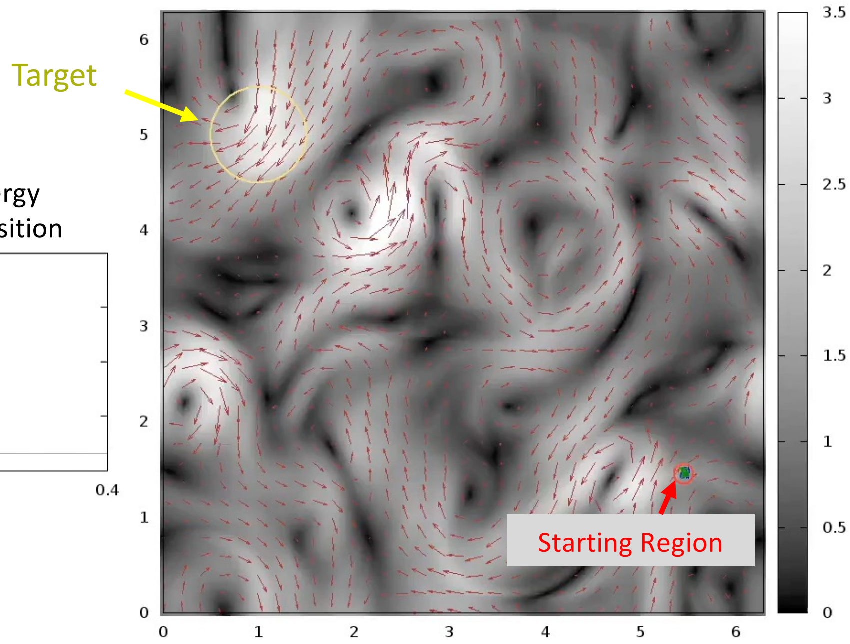
Total Reward





Flow kinetic energy at the particle position

$$\frac{V_s}{\max(u)} \sim 0.2$$



Reinforcement Learning (blue)

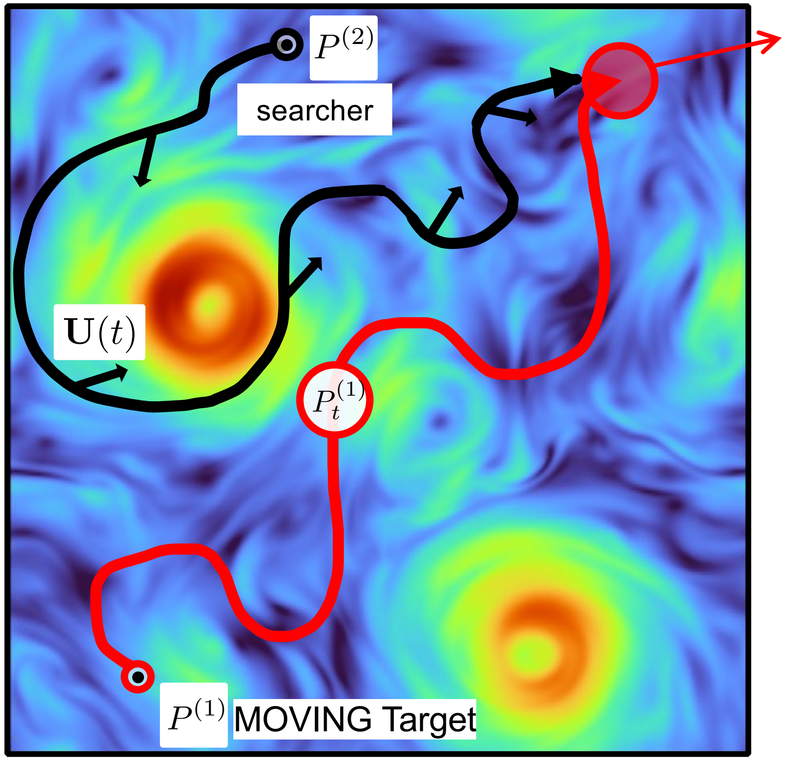
vs

Trivial Policy (green)

Catching a drifting target in turbulence via Optimal Control

C. Calascibetta,[1] L. Biferale,[1] F. Borra,[2] A. Celani,[3] and M. Cencini[4]

2 AGENTS



Goal: **minimize** the **separation/capture** in a finite time-horizon

Problem setup

$$\begin{cases} \dot{\mathbf{X}}_t^{(1)} = \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) \\ \dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}(\mathbf{X}_t^{(2)}, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s \hat{\boldsymbol{n}}(t) \end{cases} \longrightarrow \mathbf{U}(t) = ?$$

$$\hat{\boldsymbol{n}}(t) = (\cos[\theta_t], \sin[\theta_t])$$

Tools:
(1) Heuristic policies
(2) Optimal Control (OC) theory
(3) Reinforcement Learning (RL)

See also: Jaya Kumar Alageshan, Akhilesh Kumar Verma, Jérémie Bec, and Rahul Pandit Phys. Rev. E **101**, 043110 –2020

(2) **Optimal Control theory** – Pontryagin minimum principle

state variables *control variables*

Minimize $J = C_F(\mathbf{X}(t_f)) + \int_{t_0}^{t_f} dt\,[L(\mathbf{X}(t), \mathbf{U}(t), t)]$

performance index *Lagrangian function*

Imposing $\dot{\mathbf{X}}_t = \boldsymbol{f}(\mathbf{X}(t), \mathbf{U}(t), t)$

and other possible constraints,

e.g.: $\begin{cases} \mathbf{X}(t_f) = \mathbf{X}_\star\,, \quad \mathbf{X}(t_0) \le \mathbf{X}_\star\,, \\ \|\mathbf{U}(t)\|^2 = 1\,, \quad \|\mathbf{U}(t)\|^2 \le 1\,, \text{exc.} \end{cases}$

- **Model based** and analytical tool
- Perfect knowledge required

$\|\boldsymbol{R}_{t_0}\| \sim \frac{V_s}{\lambda_{lyapunov}}$ border of controllability

In our case:

capture's distance

$\|\boldsymbol{R}^\star\| = \|\boldsymbol{R}_{t_0}\|/100$

Minimize $J = \|\boldsymbol{R}_{t_f}\|^2 + c\int_{t_0}^{t_f} dt\,\theta(\|\boldsymbol{R}_{t_f}\|^2 - \|\boldsymbol{R}^\star\|^2)$

Imposing (2) and the control constraint $\|\hat{\boldsymbol{n}}(t)\|^2 = 1$

$$\begin{cases} \dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t)\,, \\ \mathbf{U}(t) = V_S \hat{\boldsymbol{n}}(t)\,. \end{cases} \quad (2)$$

LINEAR REGIME

Optimal Control theory to **minimize Lagrangian particles dispersion** in turbulent flows

Constrained performance index:

$$\tilde{J} = \|\boldsymbol{R}_{t_f}|^2 + \int_{t_0}^{t_f} dt\,\{c[\theta(\|\boldsymbol{R}_t\|^2 - \|\boldsymbol{R}^\star\|^2)] + \underbrace{\boldsymbol{\lambda}^{\mathrm{T}}(t)[\nabla \boldsymbol{v}_t \boldsymbol{R}_t + V_s \hat{\boldsymbol{n}}(t) - \dot{\boldsymbol{R}}] + \mu(t)(1 - \|\hat{\boldsymbol{n}}(t)\|^2)\}}_{\text{Constrained minimization problem}}$$

Constrained minimization problem

Integrating by parts,

$$\tilde{J} = \|\boldsymbol{R}_{t_f}\|^2 - \boldsymbol{\lambda}^{\mathrm{T}}(t_f)\boldsymbol{R}_{t_f} + \boldsymbol{\lambda}^{\mathrm{T}}(t_0)\boldsymbol{R}_{t_0} + \int_{t_0}^{t_f} dt\,[H(\boldsymbol{R}_t, \boldsymbol{\lambda}(t), \hat{\boldsymbol{n}}(t), \mu(t), t) + \dot{\boldsymbol{\lambda}}^{\mathrm{T}}\boldsymbol{R}_t]\,,$$

$$H(\boldsymbol{R}_t, \boldsymbol{\lambda}(t), \hat{\boldsymbol{n}}(t), \mu(t), t) = c[\theta(\|\boldsymbol{R}_t\|^2 - \|\boldsymbol{R}^\star\|^2)] + \boldsymbol{\lambda}^{\mathrm{T}}(t)[\nabla \boldsymbol{v}_t \boldsymbol{R}_t + V_s \hat{\boldsymbol{n}}(t)] + \mu(t)(1 - \|\hat{\boldsymbol{n}}(t)\|^2).$$

Hamiltonian function

Consider variation in $\tilde{J}$:

$$\delta\tilde{J} = \Big[(\boldsymbol{R}_t - \boldsymbol{\lambda}^{\mathrm{T}}(t))\delta\boldsymbol{R}\Big]_{t=t_f} + \Big[\boldsymbol{\lambda}^{\mathrm{T}}(t)\delta\boldsymbol{R}\Big]_{t=t_0} + \int_{t_0}^{t_f} dt\,\Big\{\Big[\frac{\partial H}{\partial \boldsymbol{R}} + \dot{\boldsymbol{\lambda}}^{\mathrm{T}}\Big]\delta\boldsymbol{R} + \frac{\partial H}{\partial \hat{\boldsymbol{n}}}\delta\hat{\boldsymbol{n}}\Big\}$$

