AI and Scientific Computing

There is Plenty of Room in the Middle

ΠΕΤΡΟΣ ΚΟΥΜΟΥΤΣΑΚΟΣ

HARVARD
School of Engineering and Applied Sciences

WITH

Lucas Amoudruz

George Arampatzis

Han Gao

Sebastian Kaltenbach


-2500 YEARS
RAFAEL

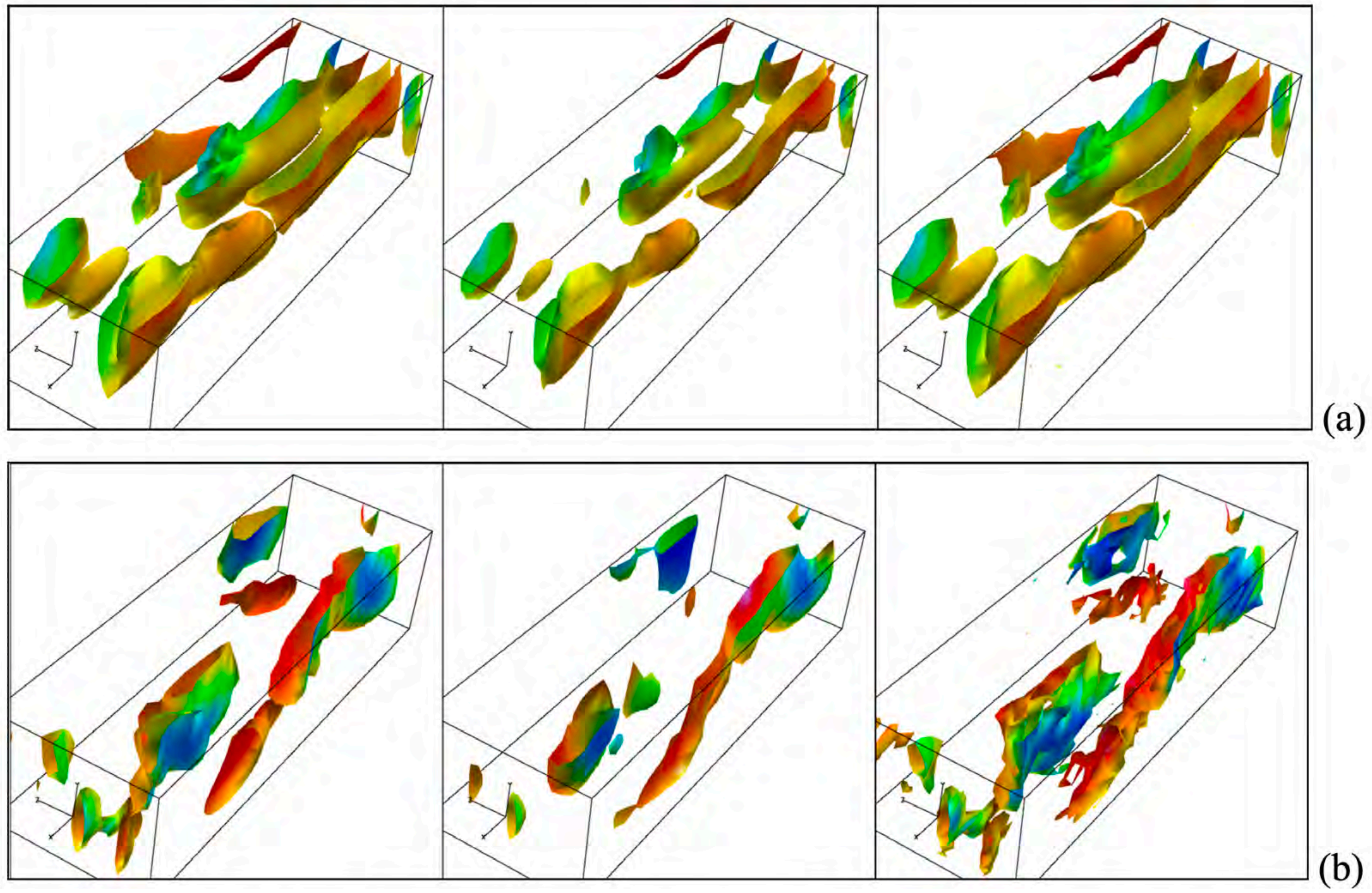
Neural Network Modeling for Near Wall Turbulent Flow

Michele Milano[1] and Petros Koumoutsakos[2]

Institute of Computational Sciences, ETH Zentrum, CH-8092 Zürich, Switzerland
E-mail: milano@inf.ethz.ch, petros@inf.ethz.ch

Received May 23, 2001; revised January 8, 2002

ORIGINAL POD NN



First ever Deep NNs for Science (?)

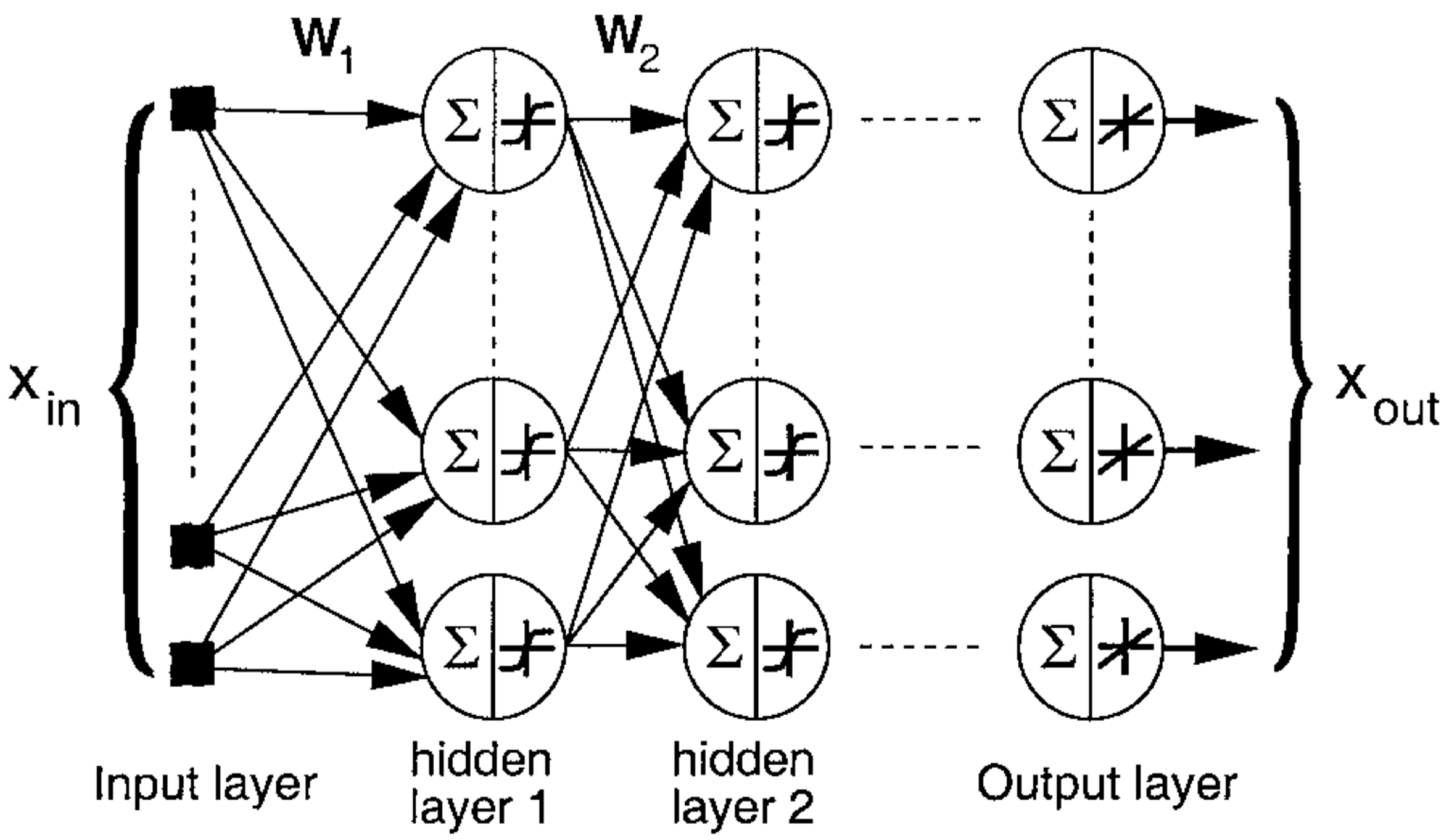
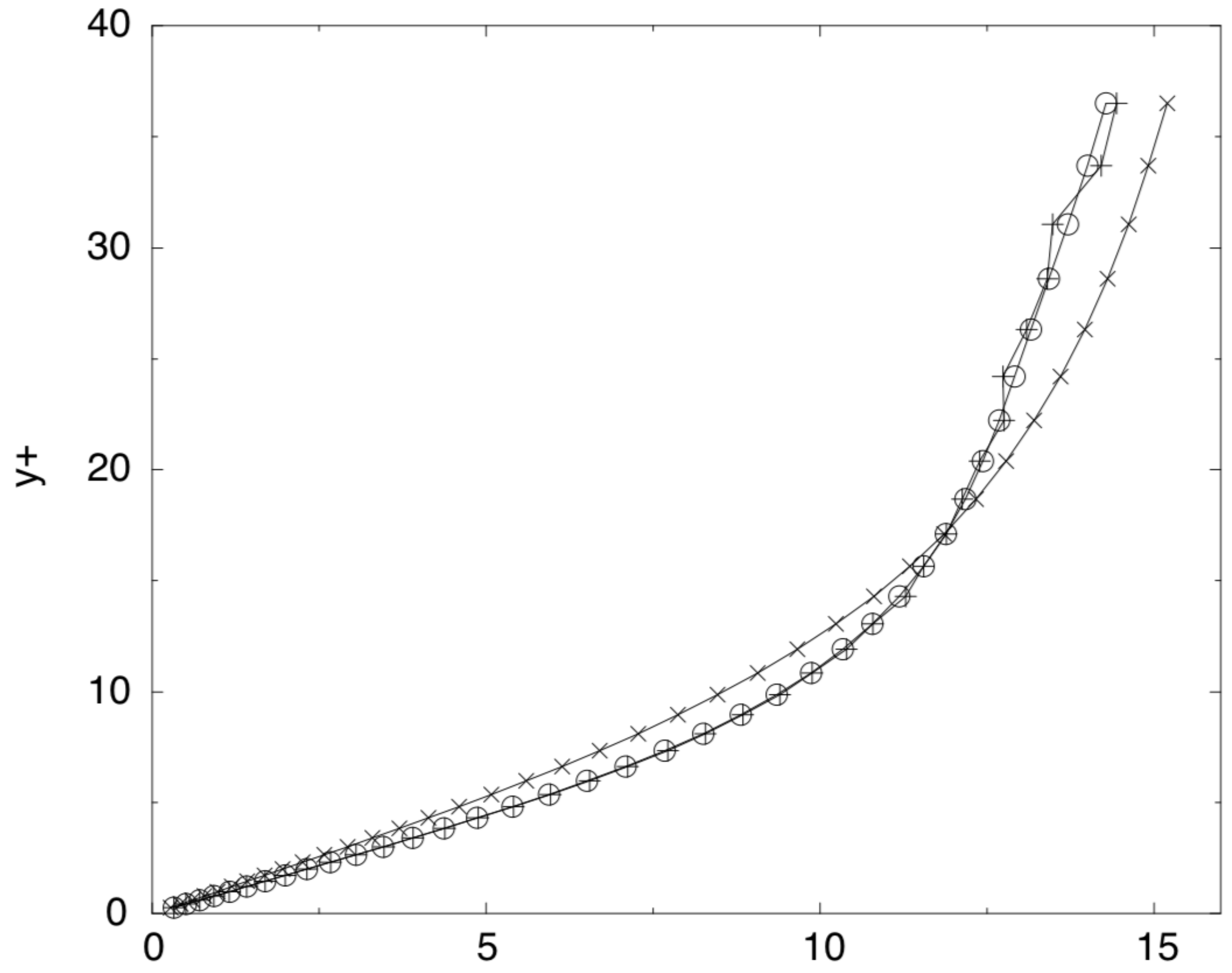


FIG. 1. Layer representation of a nonlinear neural network structure.



$$u(\mathbf{x},t) = \omega_z^w y + \frac{Re}{2}\frac{\partial P^w}{\partial x} y^2 + \mathbf{M}(P^w, S^w)$$

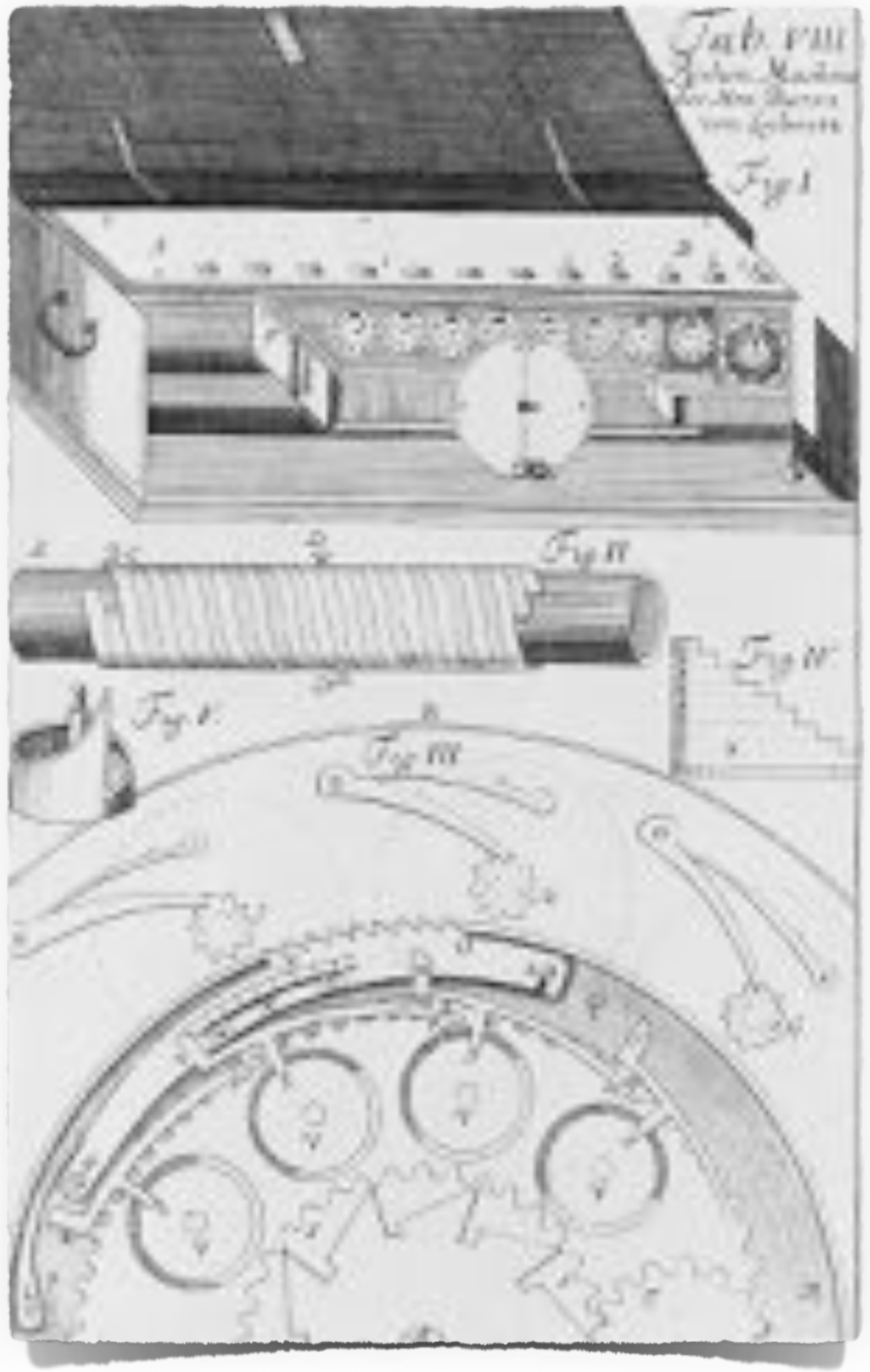
THE PURSUIT OF TRUTH

PLATO: *The Allegory of the Cave*

CREDIT: TED Ed

If **controversies** were to arise,
there would be no more need of disputation between
two philosophers than between **two accountants.**
For it would suffice to take their pencils in their hands,
and say to each other:

Calculemus—Let us calculate.



Euclid Descartes Russel Llull Hilbert Boole **Leibniz** Frege Newton Laplace Wittgenstein Turing Shannon

COMPUTING: The beginning…

1961

COMPUTERS

1981

1986

1988

1990

1992

1992

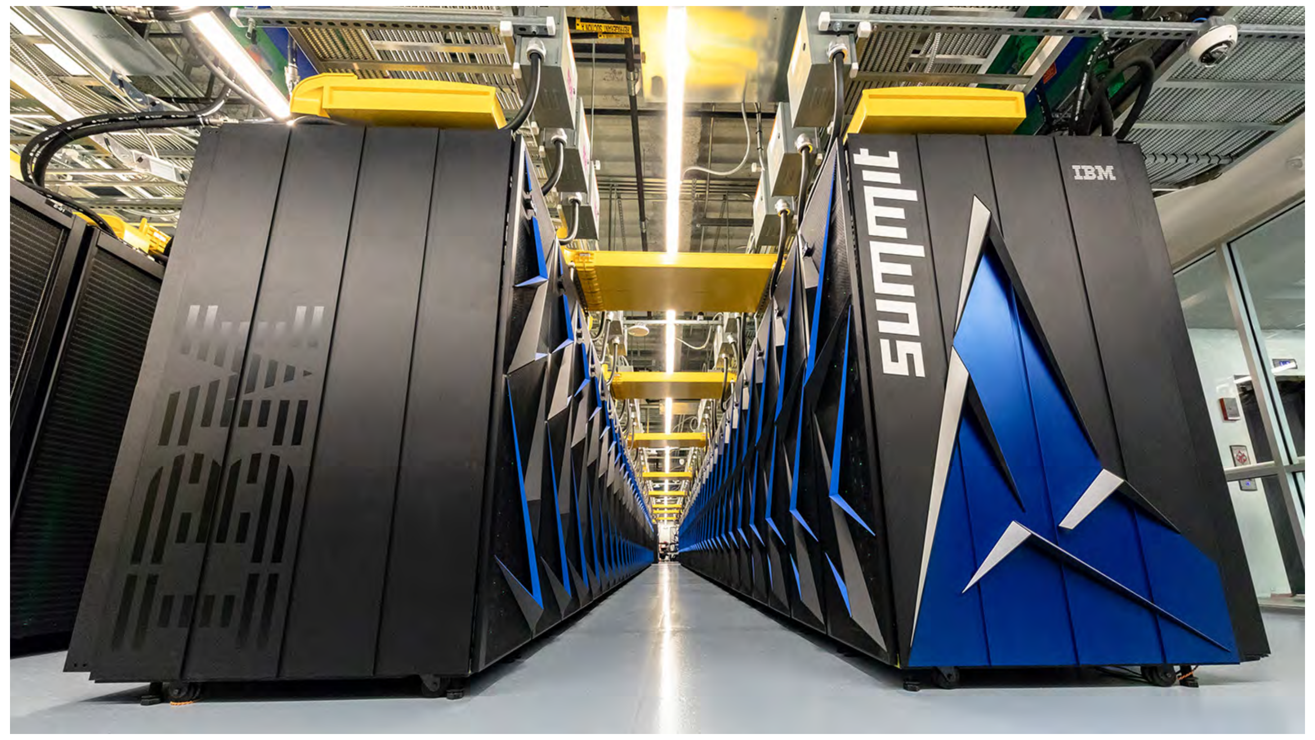
1994

1996

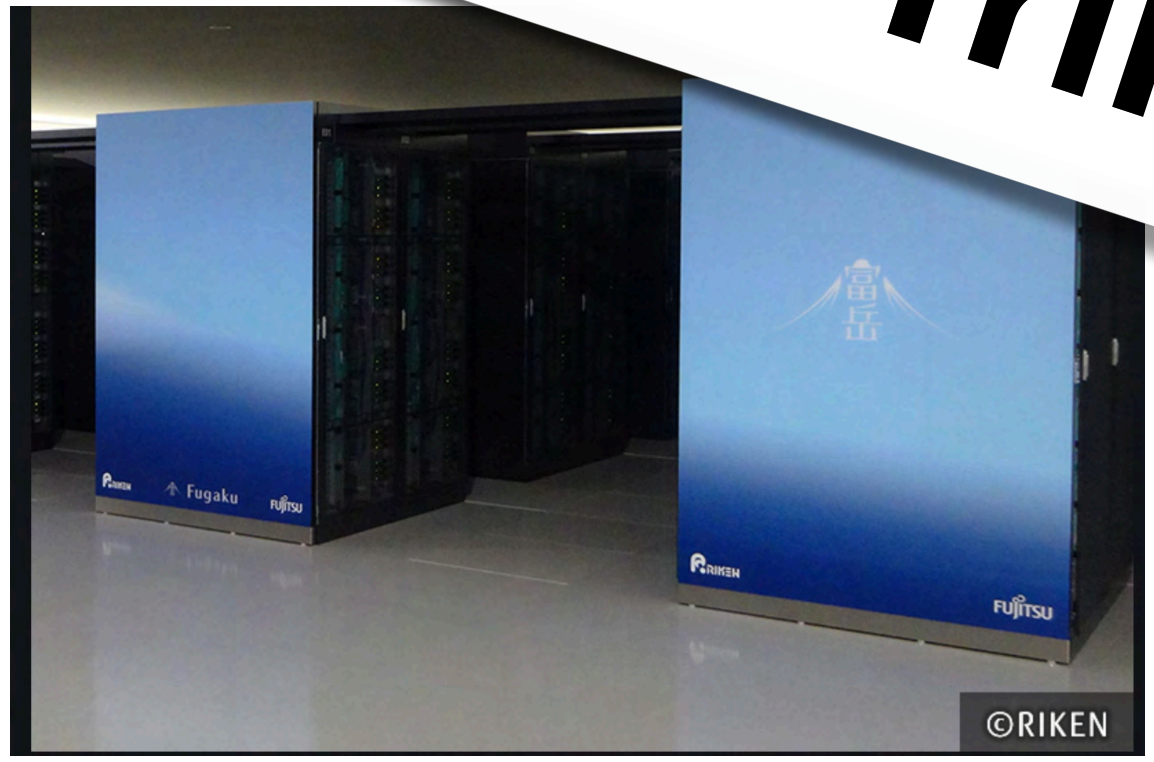
1997

2013

2019



2020



~1 Trillion X

SOURCES: http://www.computerhistory.org/timeline/computers/

COMPUTERS : A Disruptive Technology

Deep Blue beat Kasparov

Posted by: Marco van der Spek   Date: Oct 2, 2012

Category: Articles

http://www.testnewsonline.com/2012/10/02/deep-blue-beat-kasparov-because-of-bug/

MindGoogle) winning Go against Lee Sedol, one of the world's top go players.March 11, 2016