Euler-Lagrange equations:

$$\begin{cases} \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{R}} = -2c\,\boldsymbol{R}_t\delta(\|\boldsymbol{R}_t\|^2 - \|\boldsymbol{R}^\star\|^2) - (\nabla \boldsymbol{v}_t)^{\mathrm{T}}\boldsymbol{\lambda}(t)\,, \\ \boldsymbol{\lambda}(t_f) = 2\boldsymbol{R}_{t_f}\,, \\ \frac{\partial H}{\partial \hat{\boldsymbol{n}}} = 0 \implies \hat{\boldsymbol{n}}(t) = \frac{V_s\boldsymbol{\lambda}(t)}{2\mu(t)} = -\frac{\boldsymbol{\lambda}(t)}{\|\boldsymbol{\lambda}(t)\|}\,. \end{cases}$$

$$\begin{cases} \dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathrm{U}(t)\,, \\ \boldsymbol{R}_{t_0} = \text{given}\,, \\ \mathrm{U}(t) = V_S\hat{\boldsymbol{n}}(t)\,. \end{cases}$$

A. E. Bryson and Y. Ho, **Applied optimal control: optimization, estimation and control** (New York: Routledge, 1975).

Optimal Control theory to **minimize Lagrangian particles dispersion** in turbulent flows

Euler-Lagrange equations:

$$\begin{cases} \dot{\boldsymbol{\lambda}} = -\frac{\partial H}{\partial \boldsymbol{R}} = -2c\,\boldsymbol{R}_t \delta(\|\boldsymbol{R}_t\|^2 - \|\boldsymbol{R}^\star\|^2) - (\nabla \boldsymbol{v}_t)^{\mathrm{T}} \boldsymbol{\lambda}(t)\,, \\ \boldsymbol{\lambda}(t_f) = 2\boldsymbol{R}_{t_f}\,, \\ \frac{\partial H}{\partial \hat{\boldsymbol{n}}} = 0 \implies \hat{\boldsymbol{n}}(t) = \frac{V_s \boldsymbol{\lambda}(t)}{2\mu(t)} = -\frac{\boldsymbol{\lambda}(t)}{\|\boldsymbol{\lambda}(t)\|}\,. \end{cases}$$

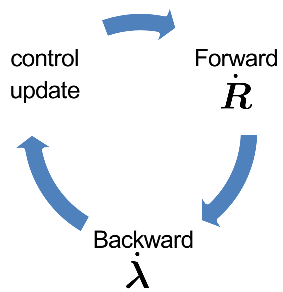
backward integration

$$\begin{cases} \dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t)\,, \\ \boldsymbol{R}_{t_0} = \text{ given}\,, \\ \mathbf{U}(t) = V_S \hat{\boldsymbol{n}}(t)\,. \end{cases}$$

Forward integration

computationally heavy

It requires iterative searching with backward and forward integration such as to identify the optimal control



Lenhart, S., & Workman, J.T. (2007). **Optimal Control Applied to Biological Models** (1st ed.). Chapman and Hall/CRC.

$$\begin{cases} \dot{\mathbf{X}}_t^{(1)} = \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) \\ \dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}(\mathbf{X}_t^{(2)}, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s \hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^{(2)} - \mathbf{X}_t^{(1)} \\ L = \text{characteristic scale of the flow} \end{cases}$$

(1) (semi) Heuristic policies

Trivial Policy: constantly chooses the direction that points towards the moving target

$$\hat{\boldsymbol{n}}(t) = -\hat{\boldsymbol{R}}_t$$

Surfing policy*: valid at large scales, i.e., $\|\boldsymbol{R}_t\| \gg L$. Based on a free parameter τ_s .

Perturbative policy: valid at small scales, i.e. $\|\boldsymbol{R}_t\| \ll L$. Based on a free parameter τ_p .

*Monthiller, Rémi, et al. **Surfing on Turbulence: A Strategy for Planktonic Navigation**. *Phys. Rev. Lett.* ***129****, 064502* (2022)

$$\begin{cases} \dot{\mathbf{X}}_t^{(1)} = \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) \\ \dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}(\mathbf{X}_t^{(2)}, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s \hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^{(2)} - \mathbf{X}_t^{(1)} \\ L = \text{characteristic scale of the flow} \end{cases}$$

(1) (semi) Heuristic policies

Surfing policy* - derivation

- Approximate linearly the underlying flow, $\boldsymbol{v}(\mathbf{X}_t^{(2)}, t)$ for $t_0 < t < \tau_s$; *(Assuming constant gradients for a time τ_s)*

$$\dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}_{t_0} + (\nabla \boldsymbol{v})_{t_0} \cdot (\mathbf{X}_t^{(2)} - \mathbf{X}_{t_0}^{(2)}) + \left(\frac{\partial \boldsymbol{v}}{\partial t}\right)_{t_0} (t - t_0) + \mathbf{U}(t) \,,$$

- Find $\mathbf{U}(t)$ such that $-(\mathbf{X}_{\tau_s}^{(2)} - \mathbf{X}_{t_0}^{(2)}) \cdot \hat{\boldsymbol{R}}_{t_0}$ is maximum;

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \hat{\boldsymbol{R}}_{t_0}}{\left\| \left[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \hat{\boldsymbol{R}}_{t_0} \right\|}$$

(Assuming constant the direction $\hat{\boldsymbol{R}}_t$ for a time τ_s)

↓

$$\|\boldsymbol{R}_t\| \gg L$$

- Numerically optimize the free parameter τ_s.

*Monthiller, Rémi, et al. **Surfing on Turbulence: A Strategy for Planktonic Navigation**. *Phys. Rev. Lett.* ***129***, *064502* (2022)

Surfing policy* - derivation

$$\dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}_{t_0} + (\nabla \boldsymbol{v})_{t_0} \cdot \left(\mathbf{X}_t^{(2)} - \mathbf{X}_{t_0}^{(2)}\right) + \left(\frac{\partial \boldsymbol{v}}{\partial t}\right)_{t_0} (t - t_0) + V_s \hat{\boldsymbol{n}}(t)$$

$$\mathbf{X}_{\tau_s}^{(2)} = \mathbf{X}_{t_0}^{(2)} + \left[e^{\tau_s (\nabla \boldsymbol{v})_{t_0}} - \mathbb{I}\right] \cdot (\nabla \boldsymbol{v}_{t_0})^{-1} \cdot \left[\boldsymbol{v}_{t_0} + (\nabla \boldsymbol{v}_{t_0})^{-1} \cdot \left(\frac{\partial \boldsymbol{v}}{\partial t}\right)_{t_0}\right] - \\ - \tau_s (\nabla \boldsymbol{v}_{t_0})^{-1} \cdot \left(\frac{\partial \boldsymbol{v}}{\partial t}\right)_{t_0} + V_s \int_{t_0}^{\tau_s} dt\, e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}} \cdot \hat{\boldsymbol{n}}(t)$$

find $\hat{\boldsymbol{n}}(t)$ such as $-\left(\mathbf{X}_{\tau_s}^{(2)} - \mathbf{X}_{t_0}^{(2)}\right) \cdot \hat{\boldsymbol{R}}_{t_0}$ is maximum,

means find $\hat{\boldsymbol{n}}(t)$ such that $-\int_{t_0}^{\tau_s} dt \underbrace{\left[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}} \cdot \hat{\boldsymbol{n}}(t) \cdot \hat{\boldsymbol{R}}_{t_0}\right]}_{-[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}}]^{\mathrm{T}} \cdot \hat{\boldsymbol{R}}_{t_0} \cdot \hat{\boldsymbol{n}}(t)}$ is maximum (i.e. by maximizing the integrand).

$\hat{\boldsymbol{n}}(t)$ must be collinear to $-[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}}]^{\mathrm{T}} \cdot \hat{\boldsymbol{R}}_{t_0}$ →

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \hat{\boldsymbol{R}}_{t_0}}{\left\| \left[e^{(\tau_s - t)\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot \hat{\boldsymbol{R}}_{t_0} \right\|}$$

*Monthiller, Rémi, et al. **Surfing on Turbulence: A Strategy for Planktonic Navigation**. *Phys. Rev. Lett.* ***129**, 064502* (2022)

$$\begin{cases} \dot{\mathbf{X}}_t^{(1)} = \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) \\ \dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}(\mathbf{X}_t^{(2)}, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s \hat{\boldsymbol{n}}(t) \end{cases}$$

$$\mathbf{U}(t) = ?$$

$$\begin{cases} \boldsymbol{R}_t = \mathbf{X}_t^{(2)} - \mathbf{X}_t^{(1)} \\ L = \text{ characteristic scale of the flow} \end{cases}$$

(1) (semi) Heuristic policies

Perturbative policy - derivation

- Consider linearity between the two agents, i.e., $\boldsymbol{v}(\mathbf{X}_t^{(2)}, t) \simeq \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) + \nabla \boldsymbol{v}_t \boldsymbol{R}_t\,,$ $\longrightarrow \|\boldsymbol{R}_t\| \ll L$

$$\dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t) \rightarrow \quad \boldsymbol{R}_{\tau_p} = \underbrace{e^{[(\nabla \boldsymbol{v})_{t_0} \tau_p]} \boldsymbol{R}_{t_0}} + V_s \int_{t_0}^{\tau_p} dt\, e^{[(\nabla \boldsymbol{v})_{t_0} (\tau_p - t)]} \hat{\boldsymbol{n}}(t)\,;$$

(Assuming constant gradients for a time τ_p)

- Find $\mathbf{U}(t)$ such that $\boldsymbol{R}_{\tau_p} \cdot \boldsymbol{R}_{\tau_p}^{free}$ is minimum;

$$\hat{\boldsymbol{n}}(t) = -\frac{\left[e^{(\tau_p - t)\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot e^{(\nabla \boldsymbol{v})_{t_0} \tau_p} \cdot \hat{\boldsymbol{R}}_{t_0}}{\left\| \left[e^{(\tau_p - t)\nabla \boldsymbol{v}_{t_0}}\right]^{\mathrm{T}} \cdot e^{(\nabla \boldsymbol{v})_{t_0} \tau_p} \cdot \hat{\boldsymbol{R}}_{t_0} \right\|}$$

- Numerically optimize the free parameter τ_p.

Optimal Control vs heuristic policies at **small scales**

Velocity field*

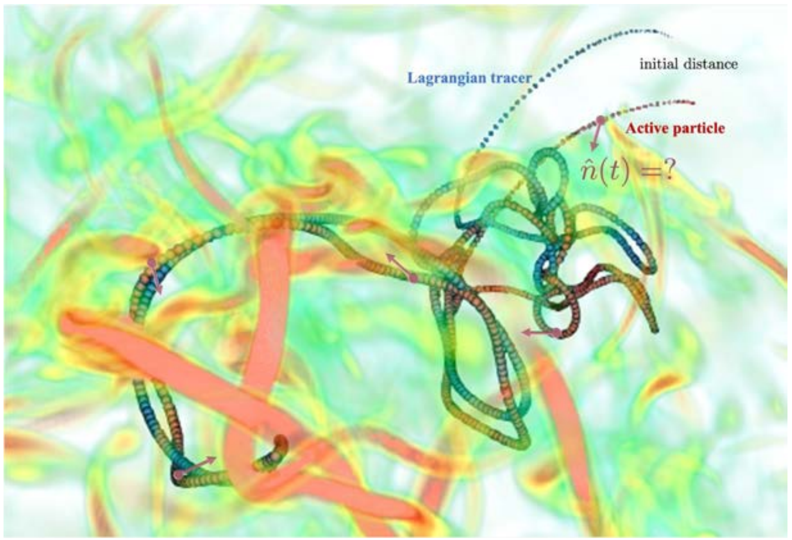
3D Direct Numerical Simulations $N = 1024^3$

NSEs $$\begin{cases} \partial_t \boldsymbol{v} = -\nabla \mathrm{p} - (\boldsymbol{v}\cdot\nabla)\boldsymbol{v} + \nu\nabla^2\boldsymbol{v} + \mathbf{F}\,, \\ \nabla\cdot\boldsymbol{v} = 0 \end{cases}$$

↓ homogeneous and isotropic forcing

DNS parameters

$$\eta_k = 0.0043$$
$$\tau_\eta = 0.023$$
$$Re \simeq 17000$$



(1) LINEAR REGIME $\|\boldsymbol{R}_{t_0}\| < \eta_k$

$$\begin{cases} \dot{\mathbf{X}}_t^{(1)} = \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) \\ \dot{\mathbf{X}}_t^{(2)} = \boldsymbol{v}(\mathbf{X}_t^{(2)}, t) + \mathbf{U}(t) \\ \mathbf{U}(t) = V_s \hat{\boldsymbol{n}}(t) \end{cases}$$

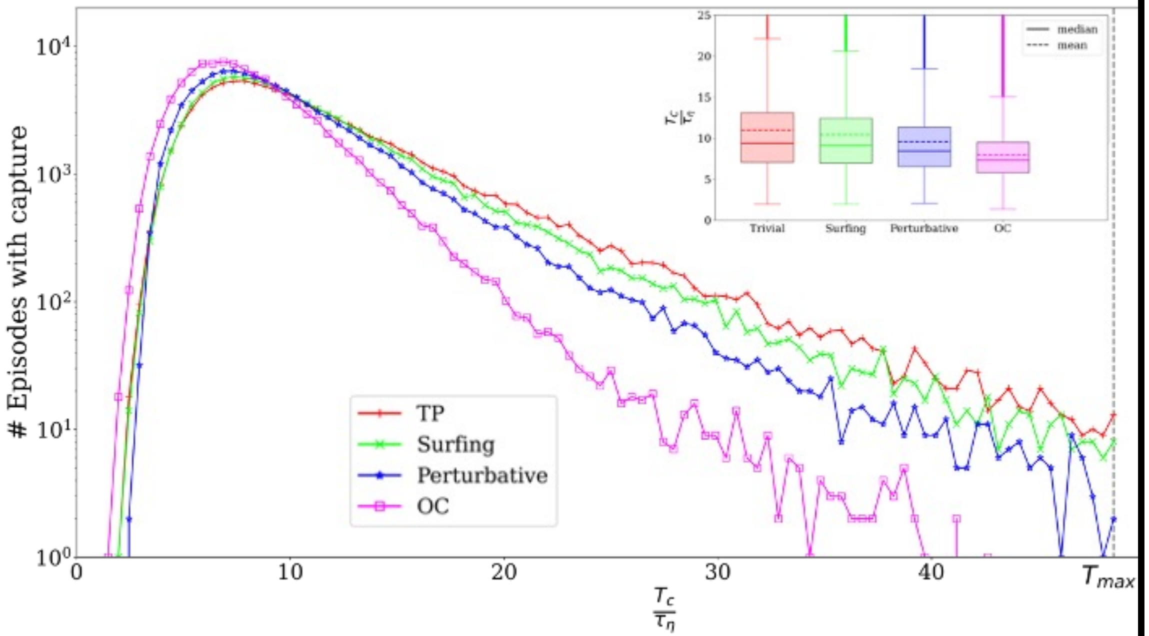
$$\boldsymbol{v}(\mathbf{X}_t^{(2)}, t) \simeq \boldsymbol{v}(\mathbf{X}_t^{(1)}, t) + \nabla\boldsymbol{v}_t \boldsymbol{R}_t$$

$$\dot{\boldsymbol{R}}_t = \nabla\boldsymbol{v}_t\boldsymbol{R}_t + \mathbf{U}(t)$$

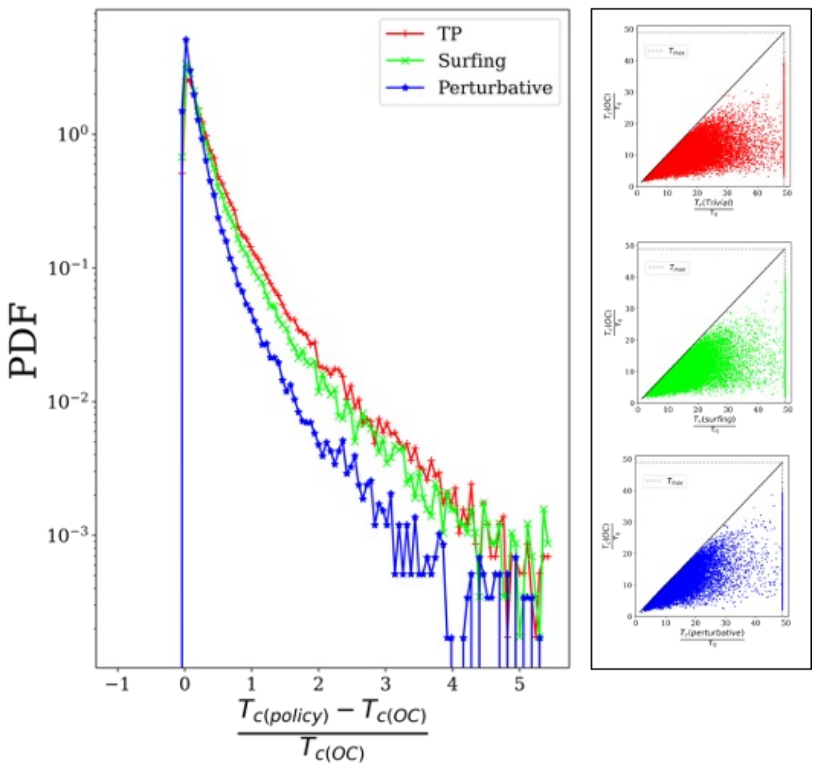
Optimal Control vs heuristic policies in linear regime

$$\dot{\boldsymbol{R}}_t = \nabla \boldsymbol{v}_t \boldsymbol{R}_t + \mathbf{U}(t)$$