SCIENCE FILE - Los Angeles Times
9 March 2017

No need for a poker face - Software program DeepStack beats the pros at Texas Hold 'Em

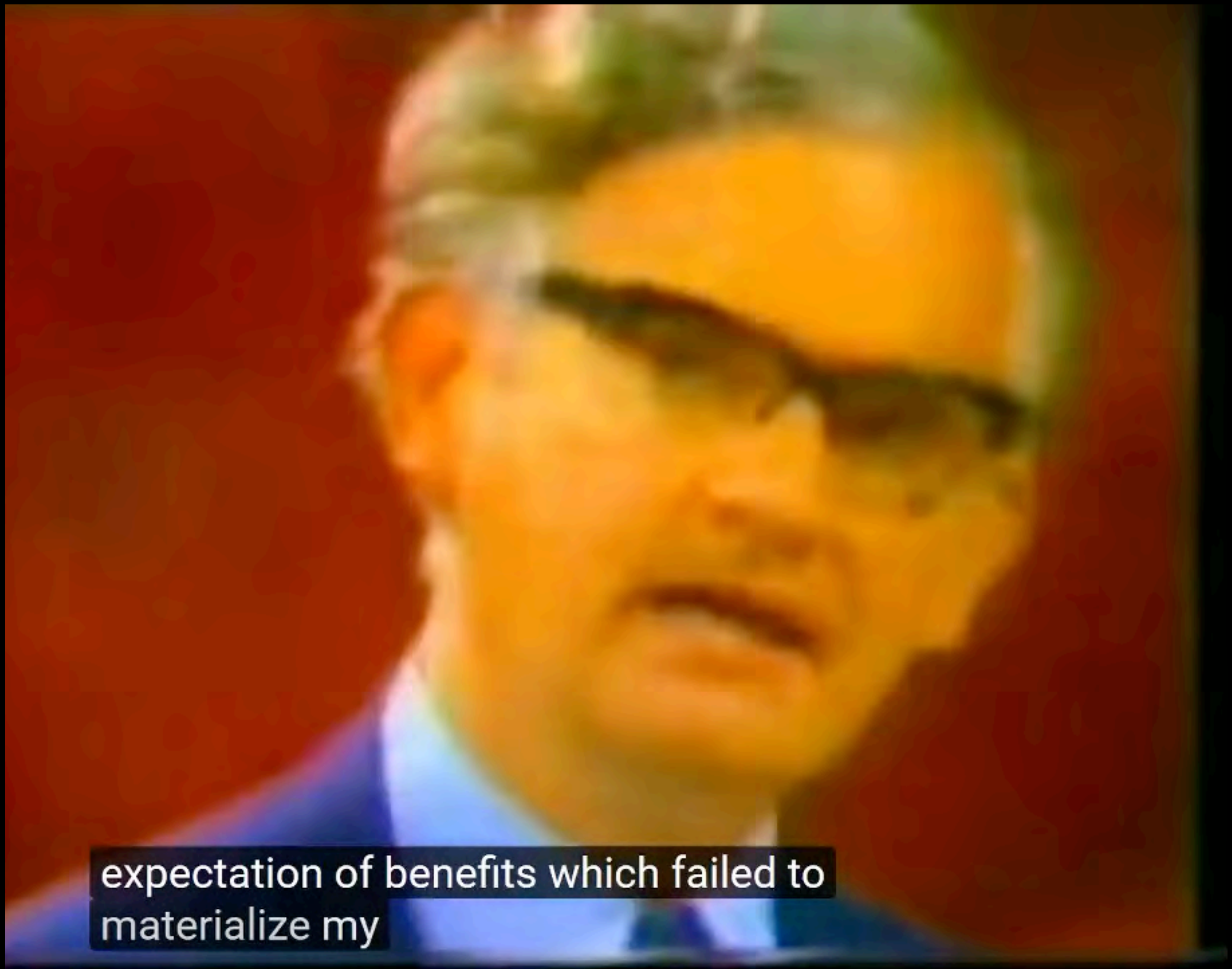
ARTICLE

doi:10.1038/nature16961

Mastering the game of Go with deep neural networks and tree search

David Silver[1]*, Aja Huang[1]*, Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

AI and Fluid Mechanics - The Lighthill Report (1973)

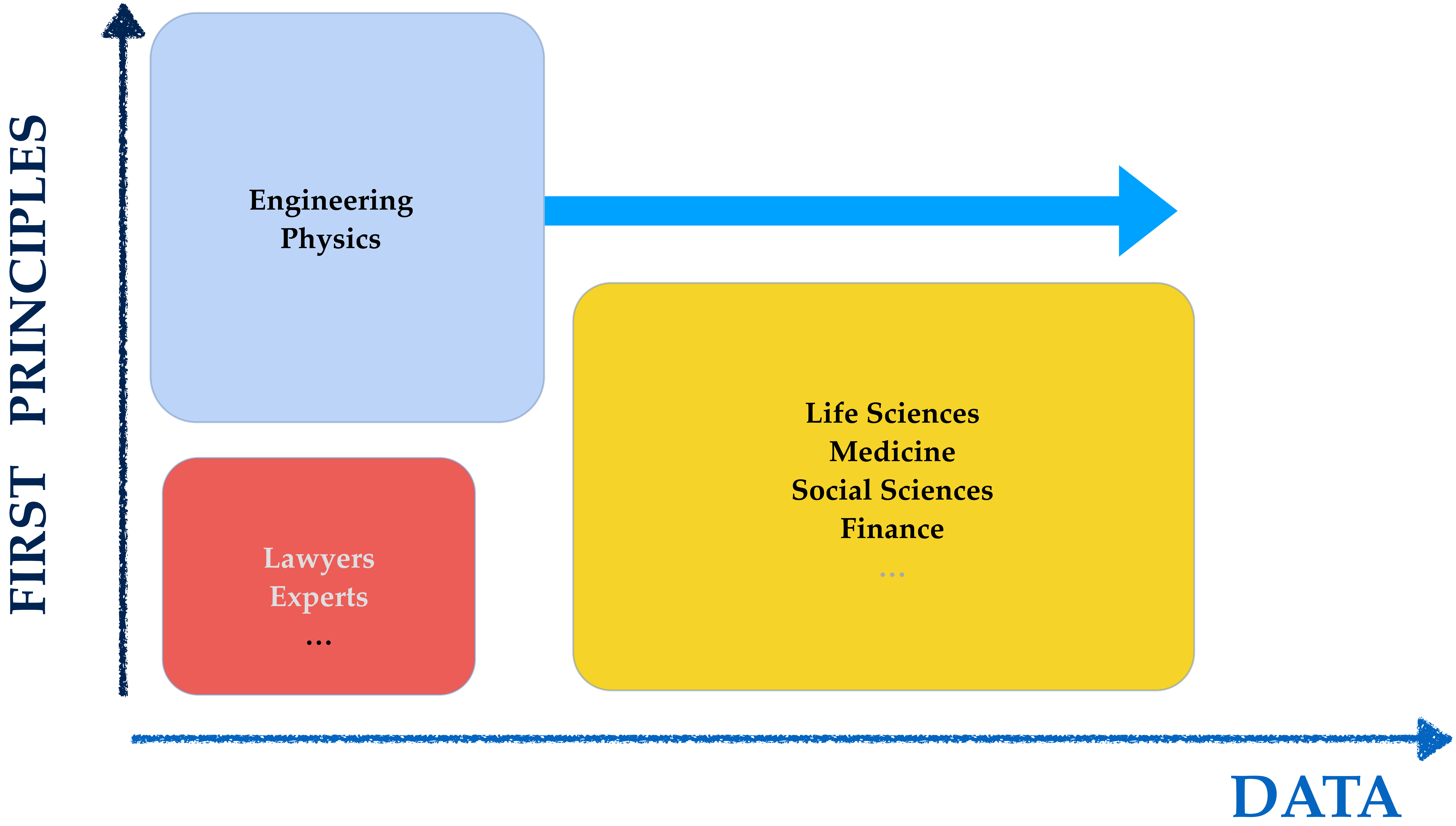


Lighthill's position does not come as a surprise. **He was, after all, a researcher in fluid dynamics and aeroacoustics, where it is easy to be misled by complicated differential equations involving 'continuous' variables and where nonexistent solutions arise so often.**

http://www.mathrix.org

Lighthill's main argument was that because one had to specify the rules in a computer to tell the robot how to behave in every possible scenario, every attempt to come up with a general purpose robot would quickly turn out to be an intractable problem, with a combinatorial explosion of possible solutions.

SOLVING PROBLEMS

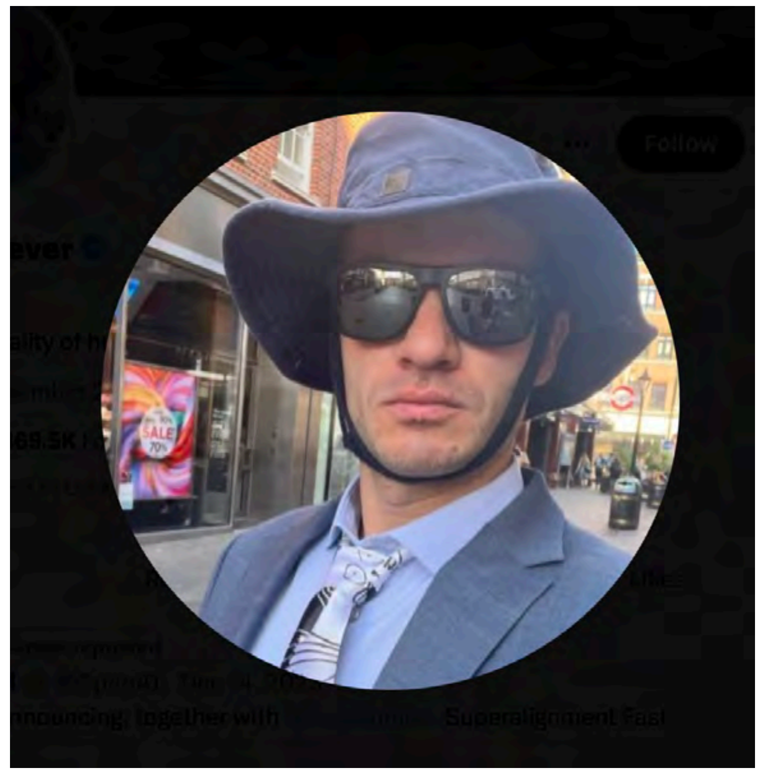


How to solve hard problems?

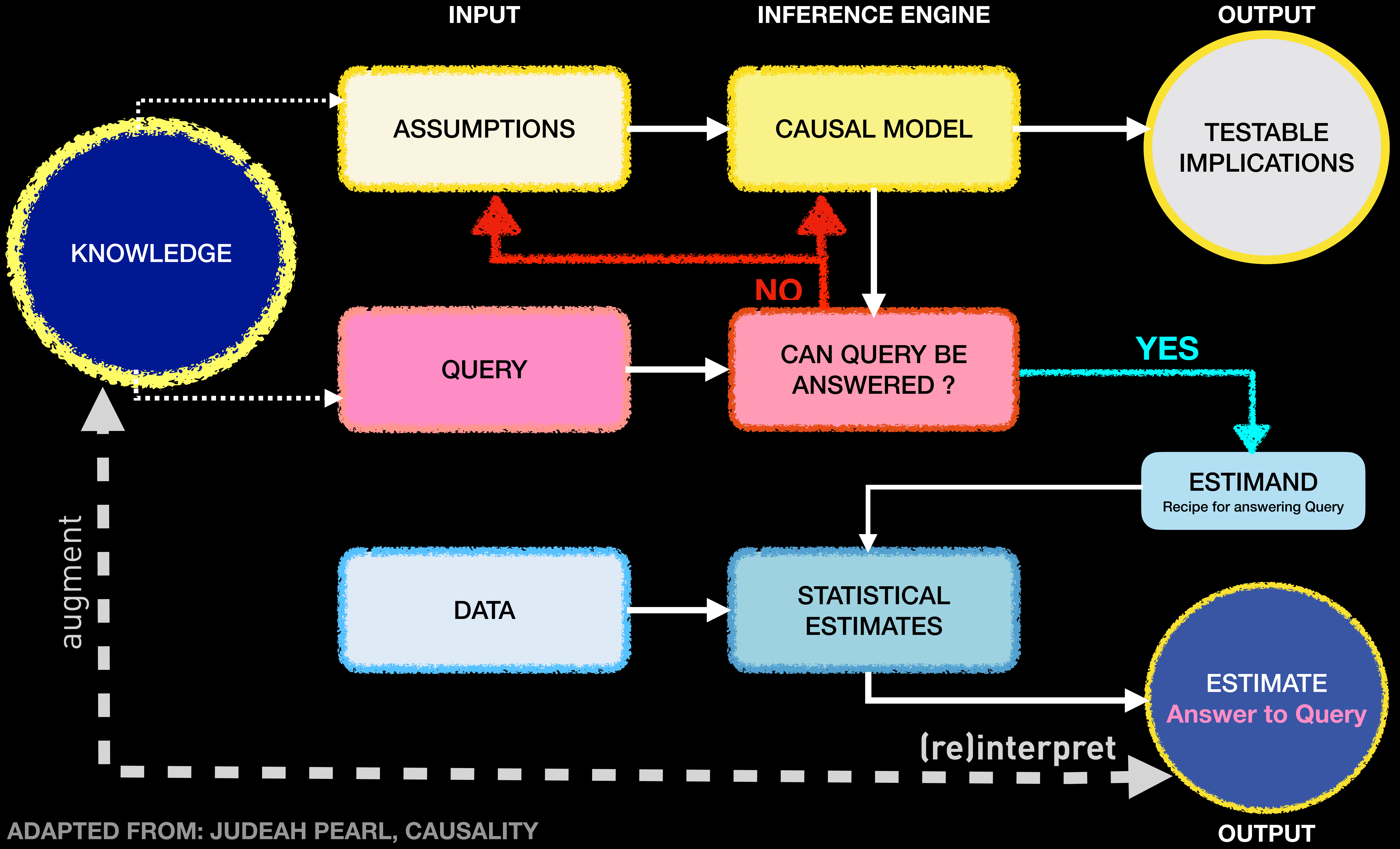
Use lots of training data.

And a big deep neural network.

And success is the only possible outcome.

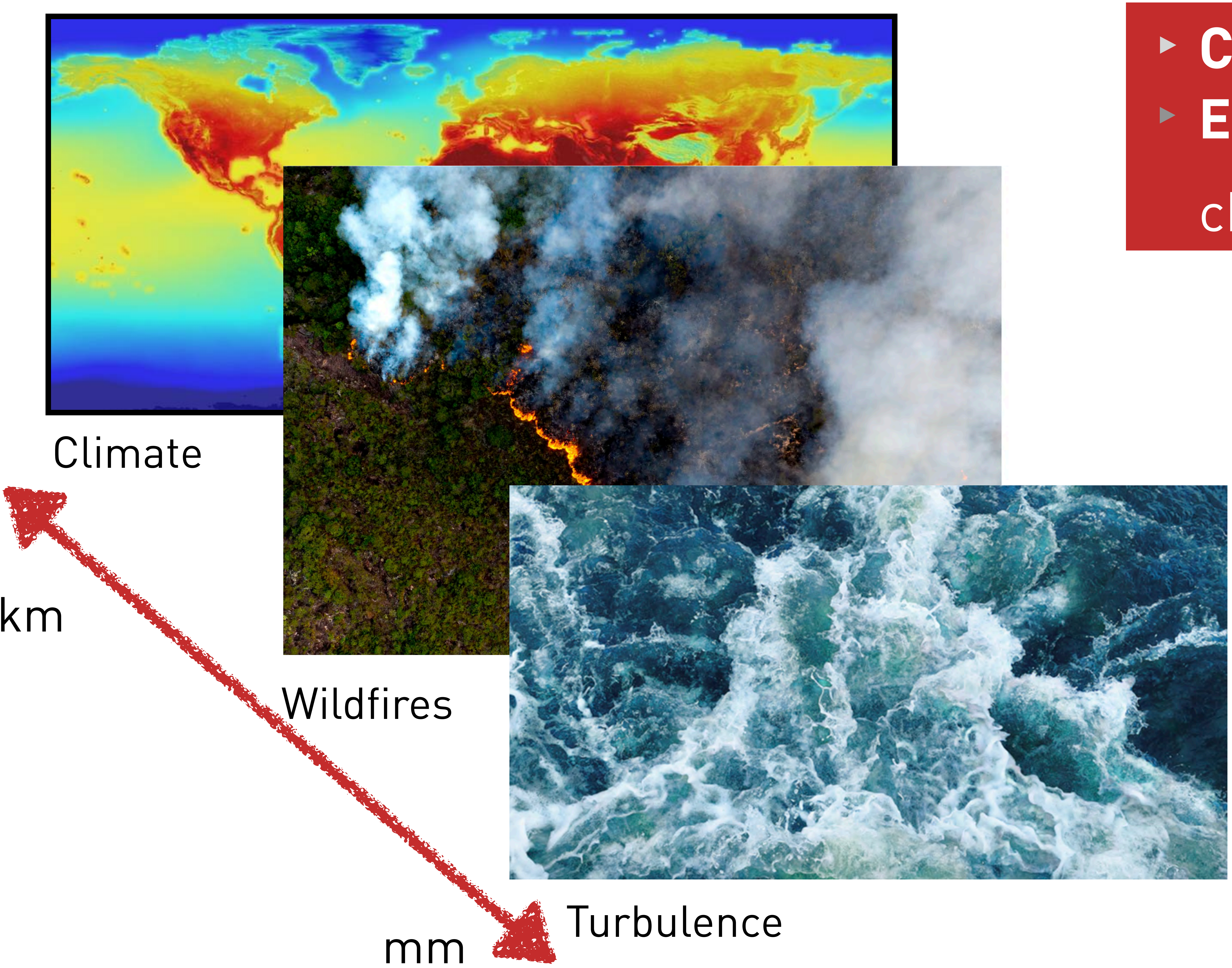


Ilya Sutskever (2015),
co-founder of OpenAI



ADAPTED FROM: JUDEAH PEARL, CAUSALITY

Forecasting Complex Systems



- **Chaotic** dynamics
- **Expensive** to simulate and/or challenging to **forecast**

How to design fast, methods that capture/predict system dynamics?

➡**Existing methods**:

Surrogate models , ROMs

LES,RANS,DMD, ...