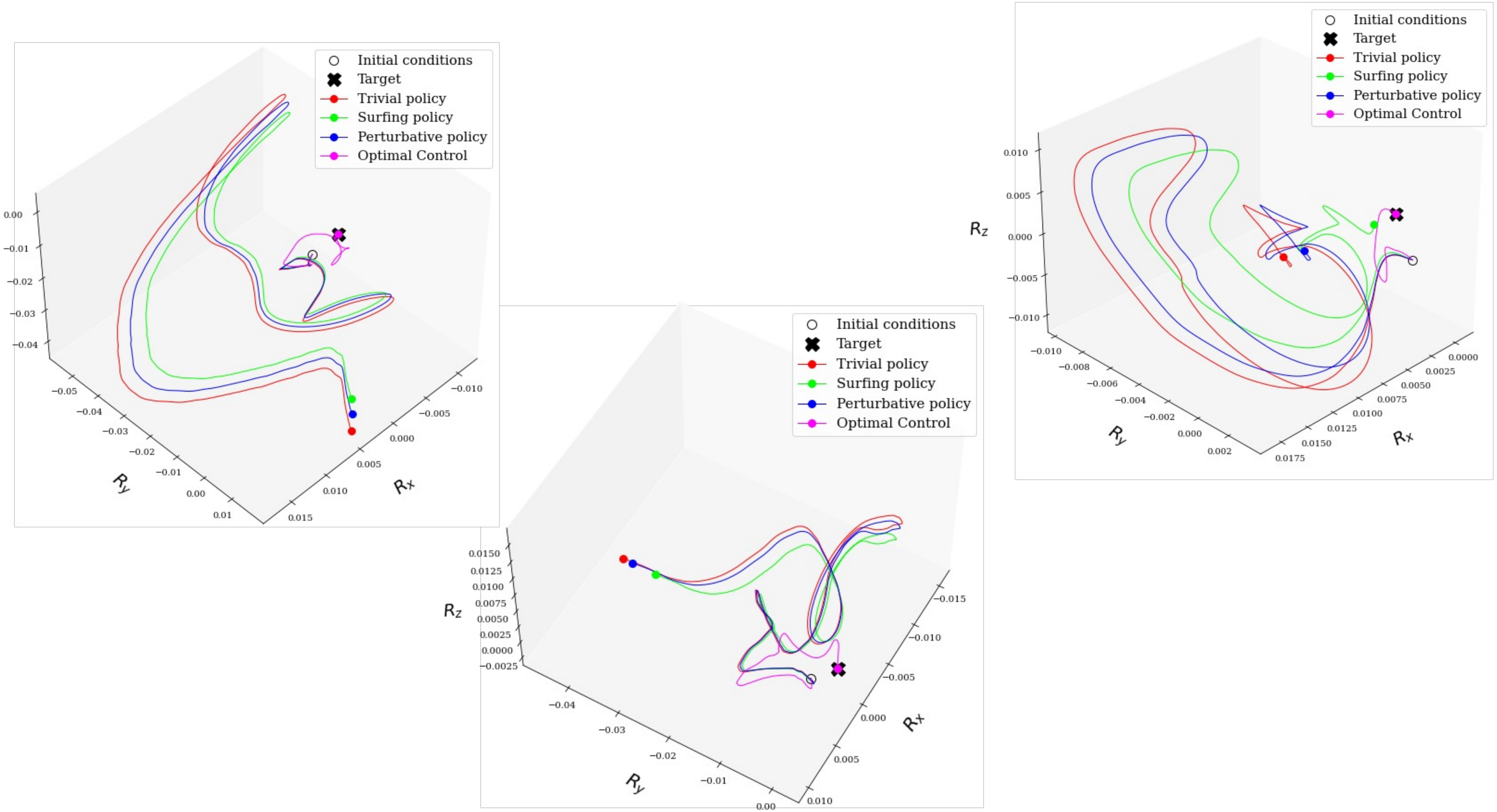
T_c = **Capture** time: (time of arrival at the desired distance)



PDF of normalized capture time



PRELIMINARY UNPUBLISHED


Initial conditions
Target
Trivial policy
Surfing policy
Perturbative policy
Optimal Control
R_x
R_y
R_z

pros & cons

Optimal Control

+ It is optimized
- It is model based and needs perfect information from the environment
- It is sensitive to variation of the initial condition
- It is difficult to consider a decision time in the control variable

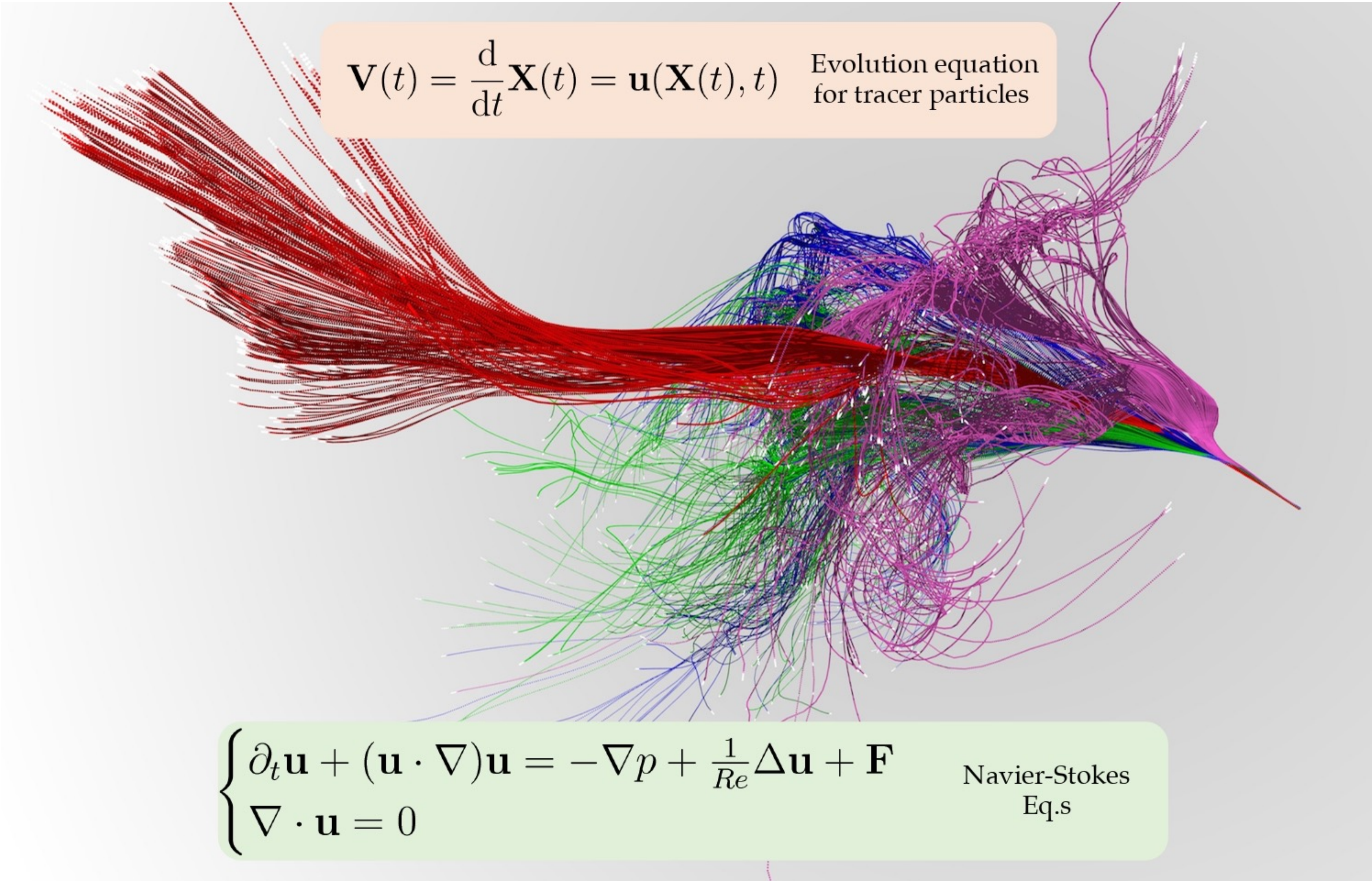
Heuristic policies

- They are not optimized
+ They need only partial information
+ They are stable wrt variation of the initial condition
+ They work also with a discrete decision time

Next step: Reinforcement Learning

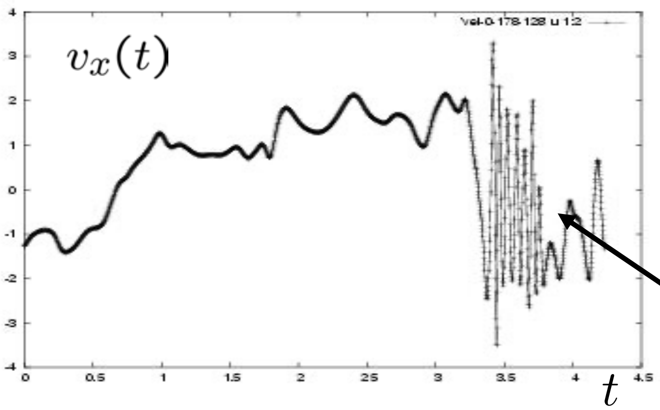
+ It is optimized
+ It is model free
+ It needs partial information
- It is data-hungry

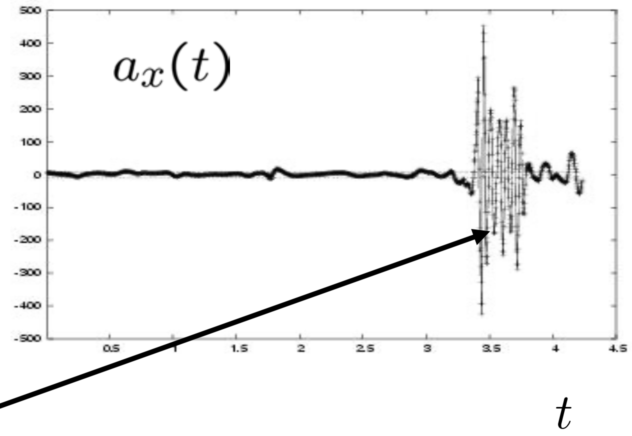
LAGRANGIAN TURBULENCE

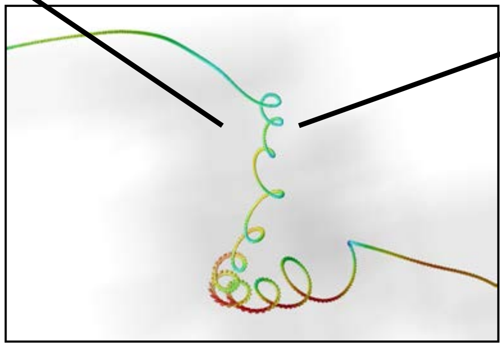


Biferale et al. (J. Fluid Mech., 2014, vol. 757, pp. 550–572)

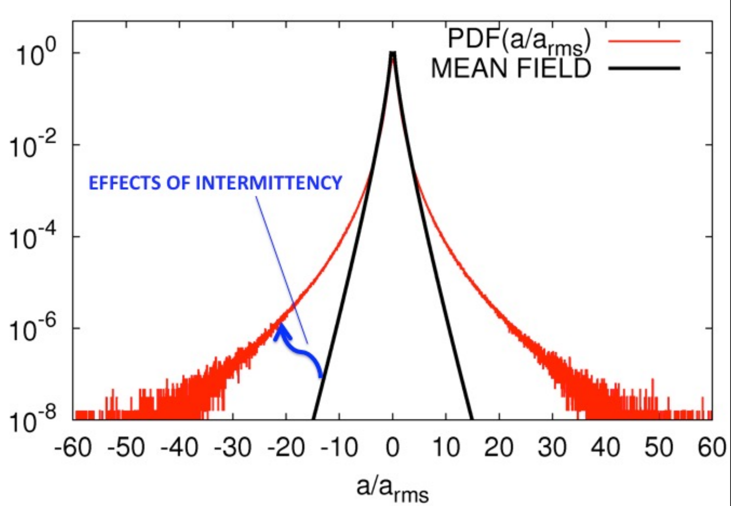
$$\begin{cases} \mathbf{a} = \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \nu \Delta \mathbf{u} + \mathbf{f} \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$







EXTREME EVENTS



ACCELERATION PROBABILITY DISTRIBUTION FUNCTION (PDF) AT RE ~ 10^5 [Bi04] COMPARED WITH THE PREDICTION FROM MEAN FIELD (KOLMOGOROV THEORY)

L.B., G Boffetta, A Celani, A Lanotte, F Toschi. Particle trapping in three-dimensional fully developed turbulence Physics of Fluids 17 (2), 021701 (2005)

La Porta, G.A. Voth, A.M. Crawford, J. Alexander et al. Fluid particle accelerations in fully developed turbulence. Nature, 409(6823), 1017 (2001)

N. Mordant, P. Metz, O. Michel and J.F. Pinton. Measurement of Lagrangian velocity in fully developed turbulence. Phys. Rev. Lett. 87(21), 214501 (2001)

F. Toschi and E. Bodenschatz. Lagrangian Properties of Particles in Turbulence. Annu. Rev. Fluid Mech. 41, 375 (2009)

L. B., G. Boffetta, A. Celani, B. Devenish, A. Lanotte and F. Toschi. Multifractal statistics of Lagrangian velocity and acceleration in turbulence. Phys. Rev. Lett. 93, 064502 ,(2004).

Generation of Lagrangian trajectories

GAN

"A Style-Based Generator Architecture for Generative Adversarial Networks" Karras, Laine & Aila, NVIDIA, 2019

Diffusion Models

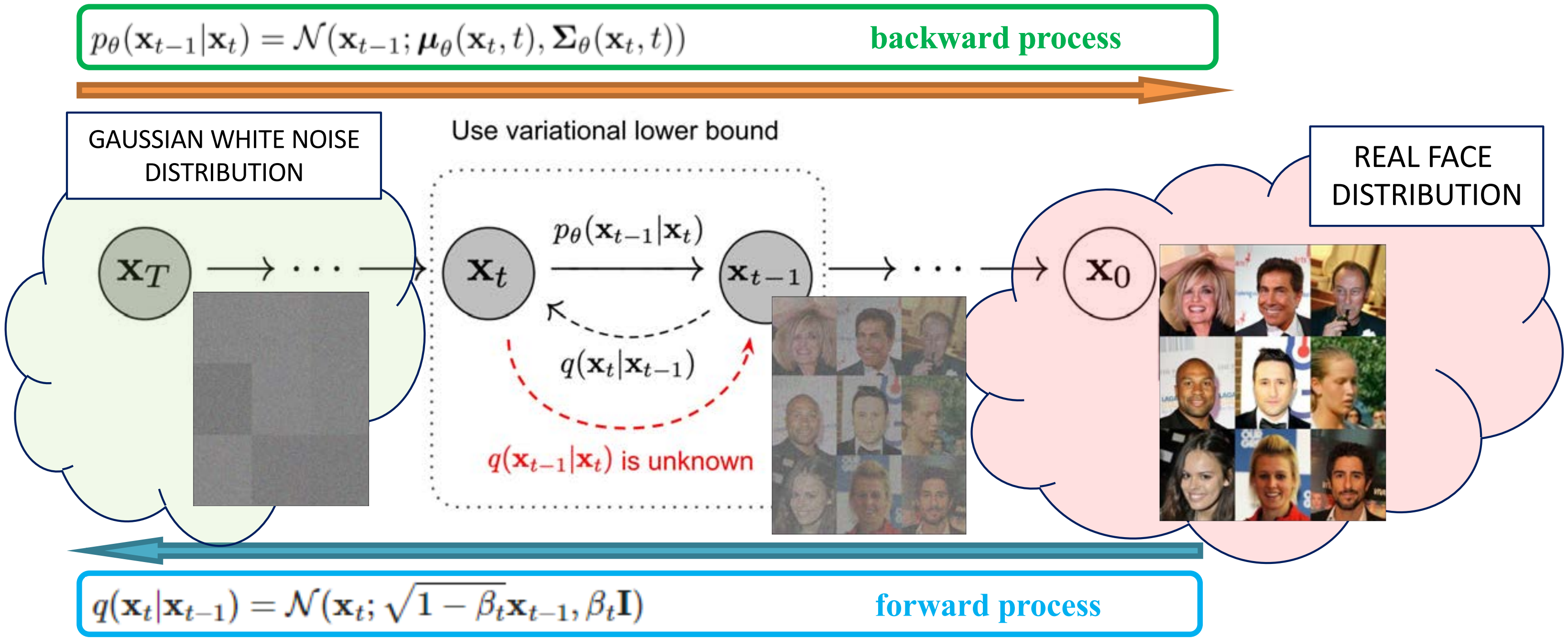
"Diffusion Models Beat GANs on Image Synthesis" Dhariwal & Nichol, OpenAI, 2021

Diffusion Models

'Synthetica Lagrangian Turbulence: all you need ia Diffusion Models' T. Li, L.B, F. Bonaccorso, M. Scarpolini and M. Buzzicotti (in preparation, 2023)

Diffusion Model	GAN
Single precision 4GPUs: => 9hrs => 20480 samples 16GPUs: => 24hrs => 204800 samples	Double precision (to be checked) 1GPU: 2s => 20480 predictions

Training objective:
Maximize the likelihood



Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021

Diffusion Models

'Synthetica Lagrangian Turbulence: all you need is Diffusion Models' T. Li, L.B, F. Bonaccorso, M. Scarpolini and M. Buzzicotti (in preparation, 2023)

Diffusion Model

Single precision
4GPUs: => 9hrs => 20480 samples
16GPUs: => 24hrs => 204800 samples