SCIENCE ADVANCES | RESEARCH ARTICLE

PHYSICS

Computing foaming flows across scales: From breaking waves to microfluidics

Petr Karnakov[1,2], Sergey Litvinov[1], Petros Koumoutsakos[1,2]*

SCIENTIFIC COMPUTING

Expensive Models based on First Principles

MACHINE LEARNING

Capabilities for Pattern Recognition

Example: Dimensionality Reduction -> PCA as NN

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

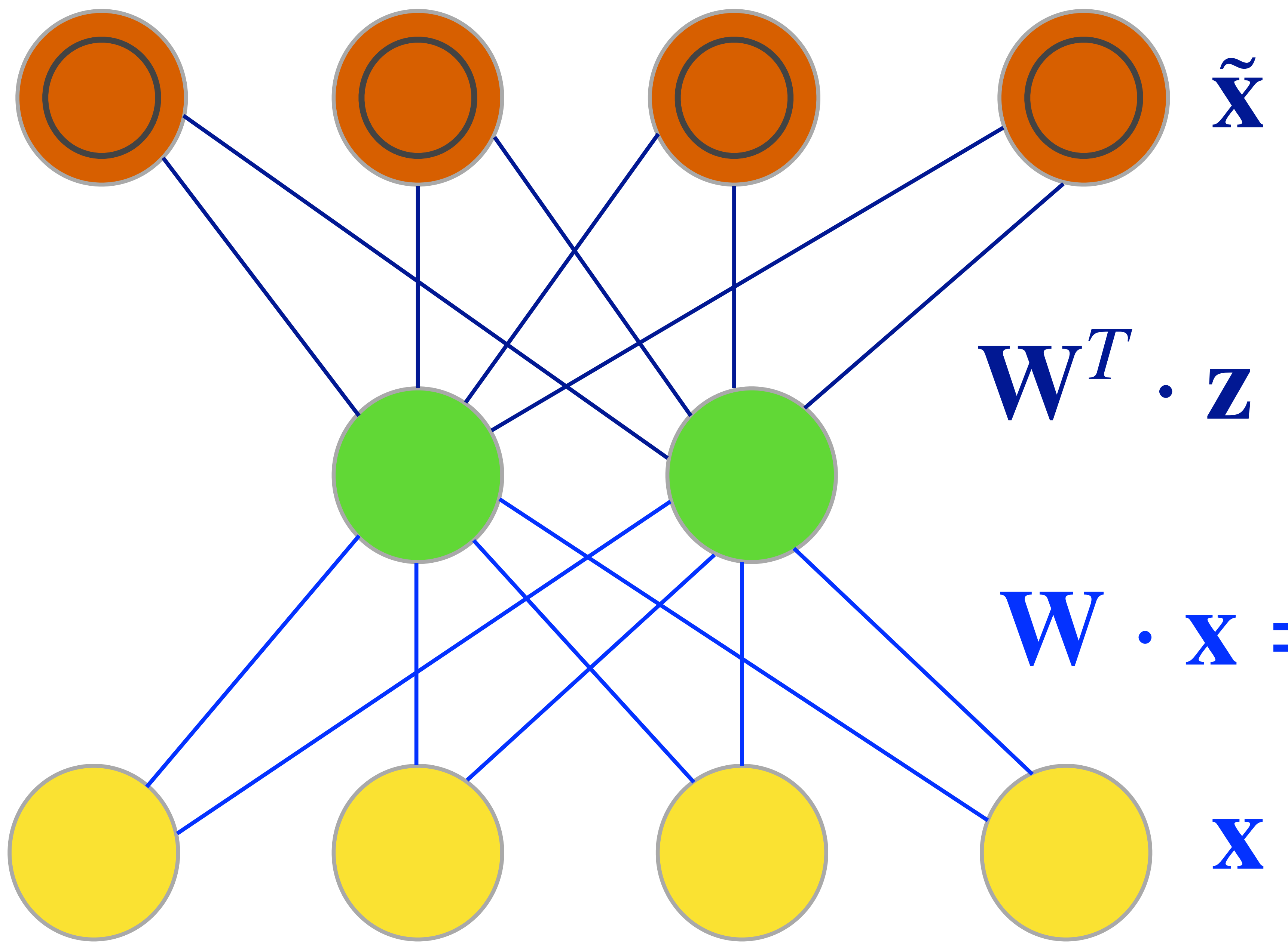
$$\mathrm{S} = \frac{1}{N}\sum_{n=1}^{N}(\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$$

$$\mathbf{S}\mathbf{u}_i = \lambda_i \mathbf{u}_i$$

retain M < D eigenvectors

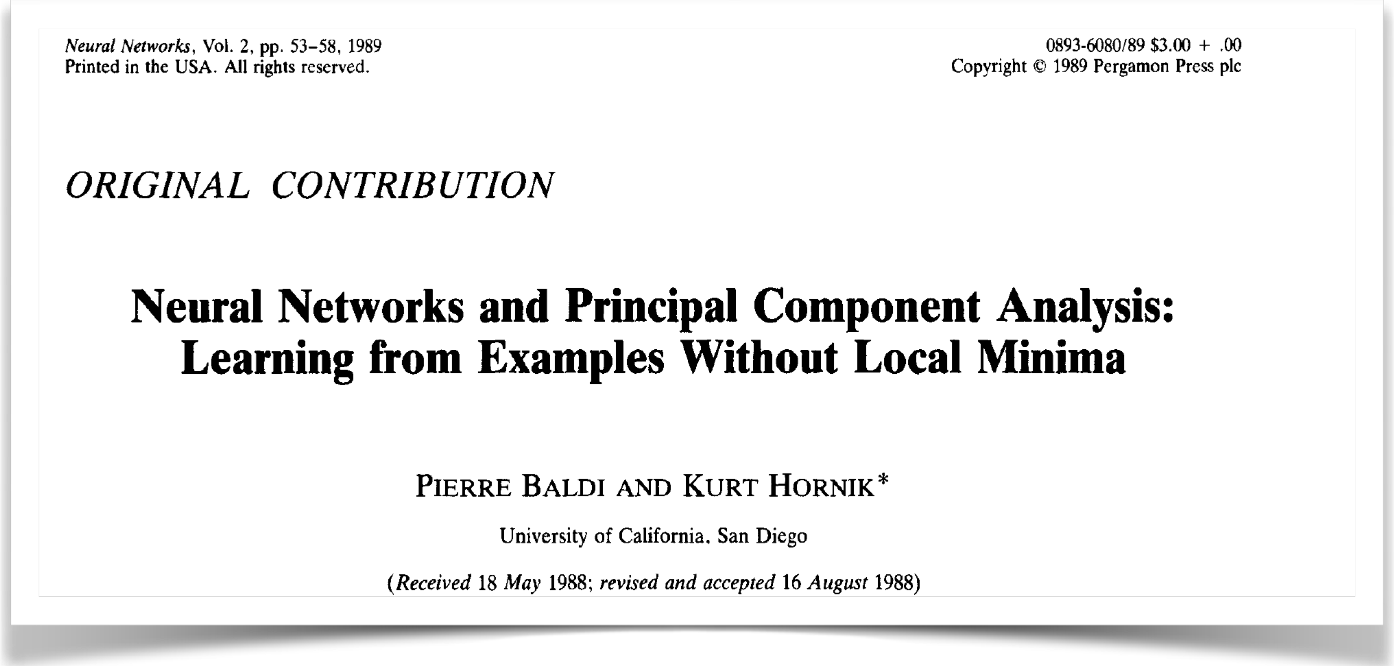
$$E = ||\tilde{\mathbf{x}} - \mathbf{x}||^2$$

$$= ||\mathbf{W}^{\mathrm{T}}\mathbf{W} \cdot \mathbf{x} - \mathbf{x}||^2$$



$$\mathbf{W}^T \cdot \mathbf{z} = \tilde{\mathbf{x}}$$

$$\mathbf{W} \cdot \mathbf{x} = \mathbf{z}$$



Neural Networks, Vol. 2, pp. 53–58, 1989
Printed in the USA. All rights reserved.
0893-6080/89 $3.00 + .00

ORIGINAL CONTRIBUTION

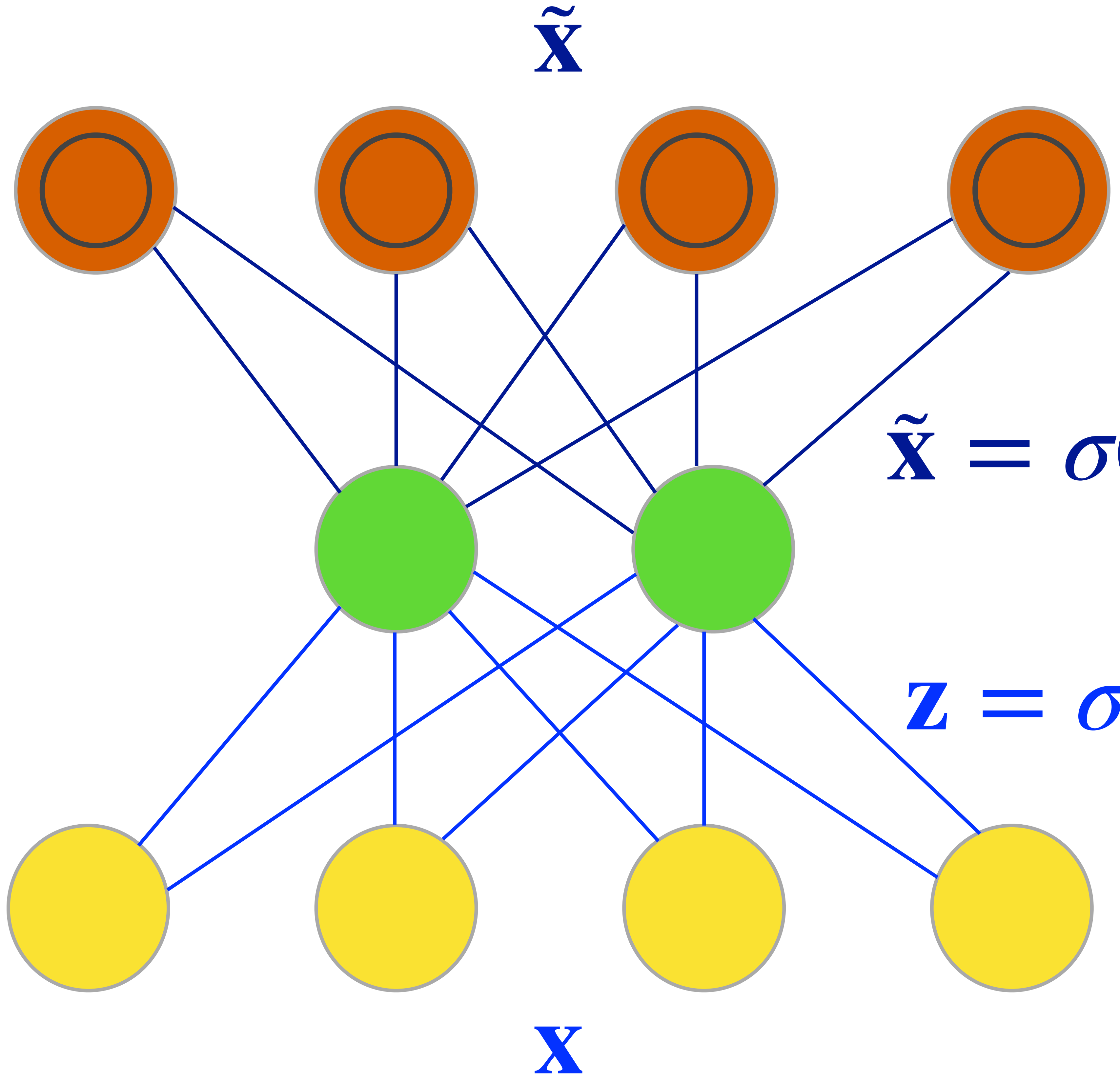
Neural Networks and Principal Component Analysis: Learning from Examples Without Local Minima

PIERRE BALDI AND KURT HORNIK*
University of California, San Diego

(Received 18 May 1988; revised and accepted 16 August 1988)

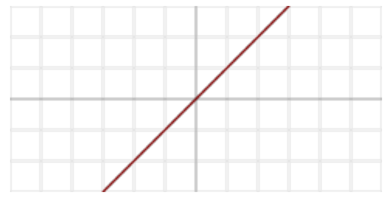
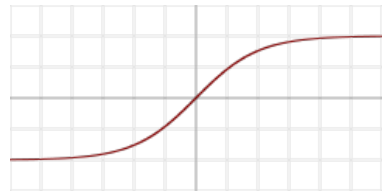
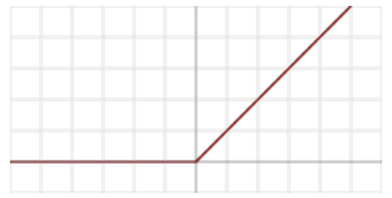
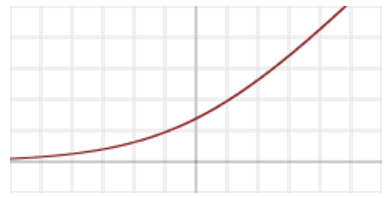
Find $\mathbf{W}$ by minimizing E

Non-Linear PCA - Autoencoders



$$\tilde{\mathbf{x}} = \sigma(\mathbf{W}_2 \cdot \mathbf{z} + b_2)$$

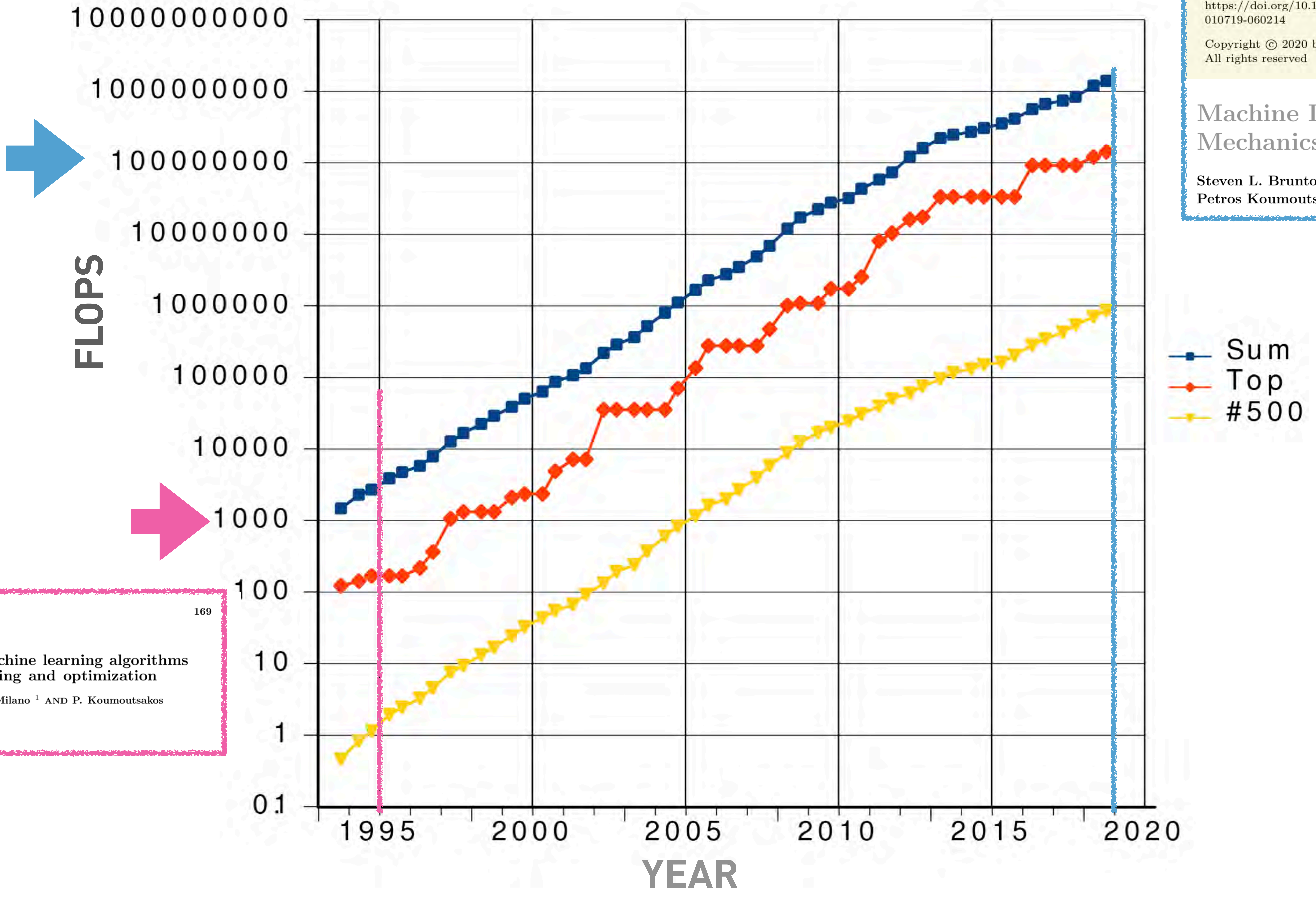
$$\mathbf{z} = \sigma(\mathbf{W}_1 \cdot \mathbf{x} + b_1)$$

activation function σ		derivative
$\sigma(x) = x$		$\sigma'(x) = 1$
$\sigma(x) = \frac{2}{1+e^{-2x}} - 1$		$\sigma'(x) = 1 - f(x))^2$
$\sigma(x) = max(0, x)$		$\sigma'(x) = max(0, 1)$
$\sigma(x) = ln(1 + e^x)$		$\sigma'(x) = \frac{1}{1+e^{-x}}$

- learns a function with target values equal to the input
- linear auto-encoder “equivalent” to PCA/POD

$$E = ||\tilde{\mathbf{x}} - \mathbf{x}||^2$$

1Y in 1995~ 2' in 2019



Center for Turbulence Research
Annual Research Briefs 1999

Application of machine learning algorithms to flow modeling and optimization

By S. Müller [1], M. Milano [1] AND P. Koumoutsakos

Annu. Rev. Fluid Mech. 2020. 52:1–31

https://doi.org/10.1146/annurev-fluid-010719-060214

Machine Learning for Fluid Mechanics

Steven L. Brunton,[1] Bernd R. Noack,[2] and Petros Koumoutsakos[3,4]

Deep learning's Big Bang moment.

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

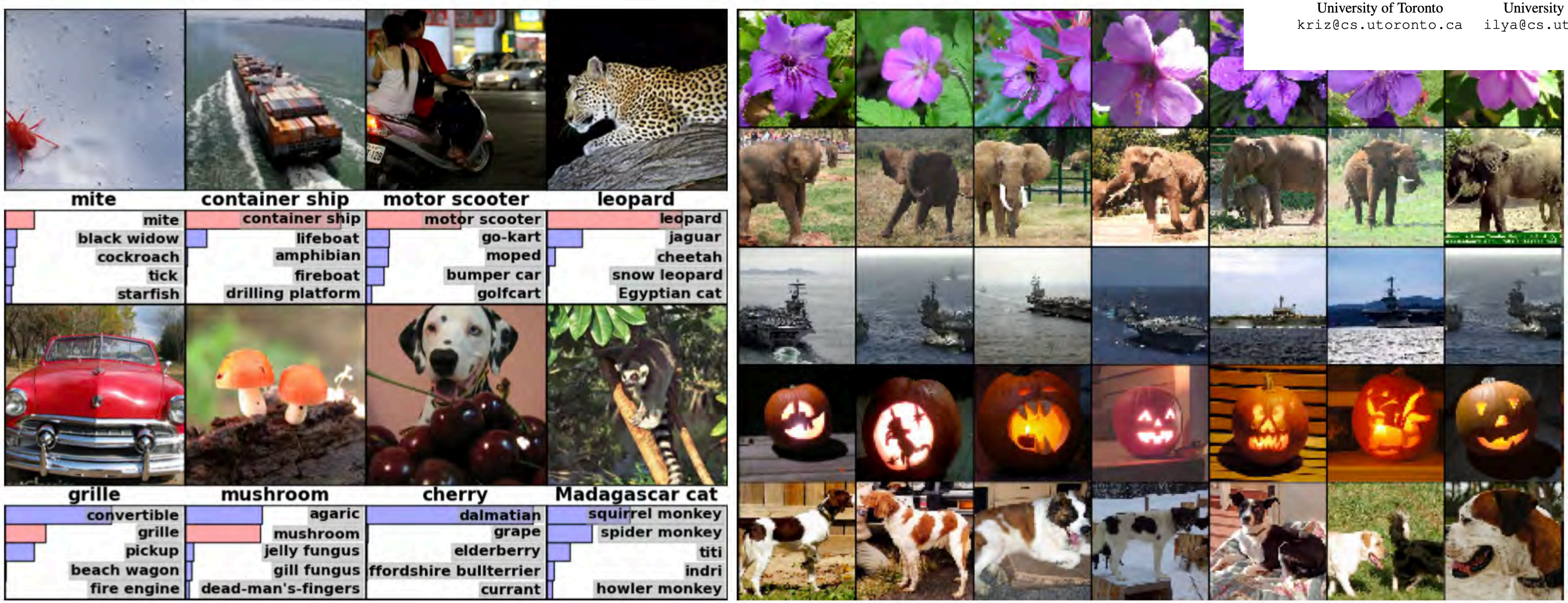
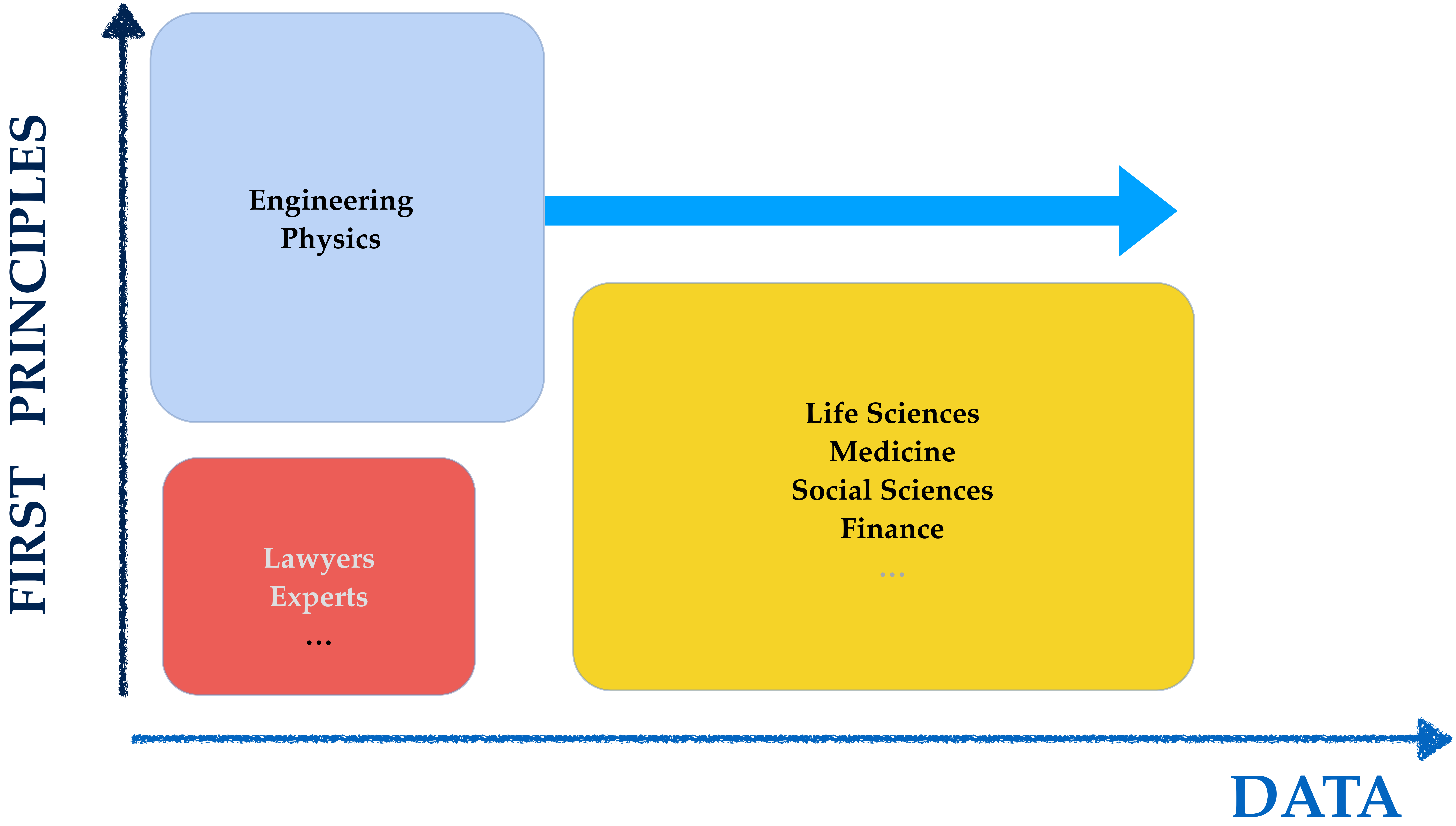