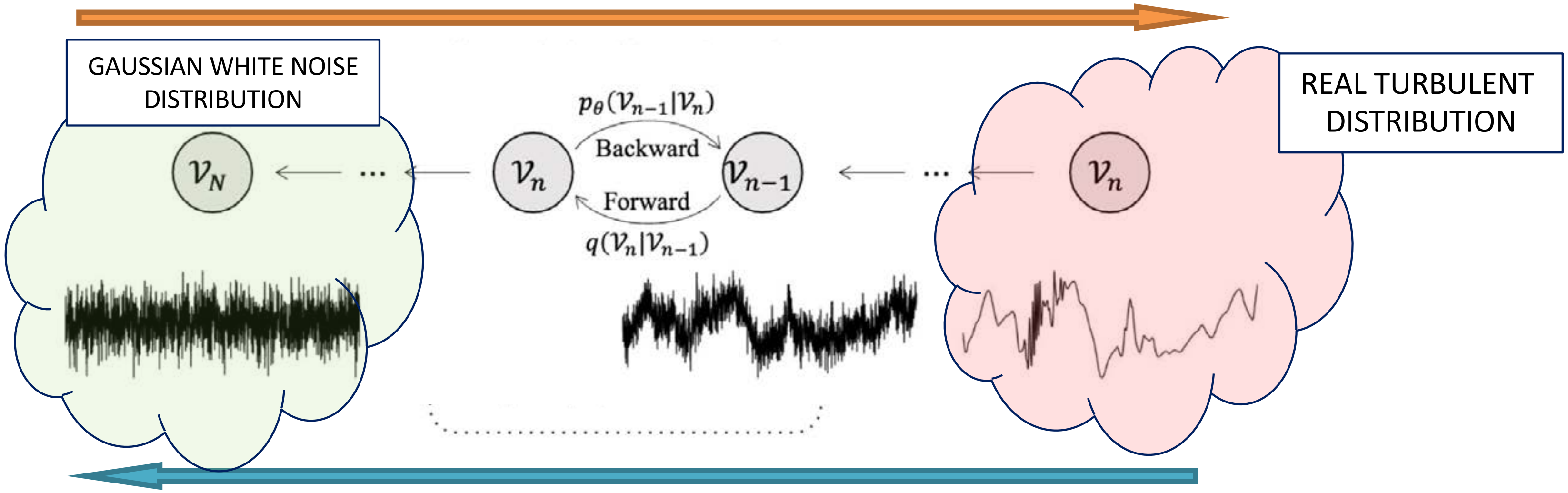
GAN

Double precision (to be checked)
1GPU: 2s => 20480 predictions

Training objective:
Maximize the likelihood

$$p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

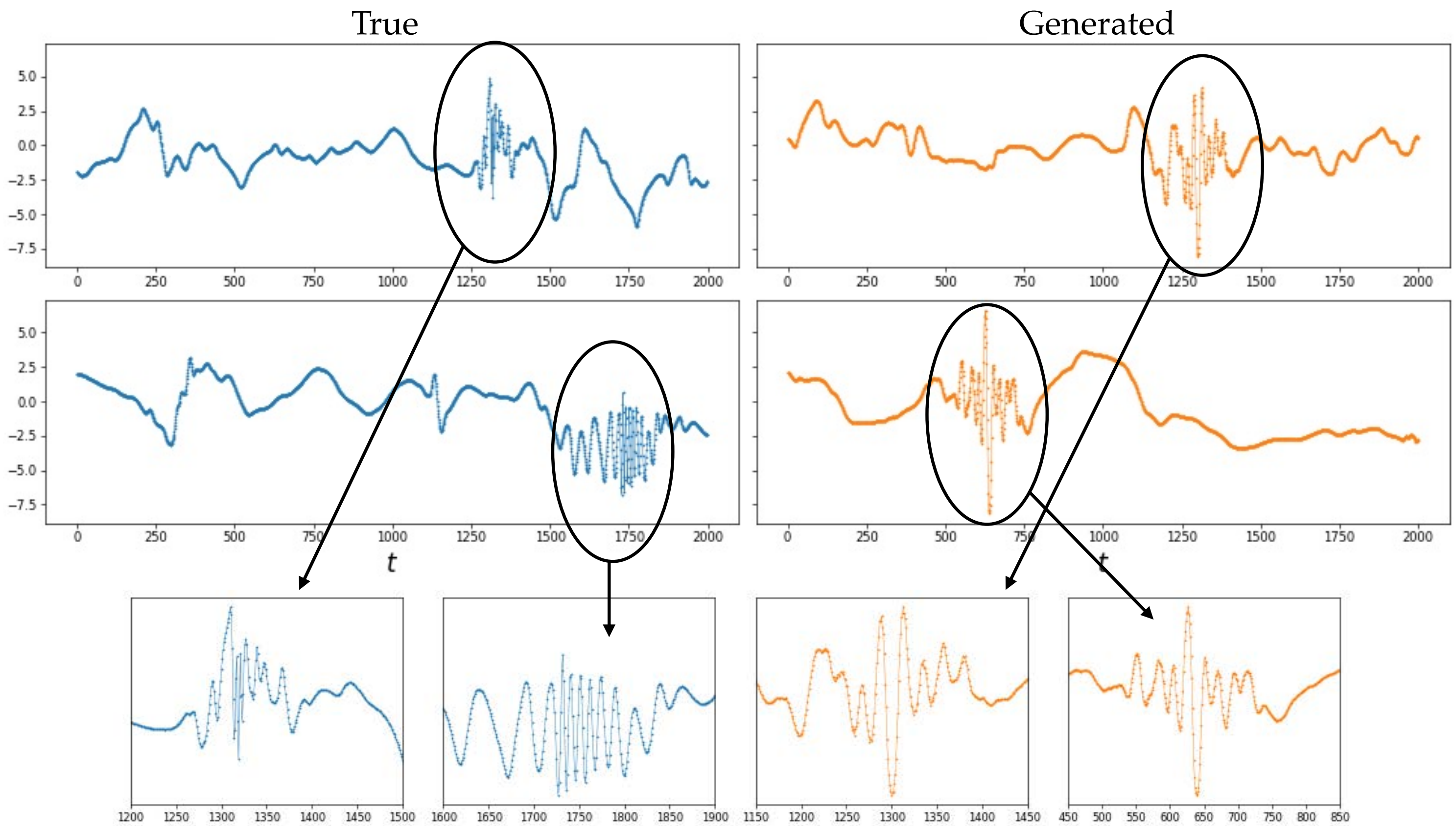
backward process



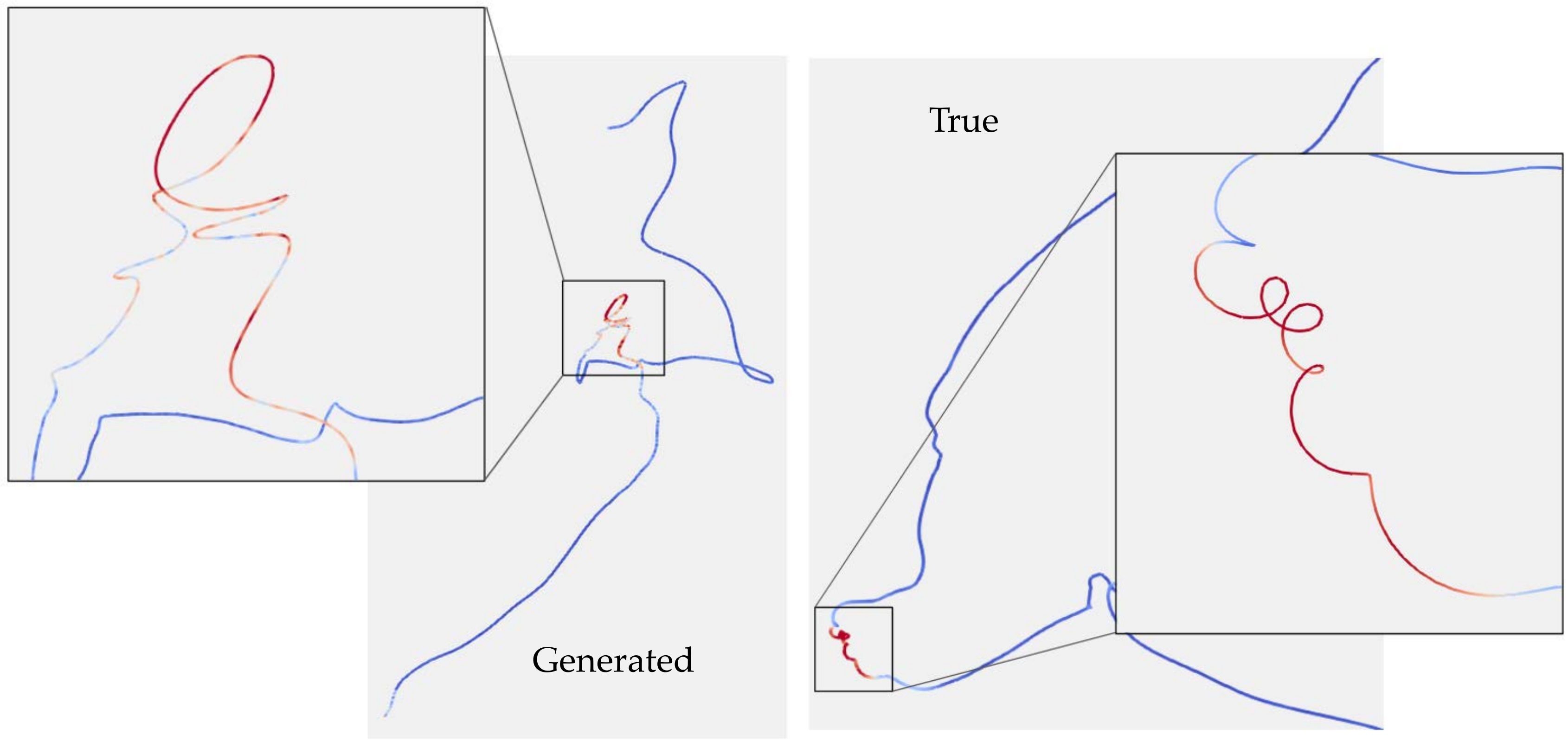
$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I})$$

forward process

Sohl-Dickstein et al., Deep Unsupervised Learning using Nonequilibrium Thermodynamics, ICML 2015
Ho et al., Denoising Diffusion Probabilistic Models, NeurIPS 2020
Song et al., Score-Based Generative Modeling through Stochastic Differential Equations, ICLR 2021



True
Generated
t
t

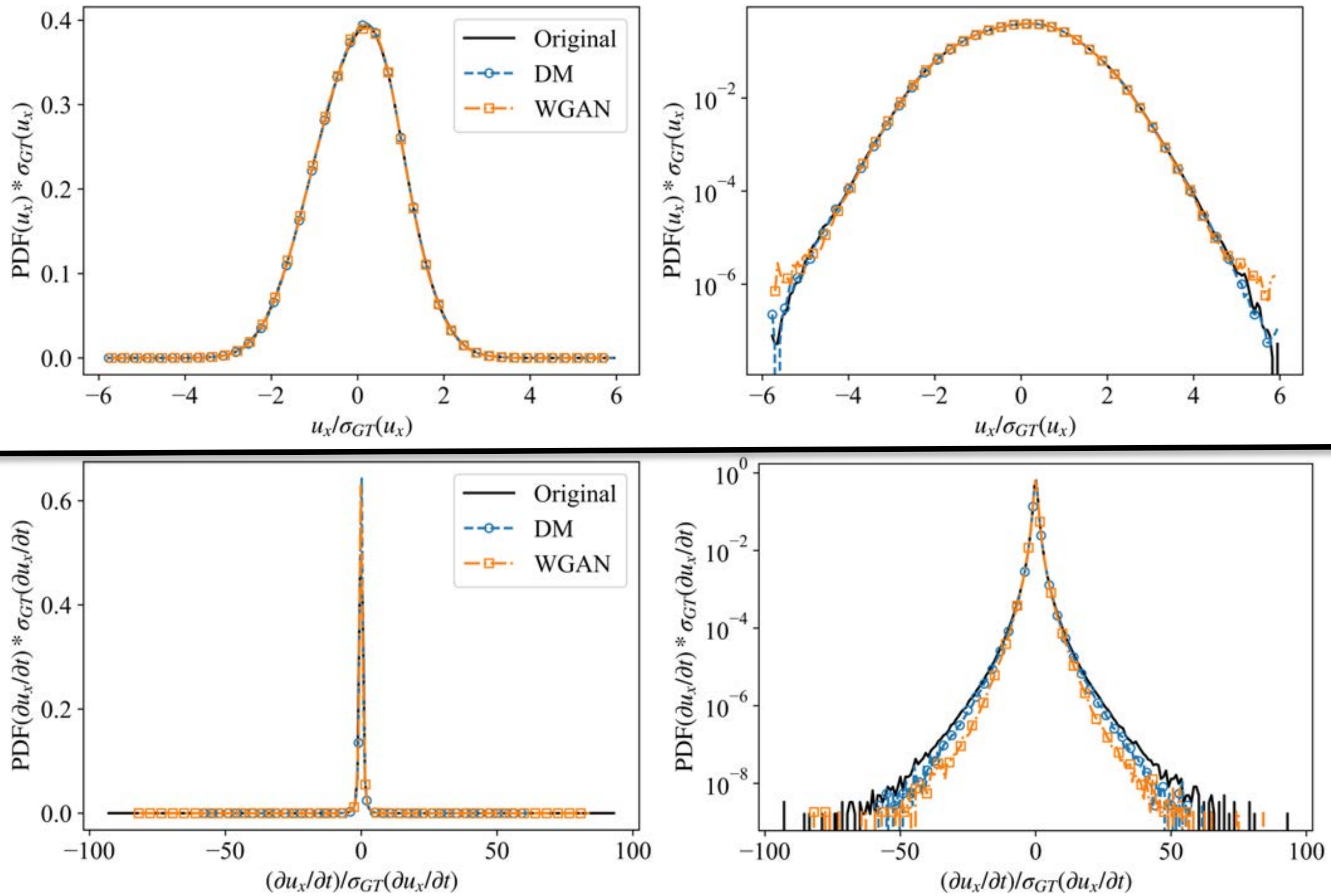

True
Generated

Statistics obtained from training on 327680 trajectories of length 2048 time steps (roughly 1 T_L)