Figure 4: **(Left)** Eight ILSVRC-2010 test images and the five labels considered most probable by our model. The correct label is written under each image, and the probability assigned to the correct label is also shown with a red bar (if it happens to be in the top 5). **(Right)** Five ILSVRC-2010 test images in the first column. The remaining columns show the six training images that produce feature vectors in the last hidden layer with the smallest Euclidean distance from the feature vector for the test image.

SOLVING PROBLEMS

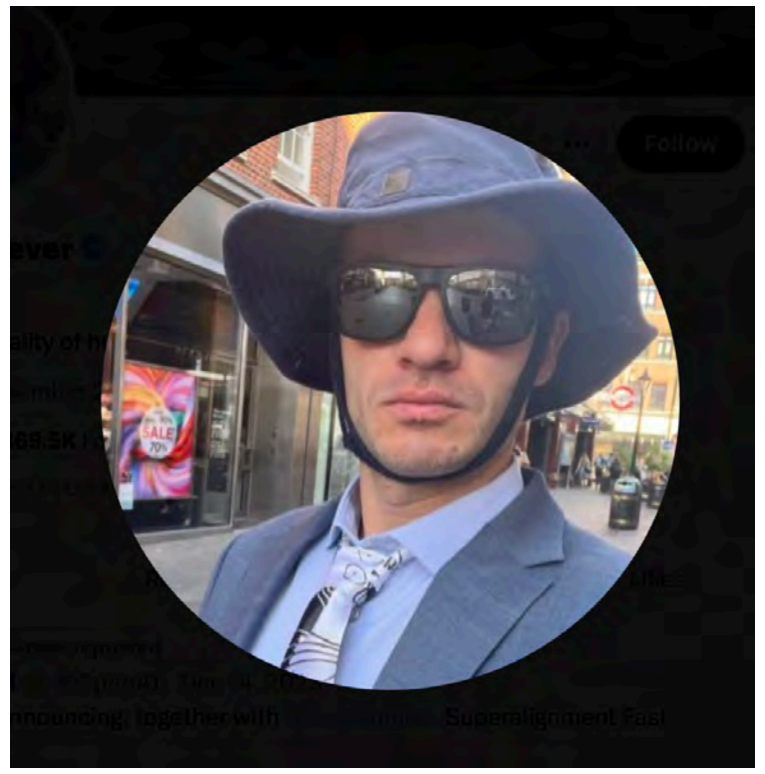


How to solve hard problems?

Use lots of training data.

And a big deep neural network.

And success is the only possible outcome.



Ilya Sutskever (2015),
co-founder of OpenAI


SCIENTIFIC COMPUTING
Mathematics
Exactness
Understanding
ALLOYS
ARTIFICIAL INTELLIGENCE
Architectures
Statistics
Goals

The Vexation of Patterns

✓ Machine Learning: Success for Pattern Recognition

➡ **Patterns** often present in Dynamical Systems

◉ Are there **Latent** spaces of Dynamical Systems?

➡ **Latent= Causal, Effective, Predictive,.....**

◉ **Can Machine Learning help identify them?**

IMAGE: De Chirico, The Vexation of the Thinker

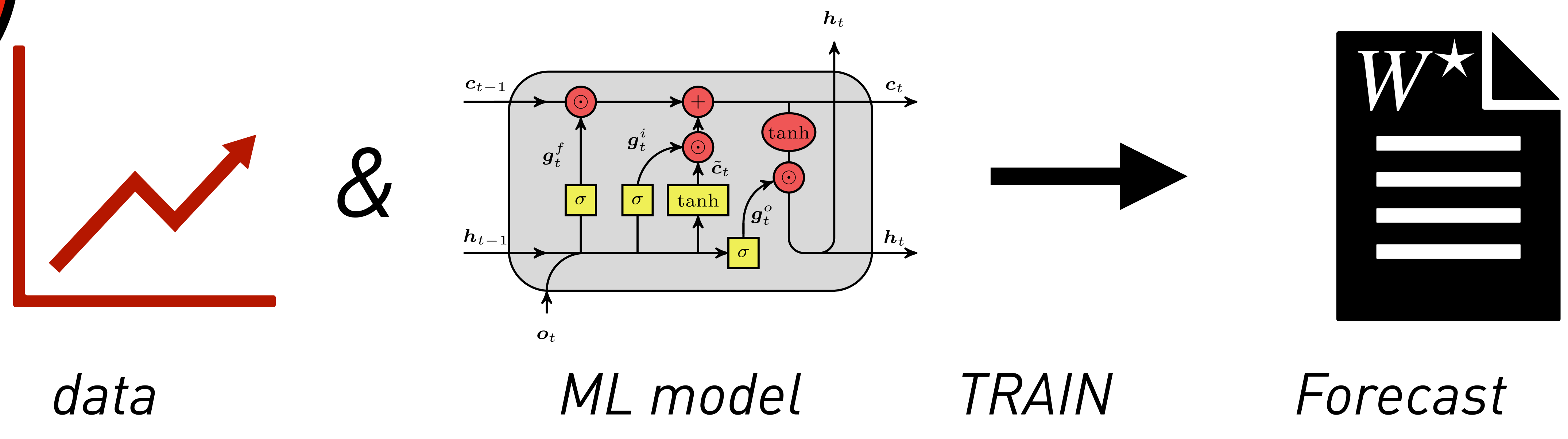
TIME

an ML Frontier

Spatiotemporal Forecasting with ML models

1

TRAIN



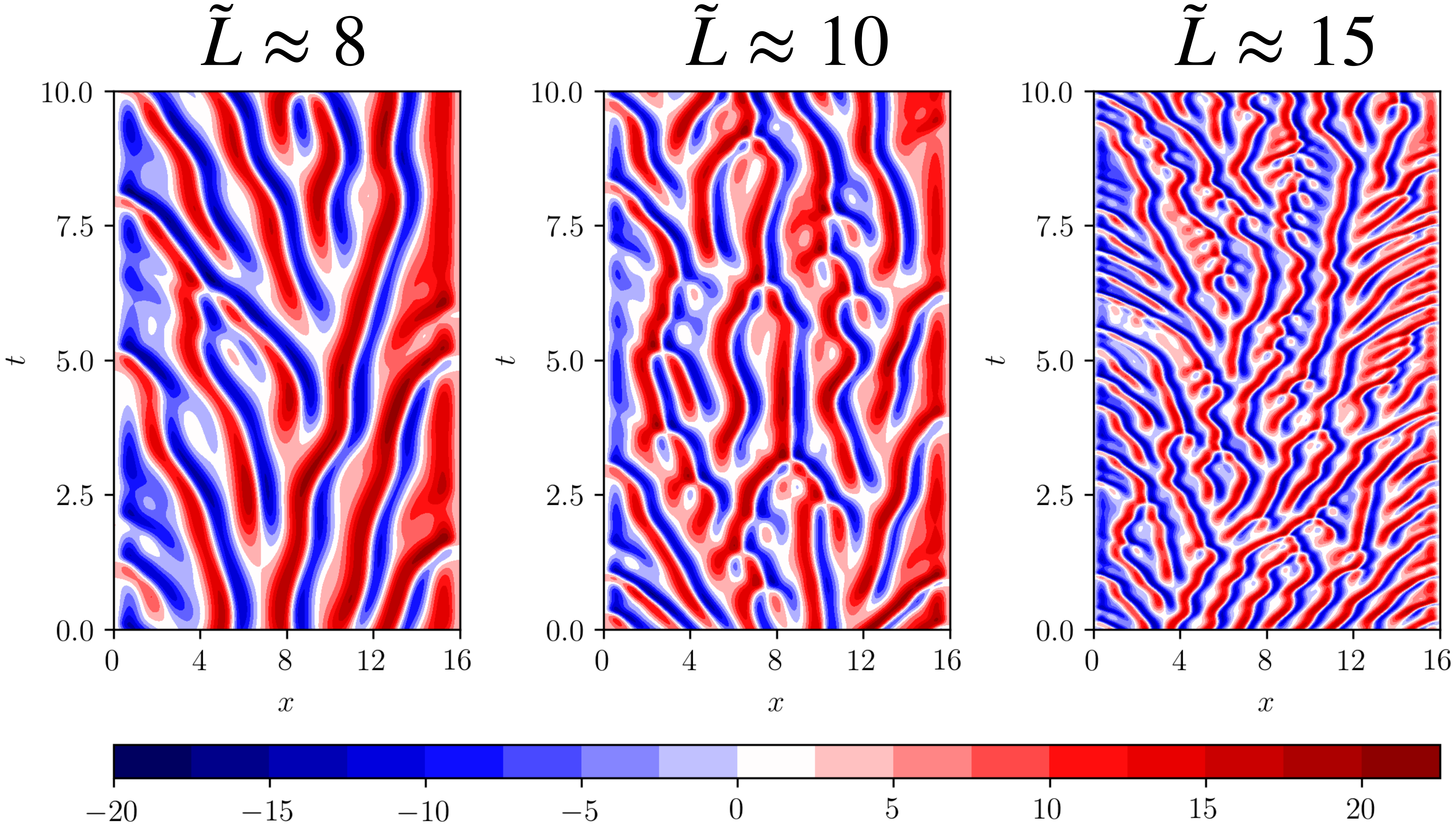
2

TEST

Can ML models forecast the dynamics of UNSEEN data?

NNs for Dynamical Systems

Kuramoto - Sivashinsky



$$\frac{\partial u}{\partial t} = -\nu \frac{\partial^4 u}{\partial x^4} - \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x},$$

$$u(0,t) = u(L,t) = \left.\frac{\partial u}{\partial x}\right|_{x=0} = \left.\frac{\partial u}{\partial x}\right|_{x=L} = 0,$$

$$u(x,0) = u_0(x),$$

$$x \in [0, L] \quad t \in [0, \infty]$$

$$\frac{du_i}{dt} = -\nu \frac{u_{i-2} - 4u_{i-1} + 6u_i - 4u_{i+1} + u_{i+2}}{\Delta x^4} - \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} - \frac{u_{i+1}^2 - u_{i-1}^2}{4\Delta x}$$

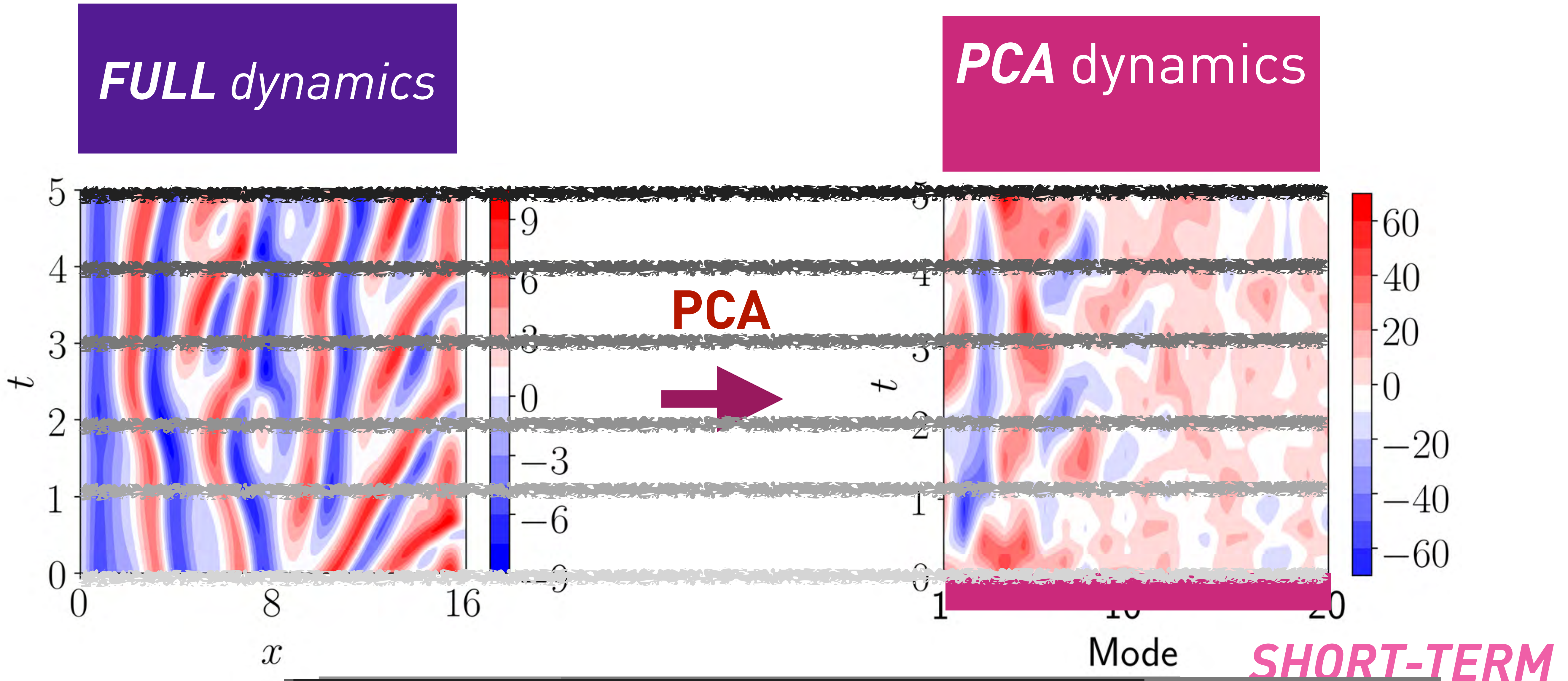
Integration with $dt = 0.02$ up to $T = 10^4$

500.000 **samples**

Forecasting using **LATENT** Dynamics

LATENT DYNAMICS:
PCA, AUTO- ENCODERS,....