PDFs are normalized with the same standard deviation of the ground truth, σ_{GT}

LIN-LIN PDF

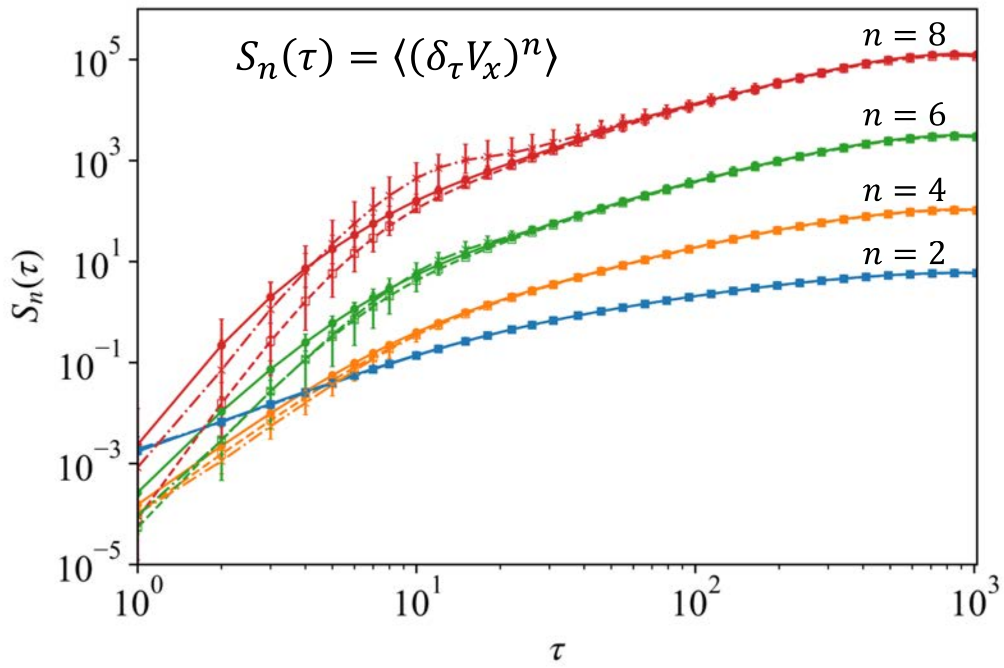
LOG-LIN PDF



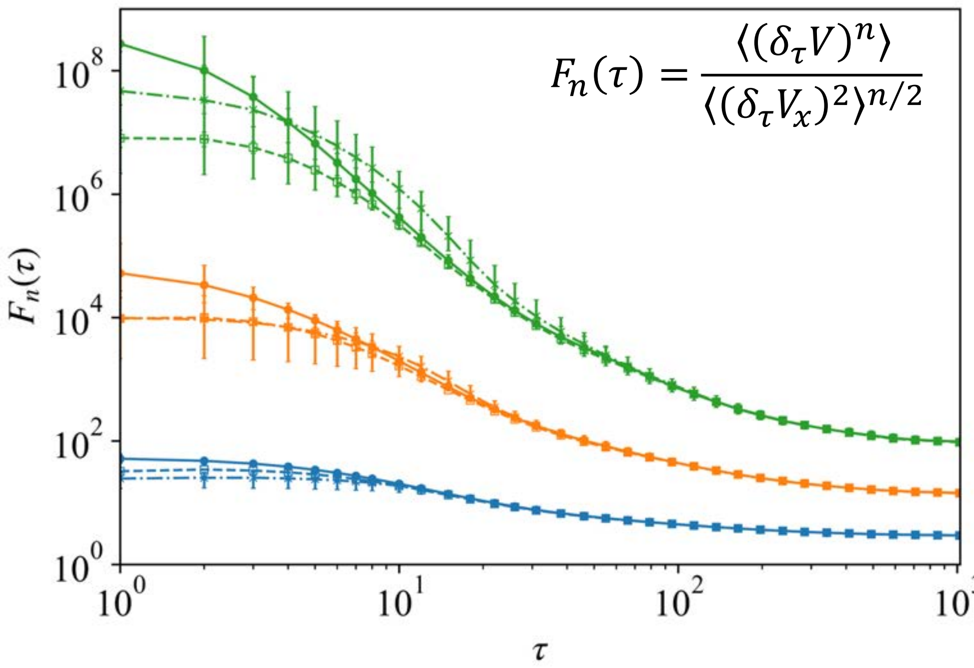
VELOCITY PDF

ACCELERATION PDF

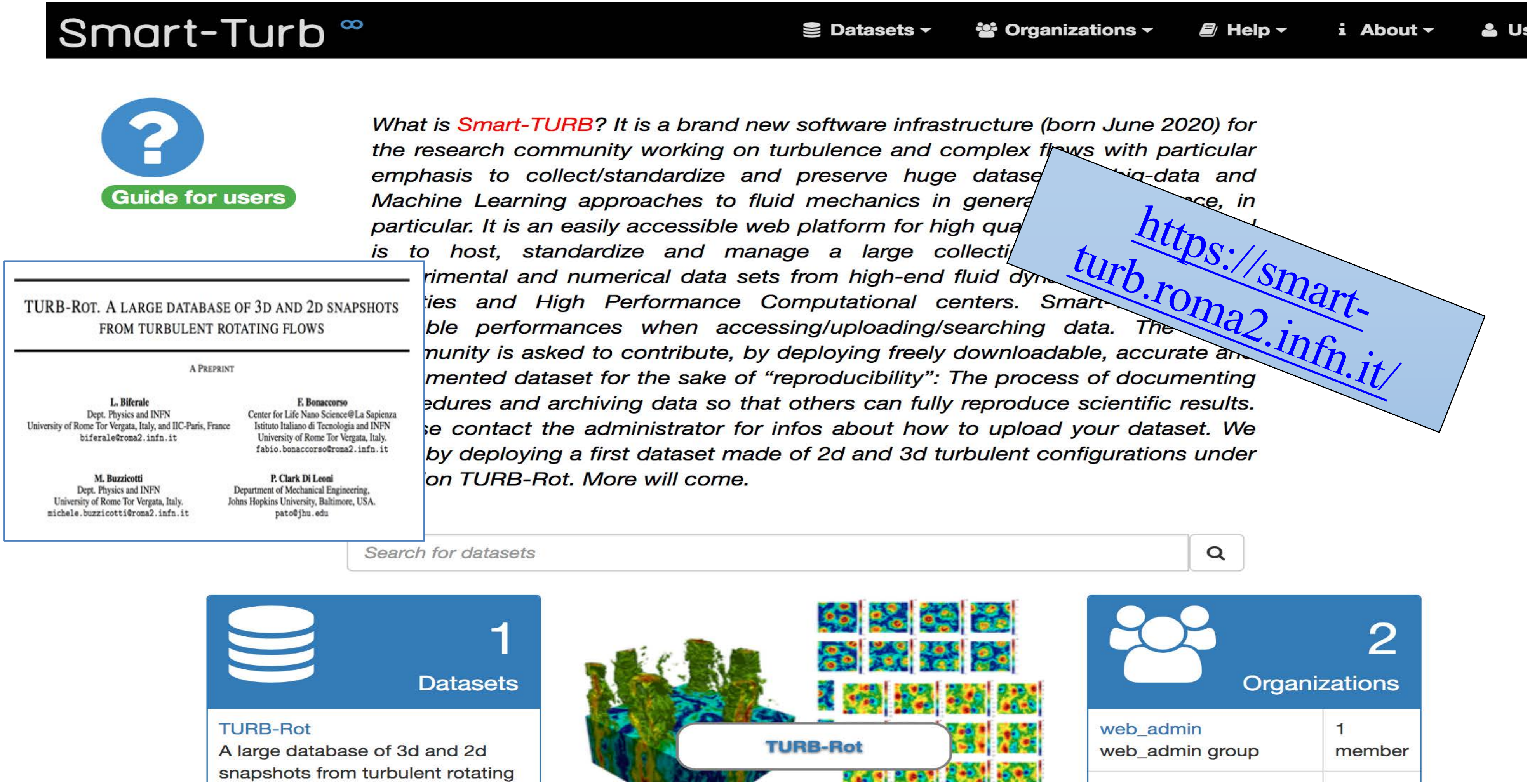
LAGRANGIAN STRUCTURE FUNCTIONS



GENERALIZED FLATNESS



Original: solid circle; DM: dashed square; WGAN: dash-dotted x


Smart-Turb
Datasets
Organizations
Help
About
Guide for users
What is Smart-TURB? It is a brand new software infrastructure (born June 2020) for the research community working on turbulence and complex flows with particular
emphasis to collect/standardize and preserve huge datase
Machine Learning approaches to fluid mechanics in genera
particular. It is an easily accessible web platform for high qua
is to host, standardize and manage a large collecti
rimental and numerical data sets from high-end fluid dy
ties and High Performance Computational centers. Smart-
ble performances when accessing/uploading/searching data. The
munity is asked to contribute, by deploying freely downloadable, accurate an
mented dataset for the sake of “reproducibility”: The process of documenting
edures and archiving data so that others can fully reproduce scientific results.
se contact the administrator for infos about how to upload your dataset. We
by deploying a first dataset made of 2d and 3d turbulent configurations under
on TURB-Rot. More will come.
https://smart-turb.roma2.infn.it/
TURB-Rot. A large database of 3D and 2D snapshots from turbulent rotating flows
A Preprint
L. Biferale
Dept. Physics and INFN
University of Rome Tor Vergata, Italy, and IIC-Paris, France
biferale@roma2.infn.it
F. Bonaccorso
Center for Life Nano Science@La Sapienza
Istituto Italiano di Tecnologia and INFN
University of Rome Tor Vergata, Italy.
fabio.bonaccorso@roma2.infn.it
M. Buzzicotti
Dept. Physics and INFN
University of Rome Tor Vergata, Italy.
michele.buzzicotti@roma2.infn.it
P. Clark Di Leoni
Department of Mechanical Engineering,
Johns Hopkins University, Baltimore, USA.
pato@jhu.edu
Search for datasets
1
Datasets
TURB-Rot
A large database of 3d and 2d snapshots from turbulent rotating
TURB-Rot
2
Organizations
web_admin
web_admin group
1 member

ARE WE CLOSE TO AI SUPREMACY IN FLUID DYNAMICS? NO

- RATE OF PUBLICATIONS IN THE DOMAIN >> RATE OF READING/PEER REVIEWING -> DANGER OF INFLATIONARY ERA
- HUNDREDS OF PAPERS IN THE ARXIVES CITED BY HUNDREDS OF OTHER PAPERS WITHOUT CHECK ON THE RESULTS, **NOT EVEN WRONG!**

- **NEED FOR QUANTITATIVE AI:** SCALING, VALIDATION, BENCHMARKS, GENERALISATION, GRAND-CHALLENGES TO ESTABLISH BEST-PRACTISE
- NEED FOR PHYSICAL DIMENSIONALISATION: NETWORK STRUCTURE VS PHYSICAL PARAMETERS (deepness, structure, coding, # training data vs Reynolds, Rayleigh, Time-to-solution, Mach, Mass fraction etc…)
- **NEED FOR INTERDISCIPLINARY COLLABORATIONS:** APPLIED SCIENTISTS (FOR THE GOOD QUESTIONS) + AI SPECIALISTS (TO OPEN THE BOX) + NUMERICAL SCIENTISTS (FOR THE GOOD DATA)

2021-2026 AdG GRANT Smart-TURB

A Physics-Informed Machine-Learning Platform Smart Lagrangian Harness and Control of TURBulence

biferale@roma2.infn.it biferale@gmail.com

WE ARE HIRING POST-DOCS

European Research Council

Established by the European Commission

WE ARE HIRING POST-DOCS

References

Li, Buzzicotti, Bonaccorso, Biferale, Wan and Chen. Data reconstruction of turbulent flows with Gappy-POD and Generative Adversarial Networks. In preparation 2022.

Buzzicotti, Bonaccorso, and Biferale. "Inferring Turbulent Parameters via Machine Learning." ***arXiv preprint arXiv:2201.00732*** **(2022)**.

Buzzicotti, Bonaccorso, Clark Di Leoni & Biferale. **Phys. Rev. Fluids 6, 050503 (2021).** Reconstruction of turbulent data with deep generative models for semantic inpainting from TURB-Rot database.

Biferale, Bonaccorso, Buzzicotti, & Clark di Leoni. **arXiv:2006.07469 (2020). T**URB-Rot. A large database of 3d and 2d snapshots from turbulent rotating flows.

Beintema, Corbetta, Biferale & Toschi, Controlling Rayleigh Benard Convection via Reinforcement Learning Journal of Turbulence 21 (9-10), 585-605, 2020

Biferale, Bonaccorso, Buzzicotti, Clark Di Leoni & Gustavsson. **Chaos: An Interdisciplinary Journal of Nonlinear Science. 2019 Oct 24;29(10):103138.** Zermelo's problem: Optimal point-to-point navigation in 2D turbulent flows using reinforcement learning.

Buzzicotti, Biferale, Bonaccorso, Clark Di Leoni & Gustavsson. **Springer, Cham, (2021)**. Optimal Control of Point-to-Point Navigation in Turbulent Time Dependent Flows Using Reinforcement Learning.

Clark Di Leoni, Mazzino, Biferale. **Phys. Rev. Fluids 3, 104604, (2018)**. Inferring flow parameters and turbulent configuration with physics-informed data-assimilation and spectral nudging.

Clark Di Leoni, Mazzino, Biferale. **Phys. Rev. X 10.1, 011023, (2020)**. Synchronization to big-data: nudging the Navier-Stokes equations for data assimilation of turbulent flows.

Colabrese, Gustavsson, Celani, Biferale. **Phys. Rev. Fluids 3**, **084301, (2018)**. Smart Inertial Particles.

Colabrese, Gustavsson, Celani, Biferale. **Phys. Rev. Letters 118 (15), 158004, (2017)**. Flow navigation by smart microswimmers via reinforcement learning.