NEXT:
LATENT DYNAMICS
FOR DIFFUSION MODELS



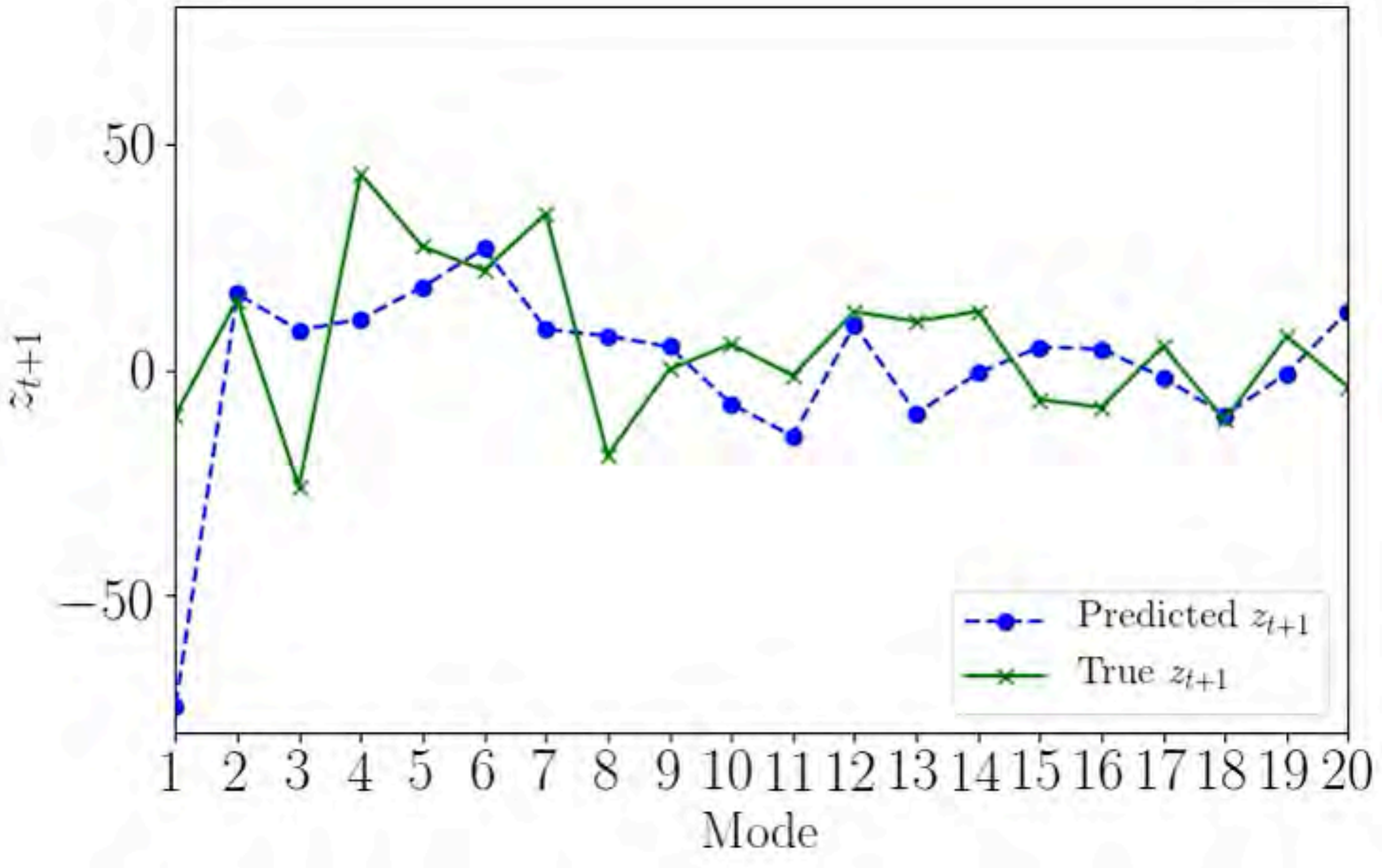
SHORT-TERM HISTORY
TRAIN

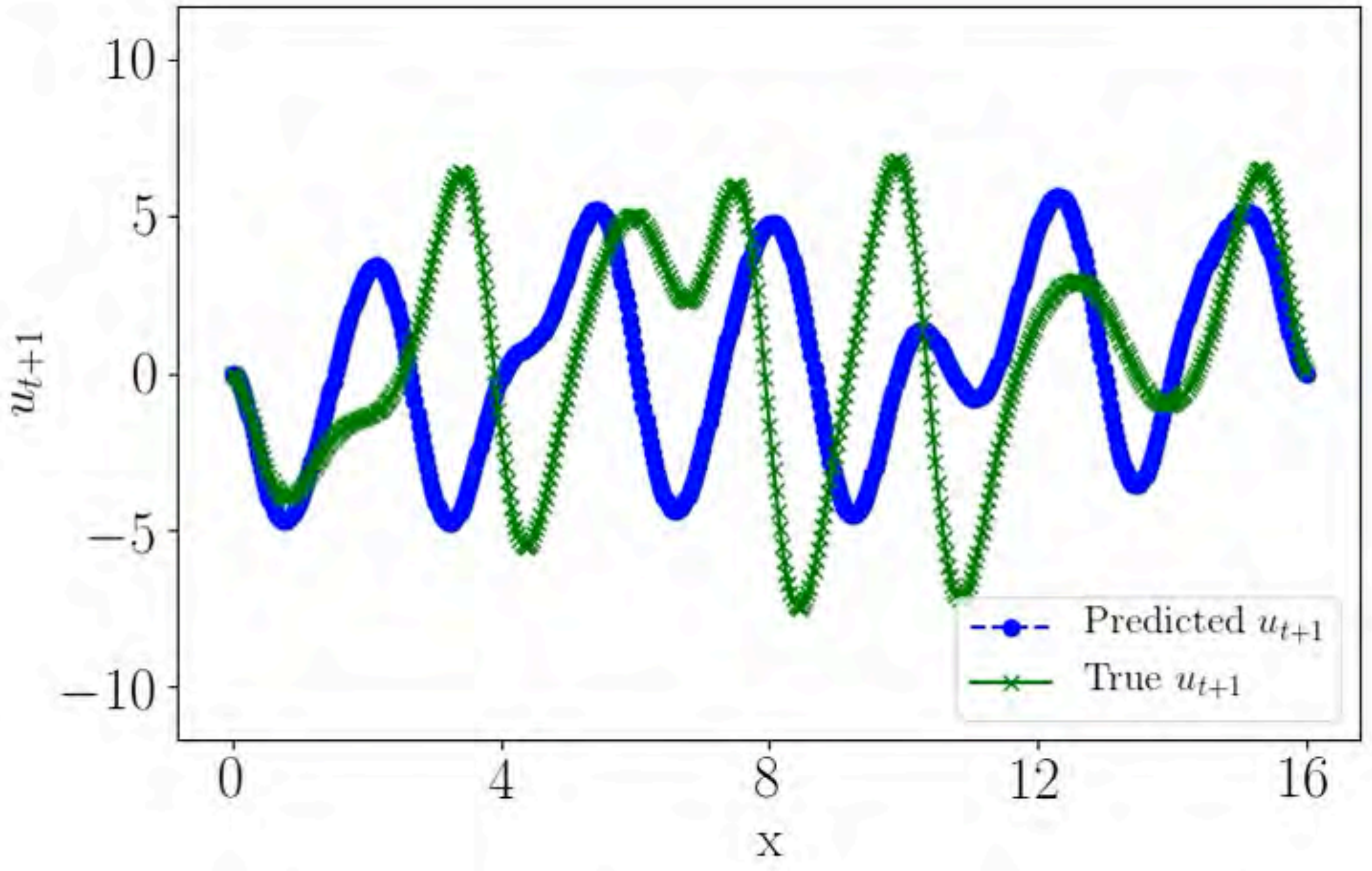
RNNs (and ML in general) **FAIL** to FORECAST (Sooner or Later) CHAOTIC SYSTEMS

reduced space

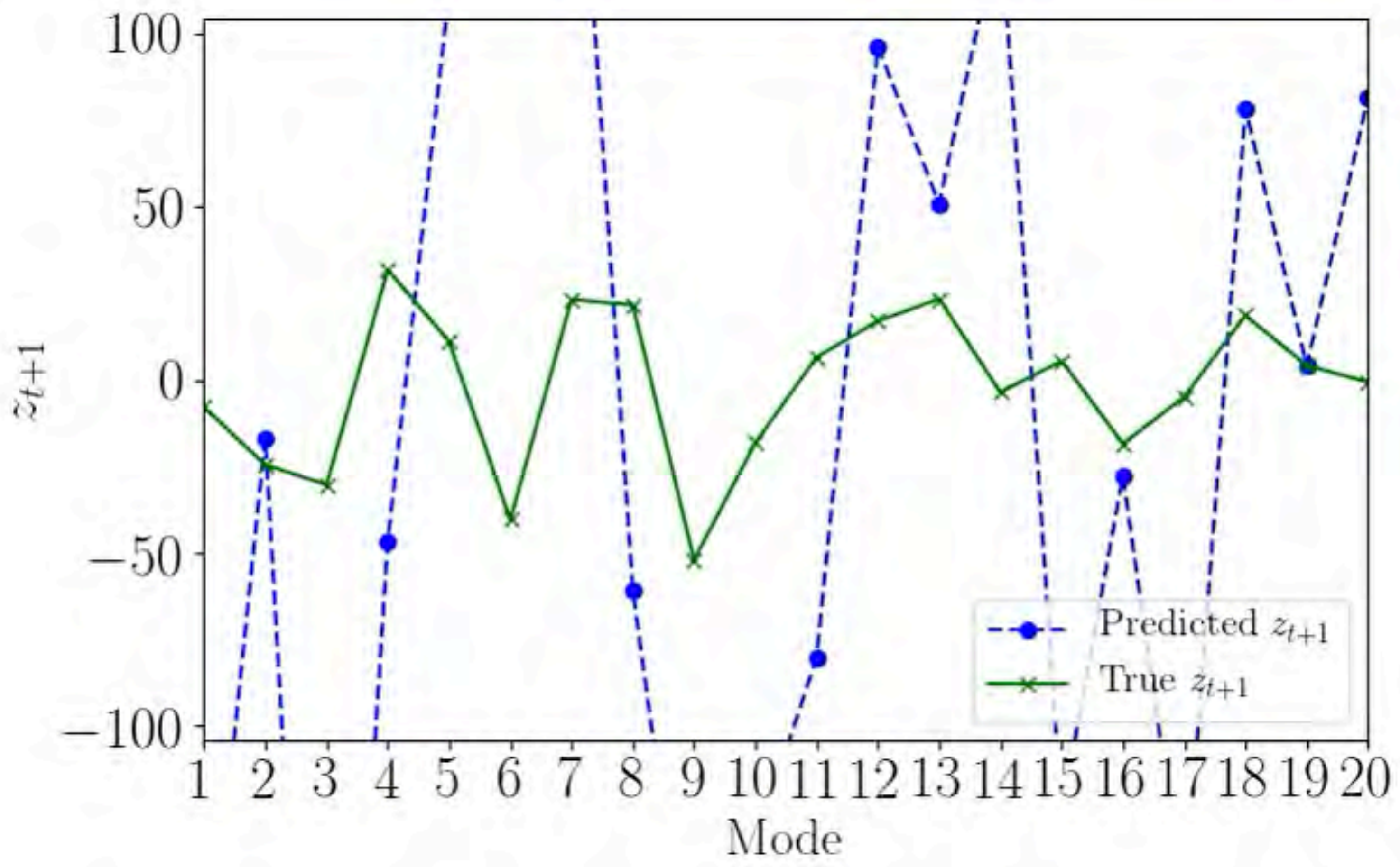
high-dimensional space

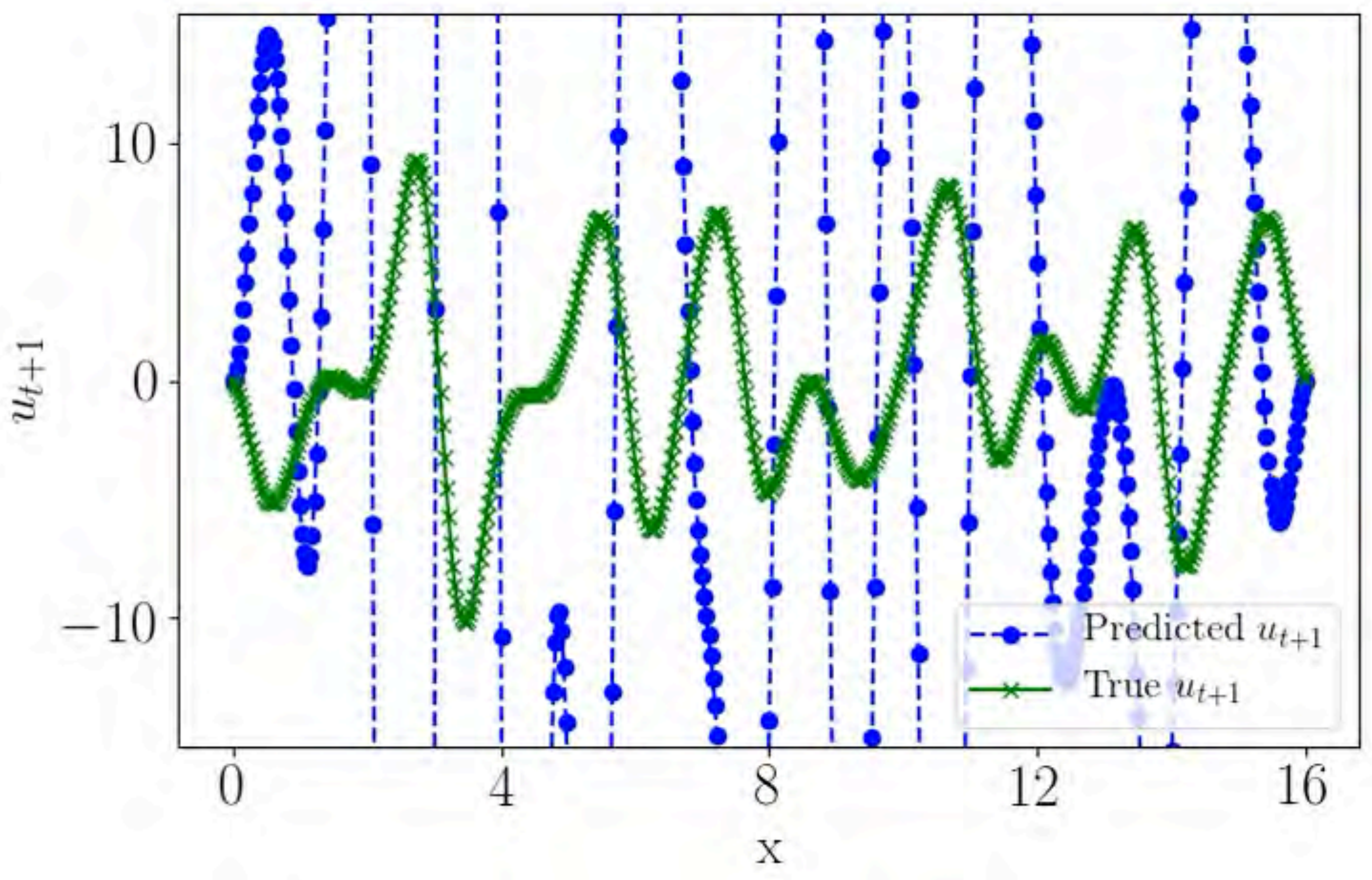
$\tilde{L} \approx 8$





$\tilde{L} \approx 10$





PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

Check for updates

Cite this article: Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P. 2018 Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A* **474**: 20170844. http://dx.doi.org/10.1098/rspa.2017.0844

Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks

Pantelis R. Vlachas[1], Wonmin Byeon[1], Zhong Y. Wan[2], Themistoklis P. Sapsis[2] and Petros Koumoutsakos[1]

[1]Chair of Computational Science, ETH Zurich, Clausiusstrasse 33, Zurich, CH-8092, Switzerland
[2]Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139, USA

2017

PLOS | ONE

RESEARCH ARTICLE

Data-assisted reduced-order modeling of extreme events in complex dynamical systems

Zhong Yi Wan[1], Pantelis Vlachas[2], Petros Koumoutsakos[2], Themistoklis Sapsis[1]*

1 Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, United States of America, **2** Chair of Computational Science, ETH Zurich, Zurich, Switzerland

* sapsis@mit.edu

2018

Neural Networks 126 (2020) 191–217

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

ELSEVIER

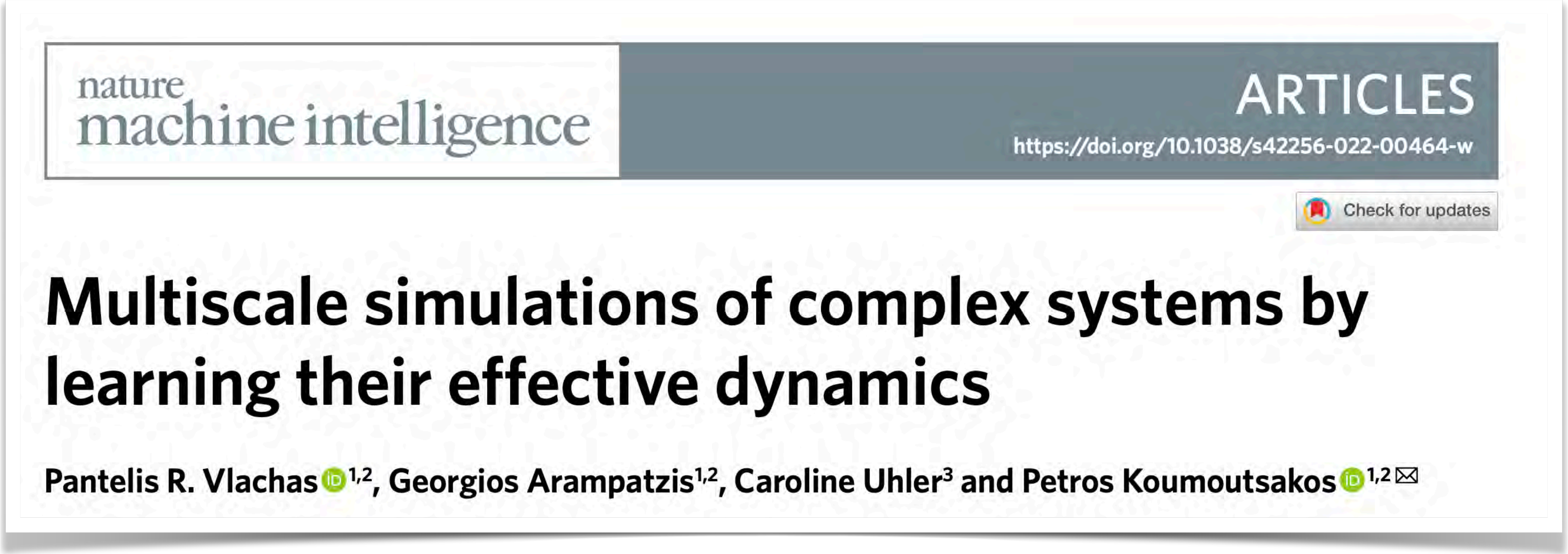
Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics

P.R. Vlachas [a], J. Pathak [b,c], B.R. Hunt [d,e], T.P. Sapsis [f], M. Girvan [b,c,d], E. Ott [b,c,g], P. Koumoutsakos [a,*]

[a] *Computational Science and Engineering Laboratory, ETH Zürich, Clausiusstrasse 33, Zürich CH-8092, Switzerland*
[b] *Institute for Research in Electronics and Applied Physics, University of Maryland, College Park, MD 20742, USA*
[c] *Department of Physics, University of Maryland, College Park, MD 20742, USA*
[d] *Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742, USA*
[e] *Department of Mathematics, University of Maryland, College Park, MD 20742, USA*
[f] *Department of Mechanical Engineering, Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, MA 02139-4307, USA*
[g] *Department of Electrical and Computer Engineering, University of Maryland, MD 20742, USA*

2020

Learning Effective Dynamics


nature machine intelligence

ARTICLES

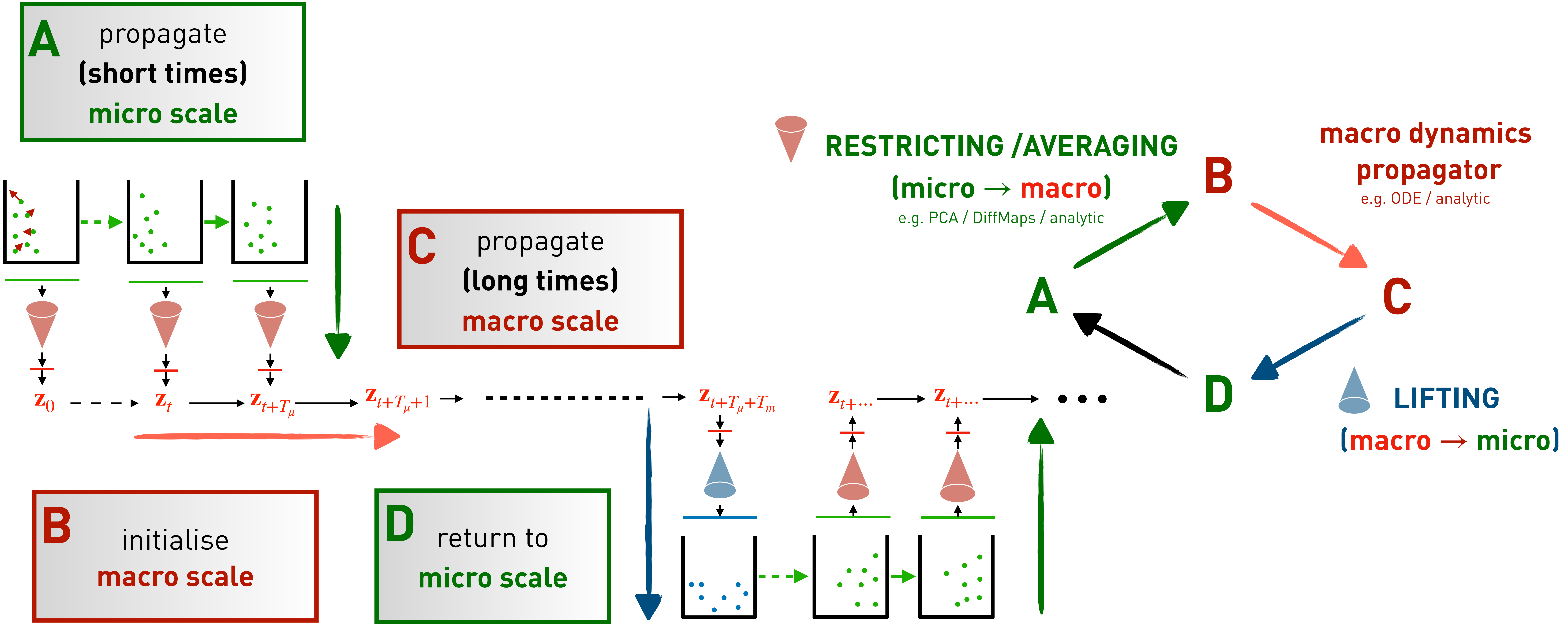
https://doi.org/10.1038/s42256-022-00464-w

Check for updates

Multiscale simulations of complex systems by learning their effective dynamics

Pantelis R. Vlachas[1,2], Georgios Arampatzis[1,2], Caroline Uhler[3] and Petros Koumoutsakos[1,2]

Equation-Free Framework - Yannis Kevrekidis



Theodoropoulos, C.; Qian, Y.H. and Kevrekidis, I.G. (2000). *Proc. Natl. Acad. Sci. 97: 9840-9845.*

Gear, C.W.; Kevrekidis, I.G. and Theodoropoulos, C. (2002). *Computers and Chemical Engineering 26: 941-963.*

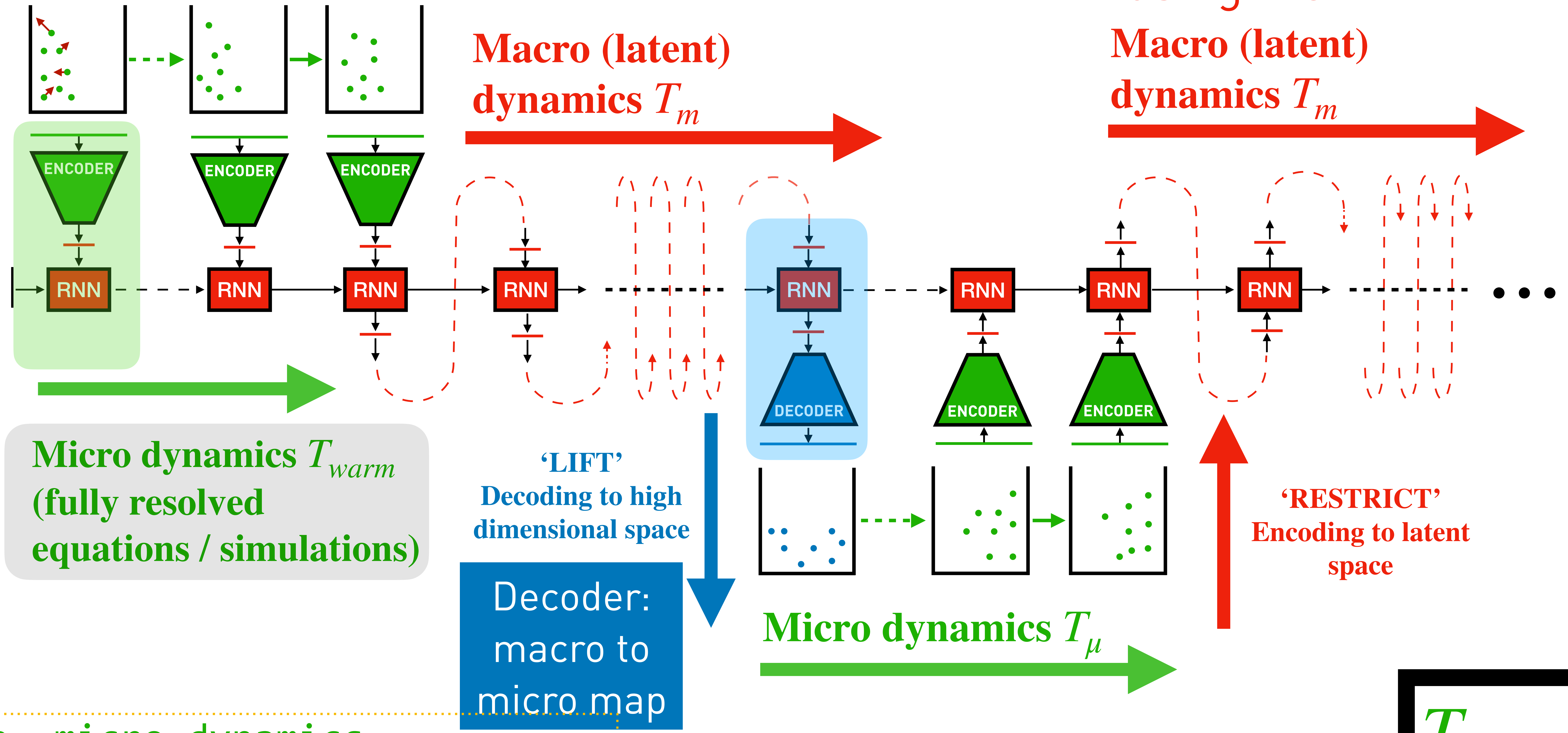
AND MANY MANY MORE

Learning Effective Dynamics

PR Vlachas, G Arampatzis, C Uhler, P Koumoutsakos,
Multiscale Simulations of Complex Systems by Learning their Effective Dynamics,
Nature Machine Intelligence, (2022)

Evolve latent dynamics with RNN-LSTM

Update the macro dynamics using the RNN



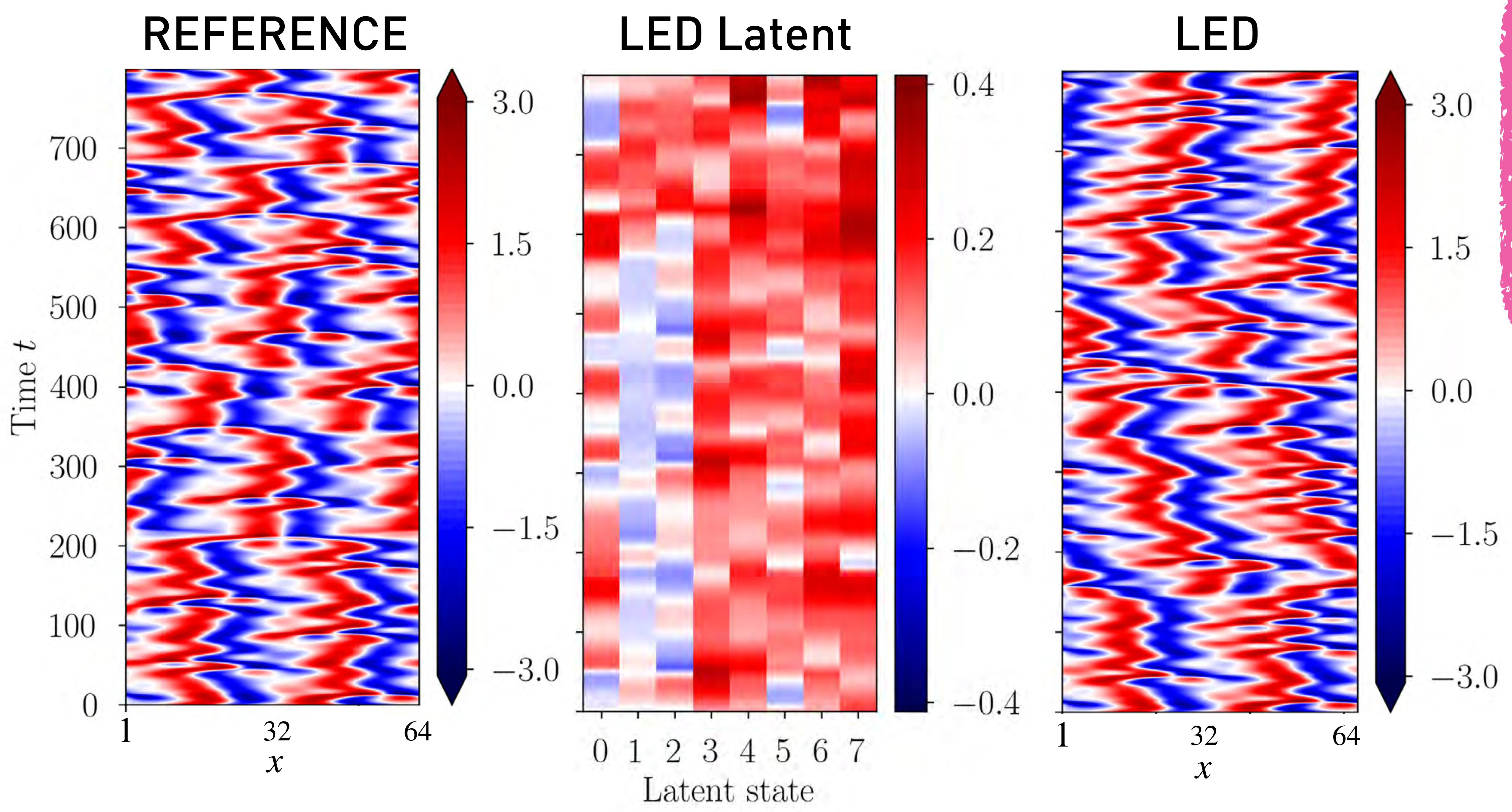
I. Evolve micro dynamics
II. Project to Latent Space with AEs
III. Train RNN-LSTM

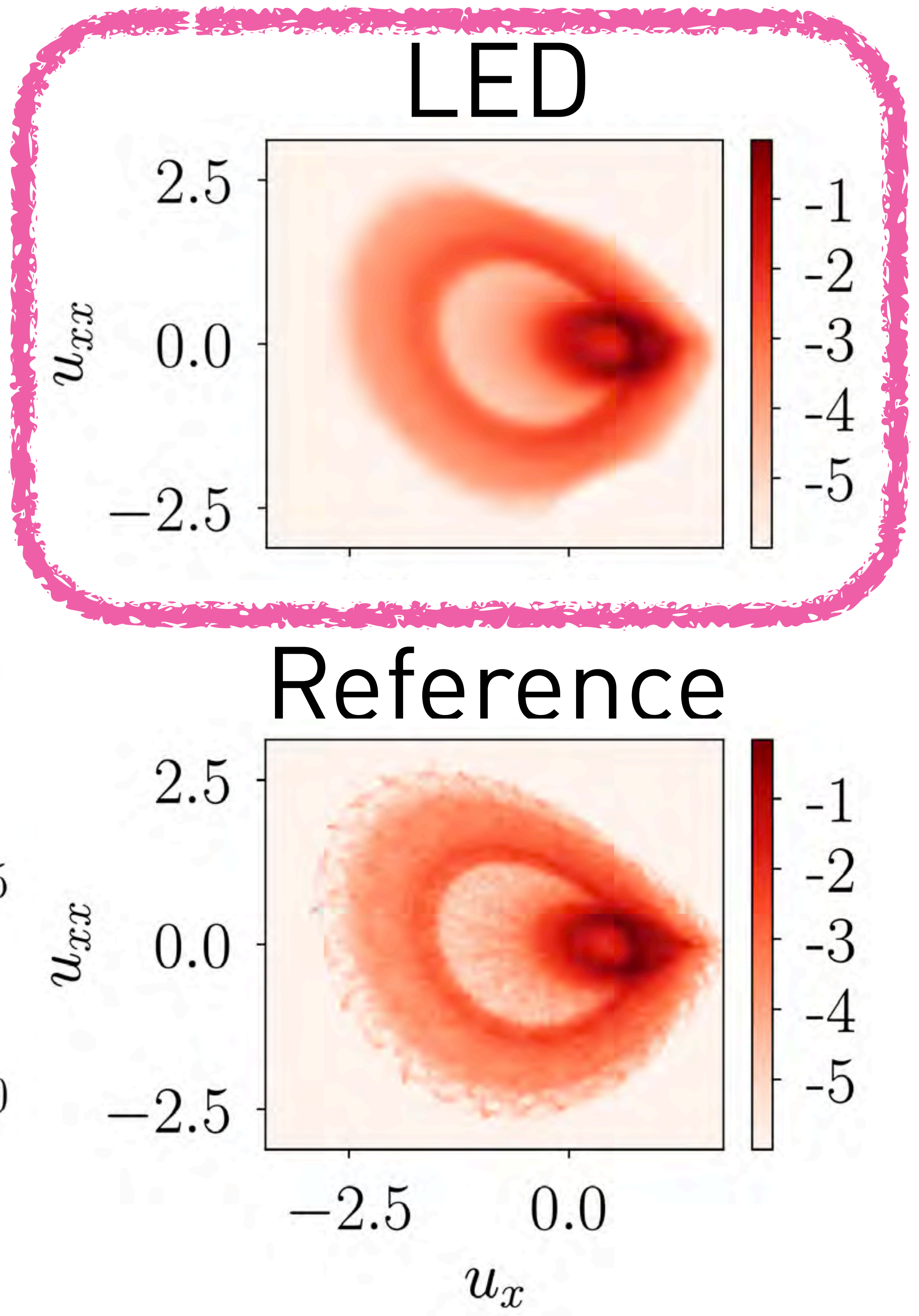
Update the micro dynamics (and continue training of RNN)

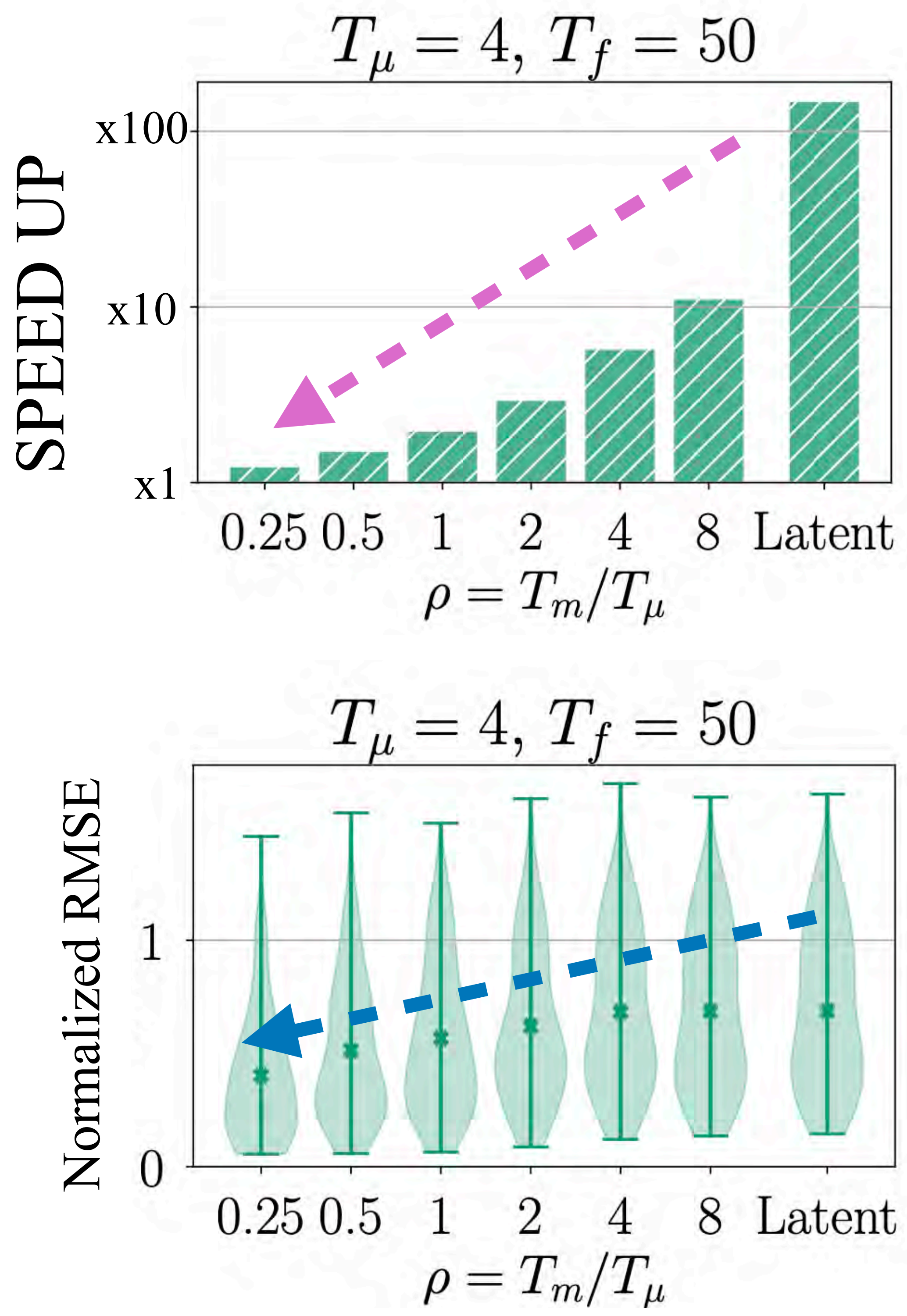
$$T_{warm}, T_\mu \ll T_m$$

Kuramoto-Sivashinsky $L = 22$

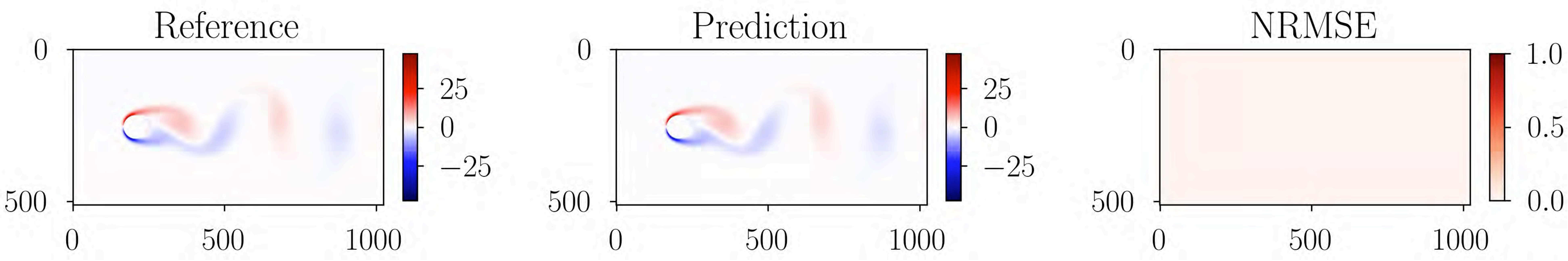
PR Vlachas, G Arampatzis, C Uhler, P Koumoutsakos,
Multiscale Simulations of Complex Systems by Learning their Effective Dynamics,
Nature Machine Intelligence, (2022)

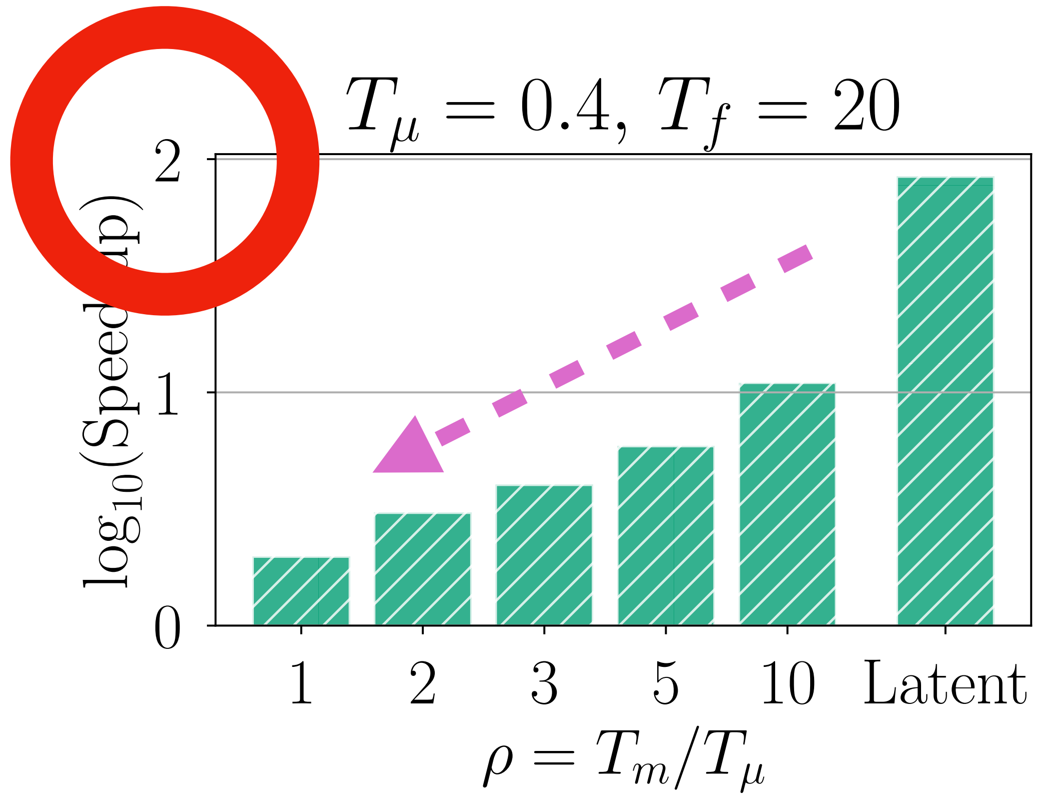






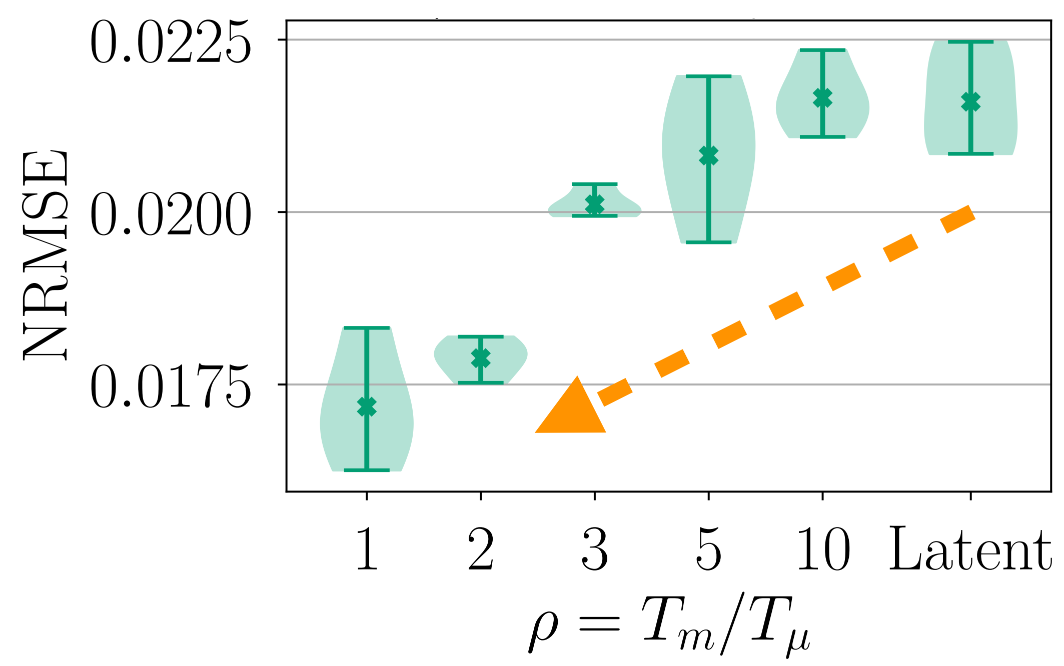
Cylinder at $Re = 100$ - (LED $d_z = 2$)

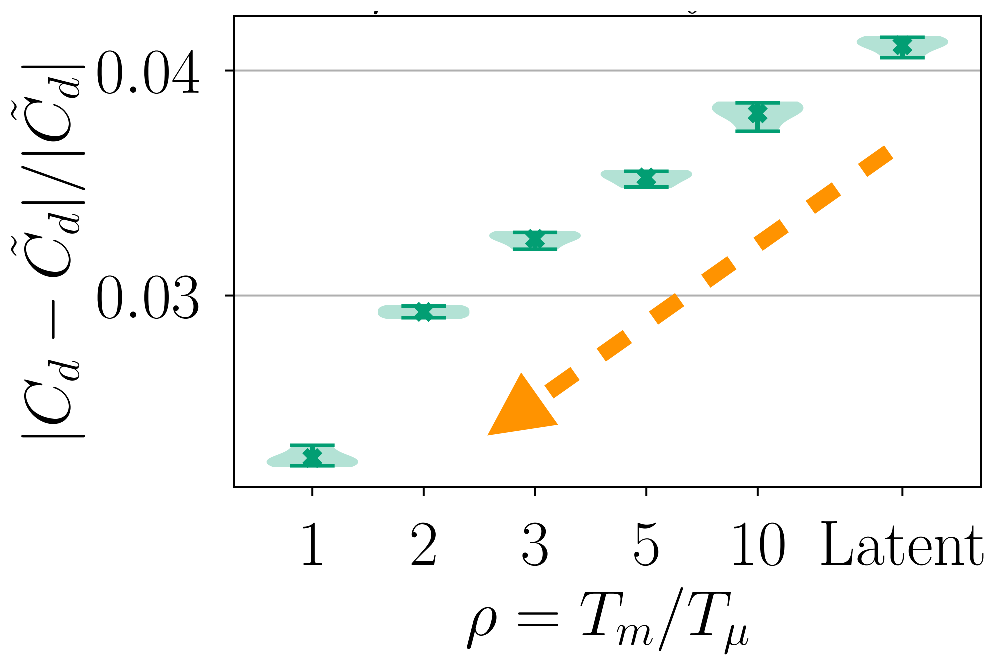


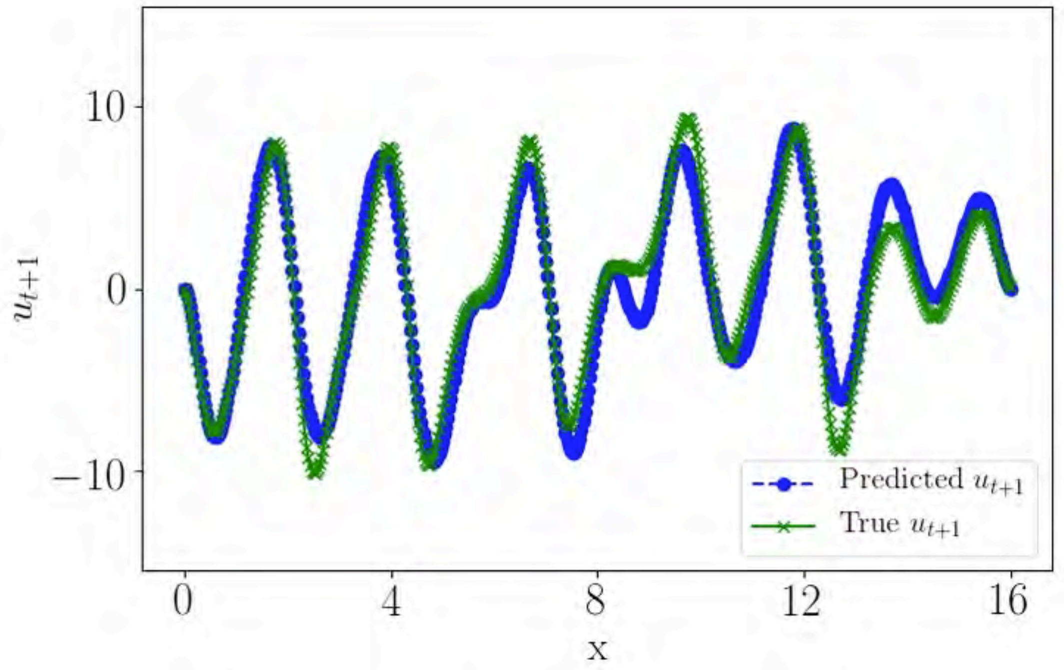


- Micro solver: Finite Differences solver (CubimUP2D) employing 12 cores
- State: velocity in x- and y- direction and pressure $\mathbf{s}_t \in \mathbb{R}^{3\times512\times1024}$
- LED with latent dimension of $d_\mathbf{z} = 2$, $\Delta t = 0.2$

- LED captures **long-term evolution** of velocity and pressure fields (low NRMSE)
- LED is up to **two orders of magnitude** faster than CubismUP2D
- Recovers drag coefficient with $\approx 2-4\,\%$ error





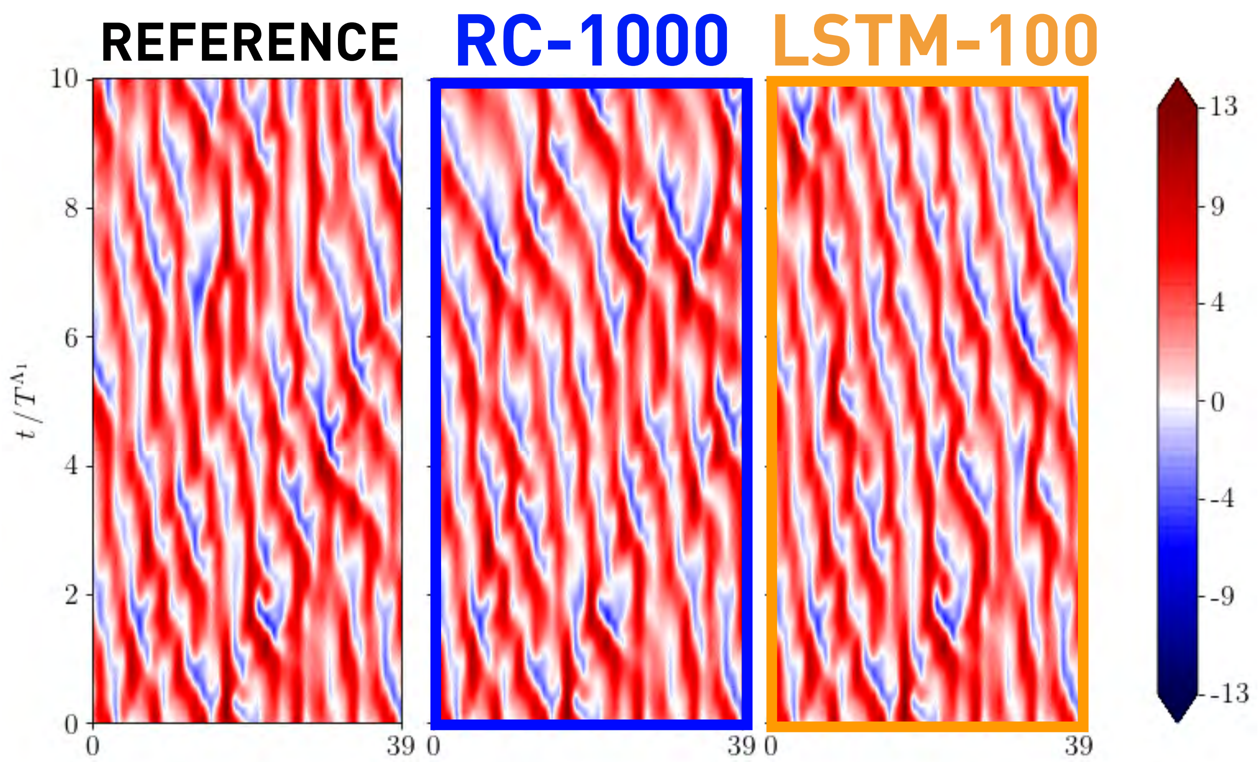


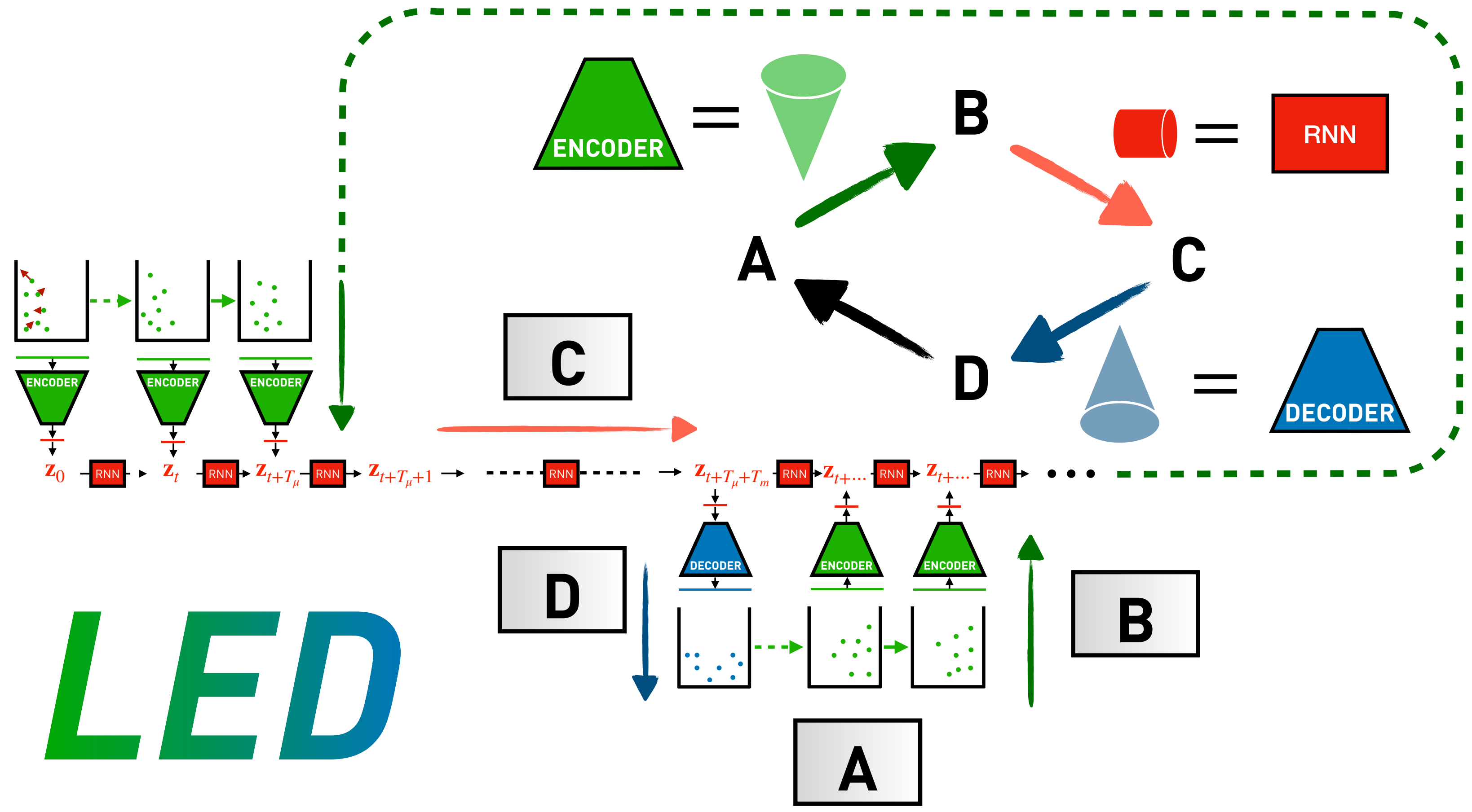
HYBRID LSTM - MSM

$$\dot{z}_t = \begin{cases} \text{LSTM}^{\mathbf{W}}(z_t, z_{t-1}, z_{t-2}, \ldots) & \textbf{if } p_{train}(z_t) \geq \theta \\ \text{MSM}^{\zeta,c}(z_t) & \textbf{if } p_{train}(z_t) < \theta \end{cases}$$

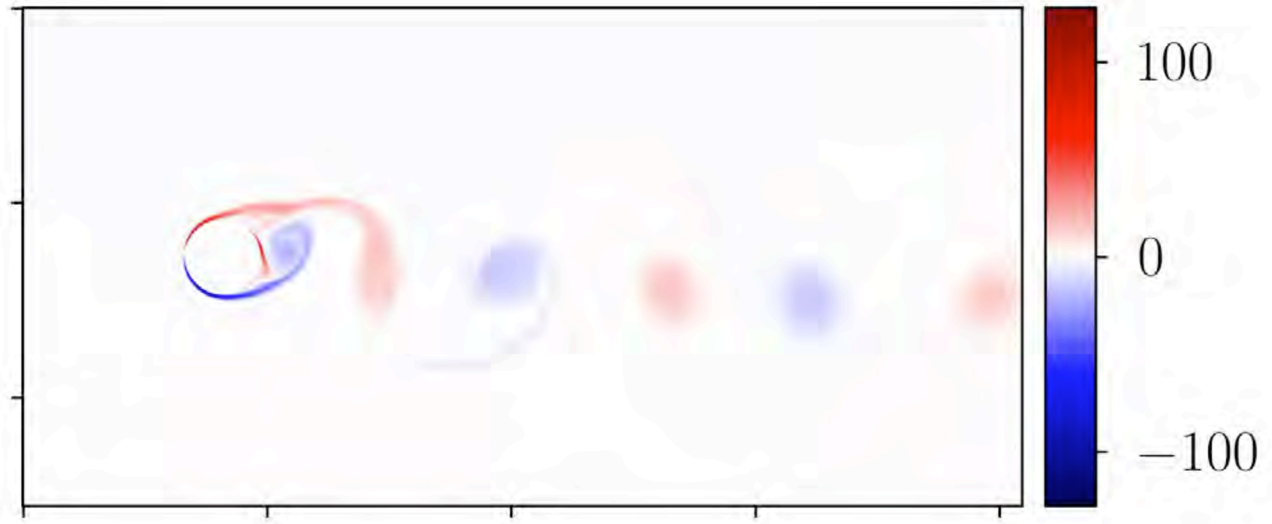
PR Vlachas, W Byeon, Z Wan, T Sapsis, P Koumoutsakos, *Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks,* **Proc. Roy. Soc. A , 2018**

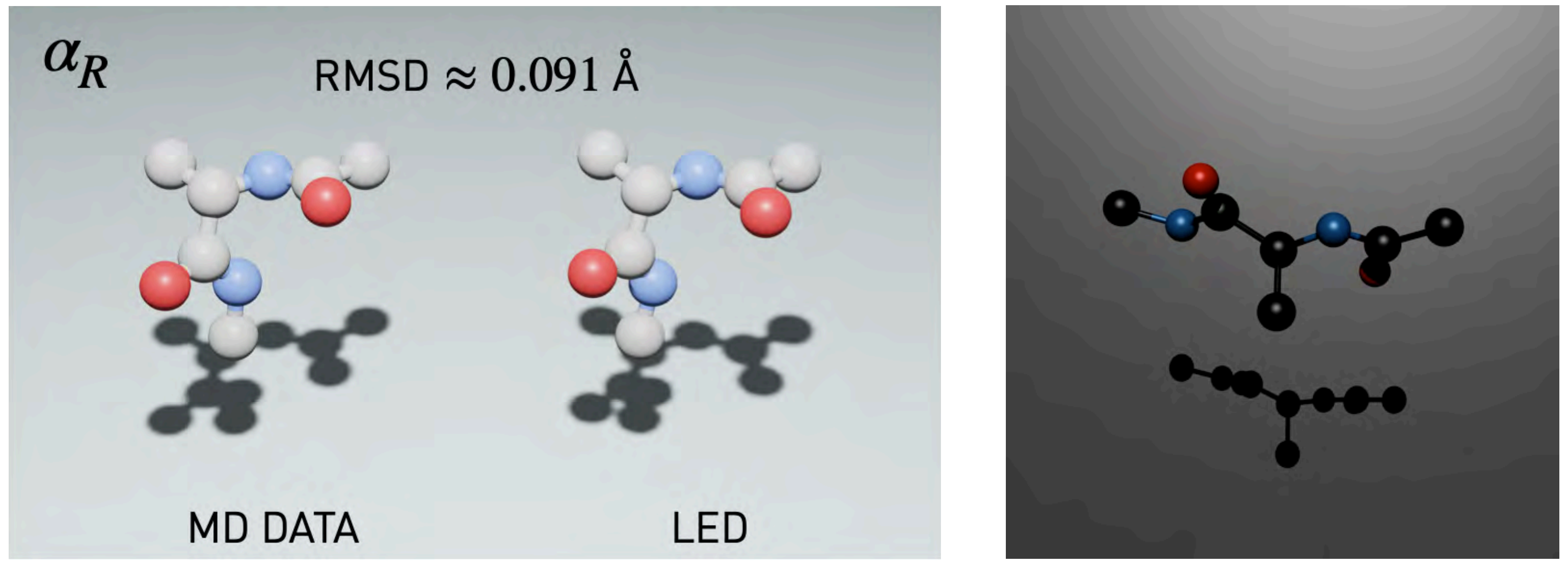
PR Vlachas , J Pathak , BR Hunt , TP Sapsis , M Girvan, E Ott and P Koumoutsakos, *Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics,* **Journal of Neural Networks, 2020**



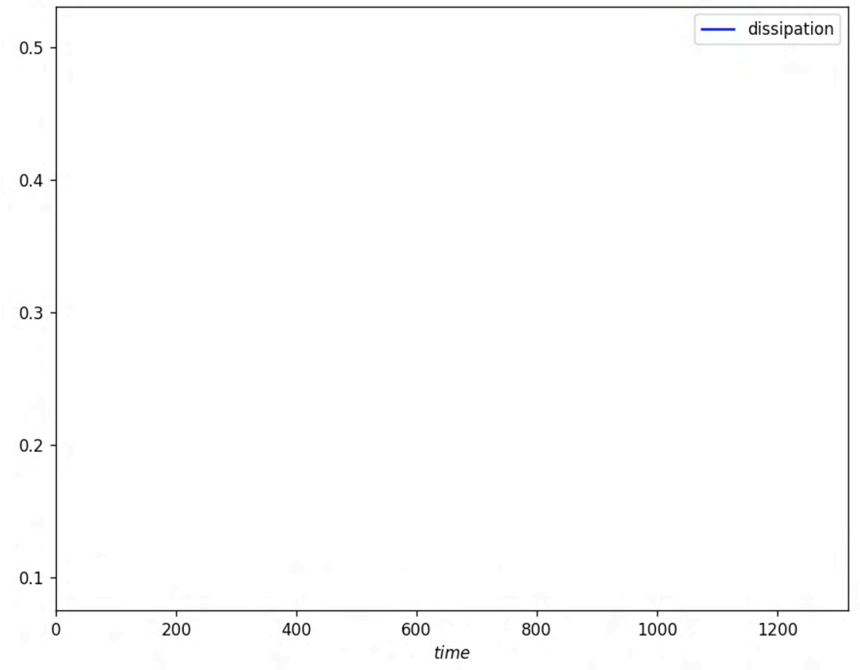


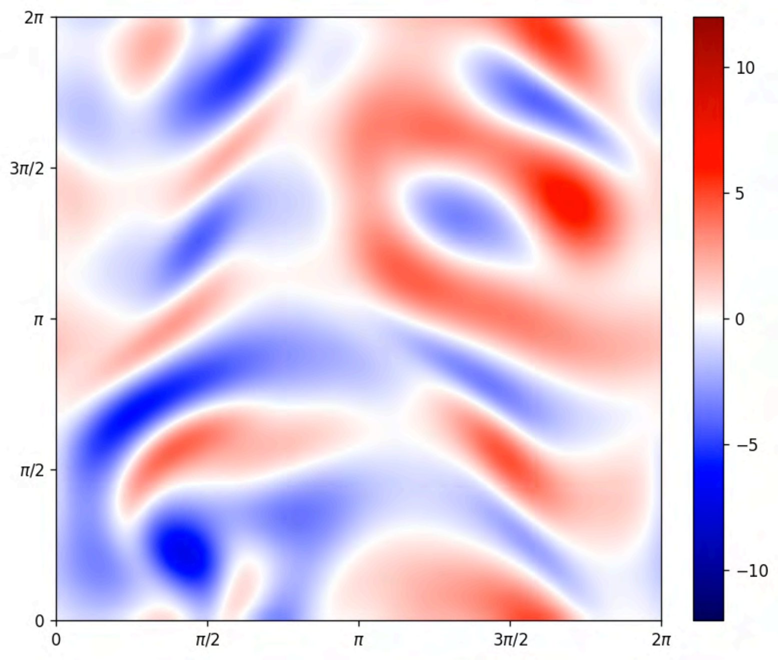
PR Vlachas, G Arampatzis, C Uhler, P Koumoutsakos, *Multiscale Simulations of Complex Systems by Learning their Effective Dynamics,* **Nature Machine Intelligence, 2022**





PR Vlachas, J Zavadlav, M Praprotnik, P Koumoutsakos, *Accelerated Simulations of Molecular Systems through Learning of their Effective Dynamics,* **Journal of Chemical Theory & Computation, 2021**

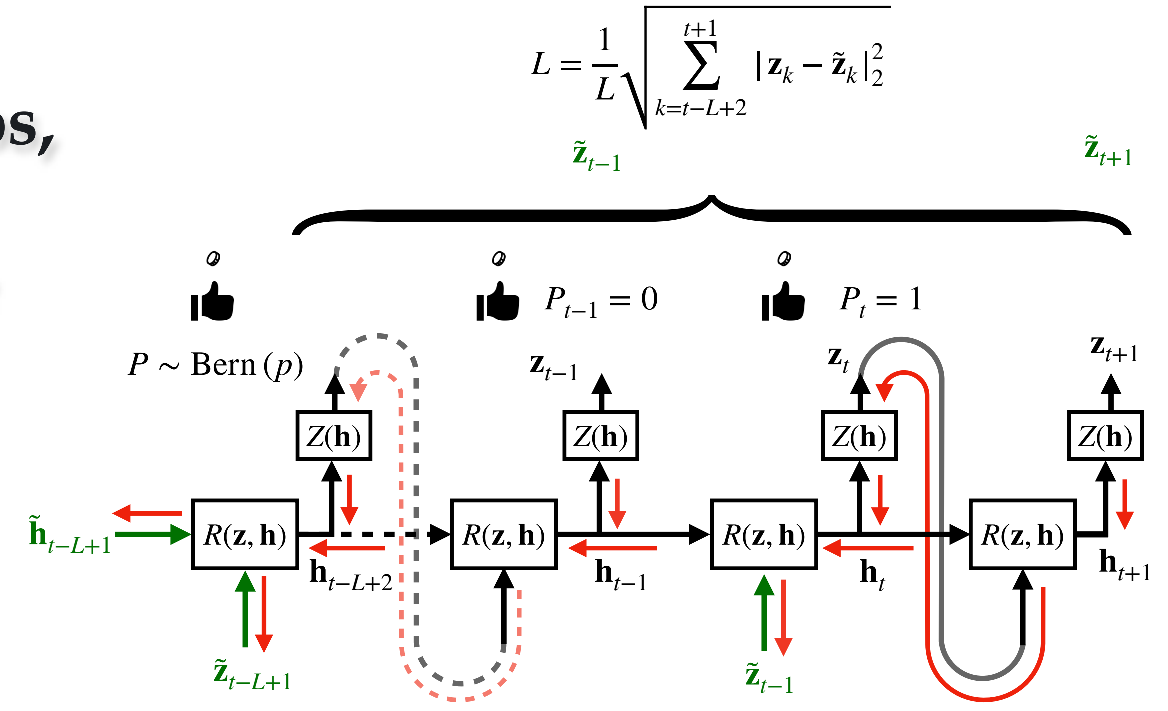




ZY Wan, P Vlachas, P Koumoutsakos, T Sapsis, *Data-assisted reduced-order modeling of extreme events in complex dynamical systems,* **PloS one, 2018**

$$\dot{\xi}_t = F(\xi_t) + \tilde{G}(\xi_t, \xi_{t-1}, \xi_{t-2}, \ldots)$$

PR Vlachas, P Koumoutsakos, *Scheduled Autoregressive Backpropagation Through Time for Long-Term Forecasting,* **Neural Networks, 2024**



Adaptive **LED**

interpretable **LED**

Computer Methods in Applied Mechanics and Engineering

Volume 415, 1 October 2023, 116204

Adaptive learning of effective dynamics for online modeling of complex systems

Ivica Kičić [a], Pantelis R. Vlachas [a b], Georgios Arampatzis [a b], Michail Chatzimanolakis [a b], Leonidas Guibas [c], Petros Koumoutsakos [b]

arXiv > stat > arXiv:2309.05812

Statistics > Machine Learning

[Submitted on 11 Sep 2023]

Interpretable learning of effective dynamics for multiscale systems

Emmanuel Menier, Sebastian Kaltenbach, Mouadh Yagoubi, Marc Schoenauer, Petros Koumoutsakos

The modeling and simulation of high-dimensional multiscale systems is a critical challenge across all areas of science and engineering. It is broadly believed that even with today's computer advances resolving all spatiotemporal scales described by the governing equations remains a remote target. This realization has prompted intense efforts to develop model order reduction techniques. In recent years, techniques based on deep recurrent neural networks have produced promising results for the modeling and simulation of complex spatiotemporal systems and offer large flexibility in model development as they can incorporate experimental and computational data. However, neural networks lack interpretability, which limits their utility and generalizability across complex systems. Here we propose a novel framework of Interpretable Learning Effective Dynamics (iLED) that offers comparable accuracy to state-of-the-art recurrent neural network-based approaches while providing the added benefit of interpretability. The iLED framework is motivated by Mori-Zwanzig and Koopman operator theory, which justifies the choice of the specific architecture. We demonstrate the effectiveness of the proposed framework in simulations of three benchmark multiscale systems. Our results show that the iLED framework can generate accurate predictions and obtain interpretable dynamics, making it a promising approach for solving high-dimensional multiscale systems.

Generative LED

arXiv > cs > arXiv:2402.17157

Search...

Help | Adv

Computer Science > Machine Learning

[Submitted on 27 Feb 2024]

Generative Learning for Forecasting the Dynamics of Complex Systems

Han Gao, Sebastian Kaltenbach, Petros Koumoutsakos

We introduce generative models for accelerating simulations of complex systems through learning and evolving their effective dynamics. In the proposed Generative Learning of Effective Dynamics (G-LED), instances of high dimensional data are down sampled to a lower dimensional manifold that is evolved through an auto-regressive attention mechanism. In turn, Bayesian diffusion models, that map this low-dimensional manifold onto its corresponding high-dimensional space, capture the statistics of the system dynamics. We demonstrate the capabilities and drawbacks of G-LED in simulations of several benchmark systems, including the Kuramoto-Sivashinsky (KS) equation, two-dimensional high Reynolds number flow over a backward-facing step, and simulations of three-dimensional turbulent channel flow. The results demonstrate that generative learning offers new frontiers for the accurate forecasting of the statistical properties of complex systems at a reduced computational cost.

Subjects: **Machine Learning (cs.LG)**; Computational Physics (physics.comp-ph); Fluid Dynamics (physics.flu-dyn); Machine Learning (stat.ML)
Cite as: arXiv:2402.17157 **[cs.LG]**
(or arXiv:2402.17157v1 **[cs.LG]** for this version)
https://doi.org/10.48550/arXiv.2402.17157

Submission history

From: Sebastian Kaltenbach [view email]
[v1] Tue, 27 Feb 2024 02:44:40 UTC (40,049 KB)

Nature Communications (accepted)

Generative AI : Probabilistic Approach to Unsupervised Learning

Enormous progress in ***unsupervised learning*** using ***generative models***

Stable Diffusion

DALL-E (Open AI)

BREAKTHROUGH : pose learning as a problem of ***density estimation:***

View the data $\{x_i\}_{i=1}^{n}$ as samples from the unknown probability distribution μ:

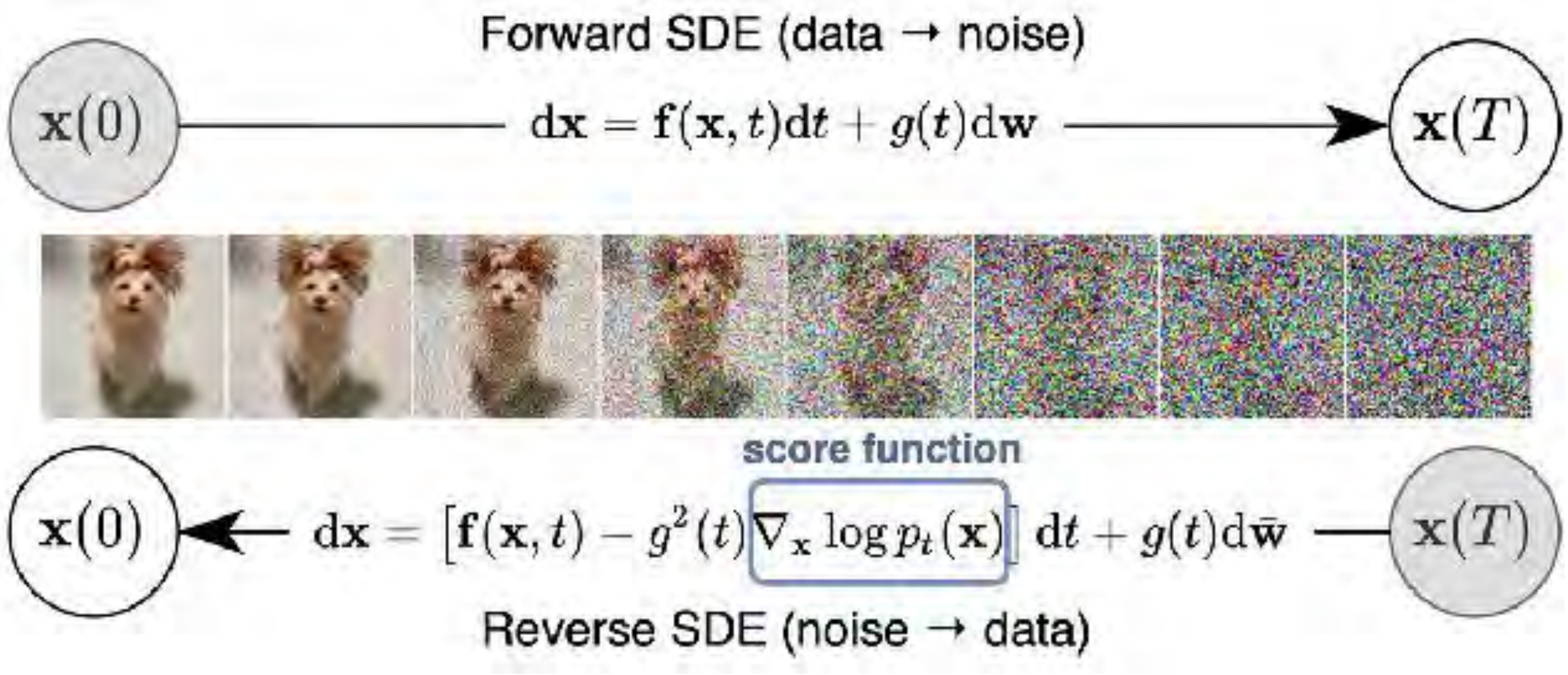
- calculate an estimate $\hat{\mu}$ of μ , and
- generate new data via sampling of $\hat{\mu}$.

Score-Based Diffusion Models

Song et al. arXiv:2011.13456 (2021);
*Hyvärinen JMLR **6** (2005);*
*Vincent, Neural Comp. **23**, 1661 (2011)*

Given data from the target μ_1:

- *Devolve them into Gaussian noise* using e.g. an Ornstein-Ulhenbeck process;
- *Time-reverse the SDE* to generate new samples from μ_1 from samples from $N(0,Id)$;



From Song's paper

*Builds a **path in distribution space** between μ_1 and $N(0,Id)$;*

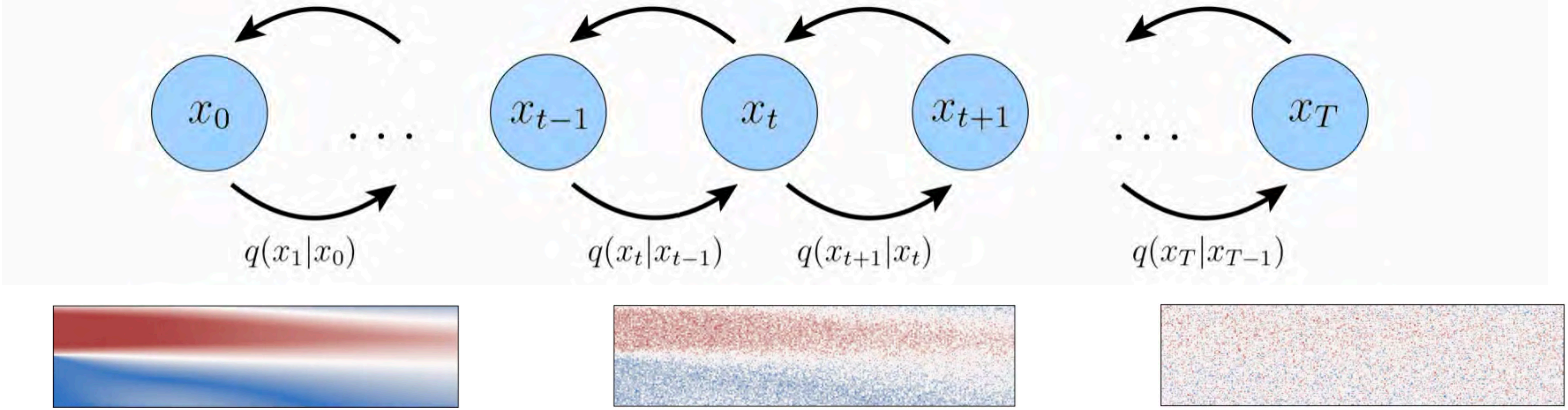
*Reduces problem to the **simulation-free regression of the score**.*

GUIDED Variational Diffusion Model

https://calvinyluo.com/2022/08/26/diffusion-tutorial.html

Diffusion model “learns” to reverse this process with guidance z_0

$p(x_0 \mid x_1; z_0)$ $p(x_{t-1} \mid x_t; z_0)$ $p(x_t \mid x_{t+1}; z_0)$ $p(x_{T-1} \mid x_T; z_0)$



1. Input is steadily noised until it becomes identical to Gaussian noise

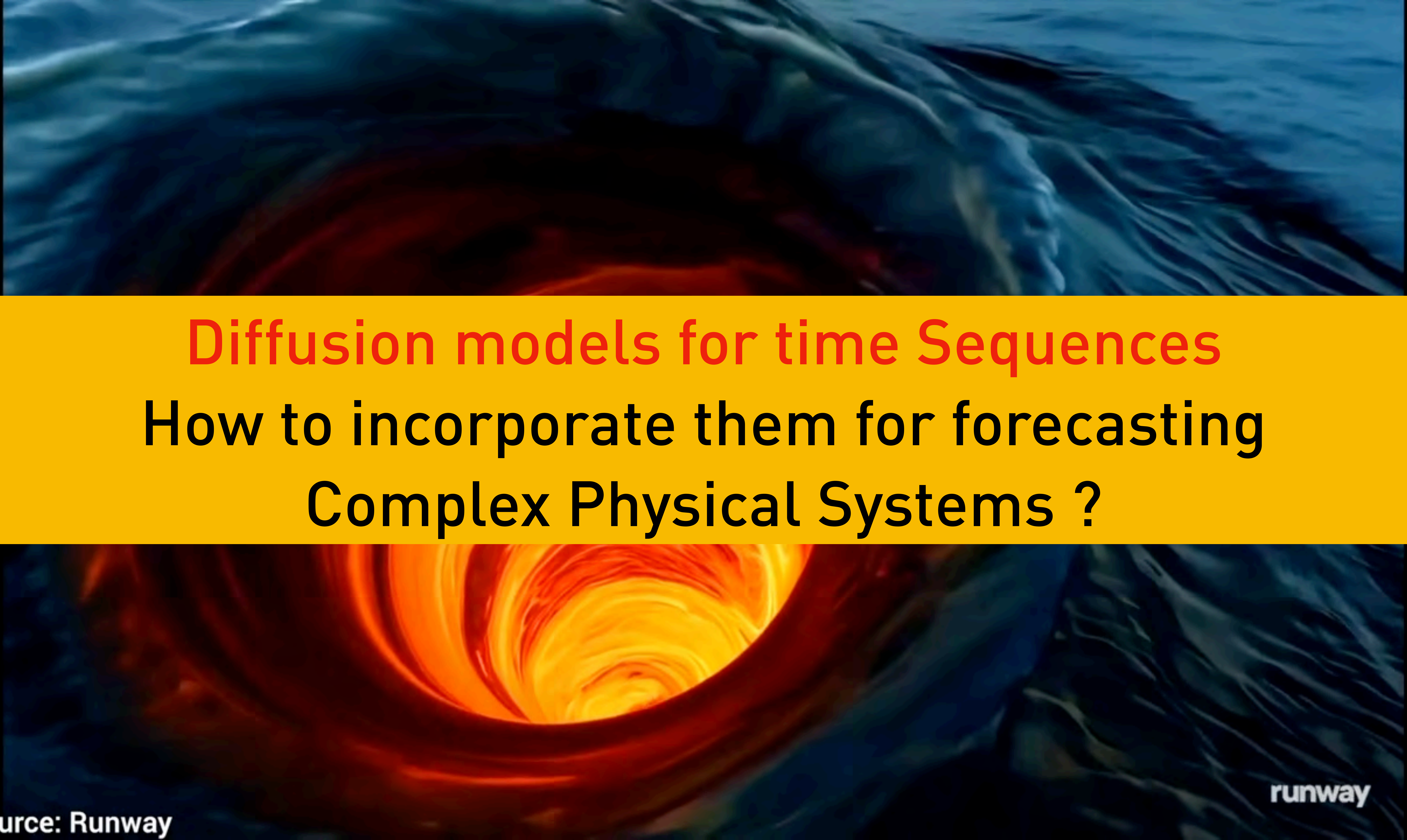
GUIDANCE THROUGH PHYSICS

Generate data from a conditional distribution $p(x \mid z)$ through **conditioning information** z.

Latent Dynamics as Guidance for Learning Effective Dynamics

Diffusion models for time Sequences

How to incorporate them for forecasting Complex Physical Systems ?



Source: Runway

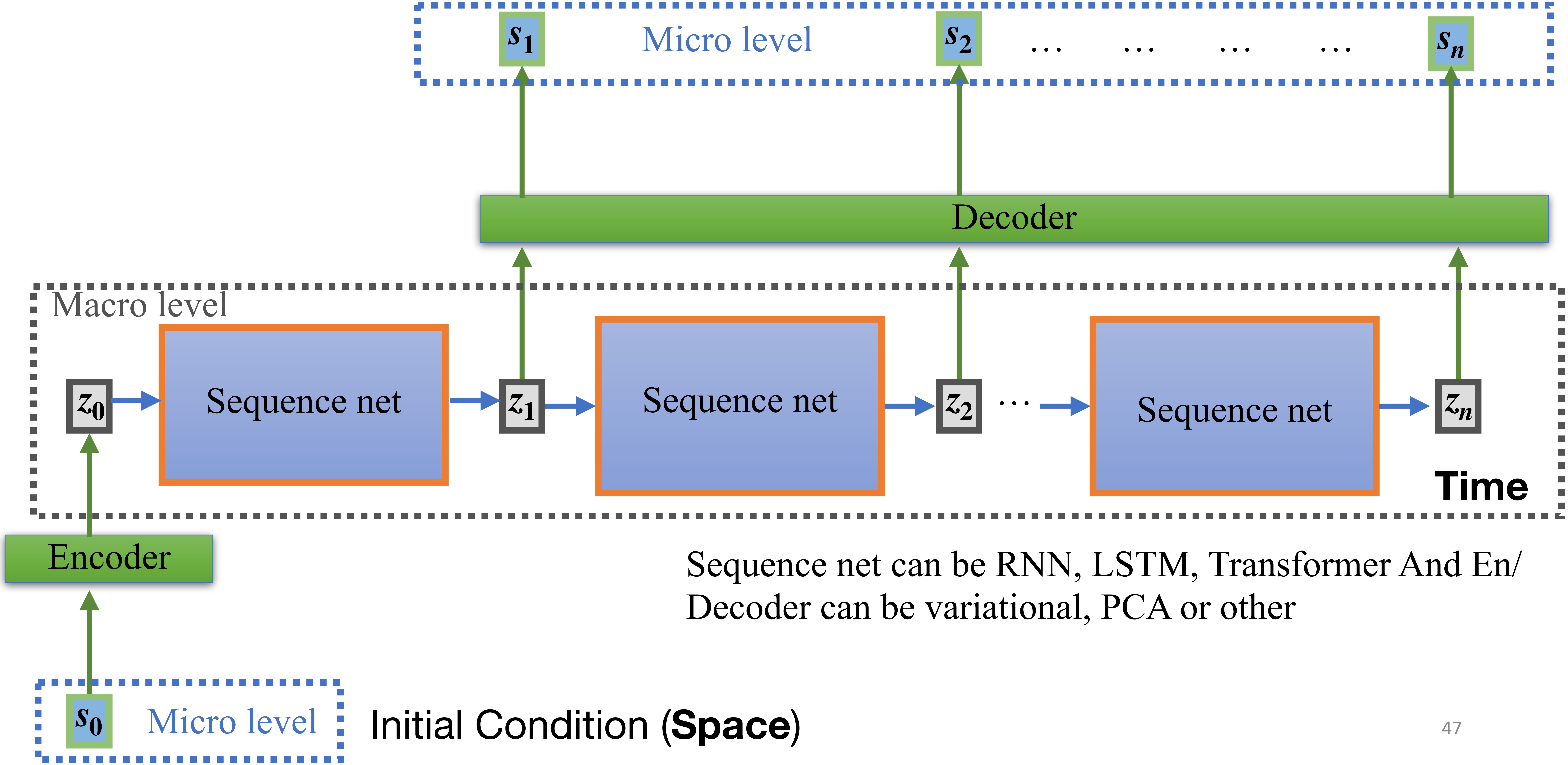
Generative Learning of Effective Dynamics (G-LED)

- Instances of high dimensional data are down sampled to a lower dimensional manifold that is evolved through an auto-regressive attention mechanism.

- In turn, Bayesian diffusion models, that map this low-dimensional manifold onto its corresponding high-dimensional space, capture the statistics of the system dynamics.

Forecasting

Spatiotemporal systems



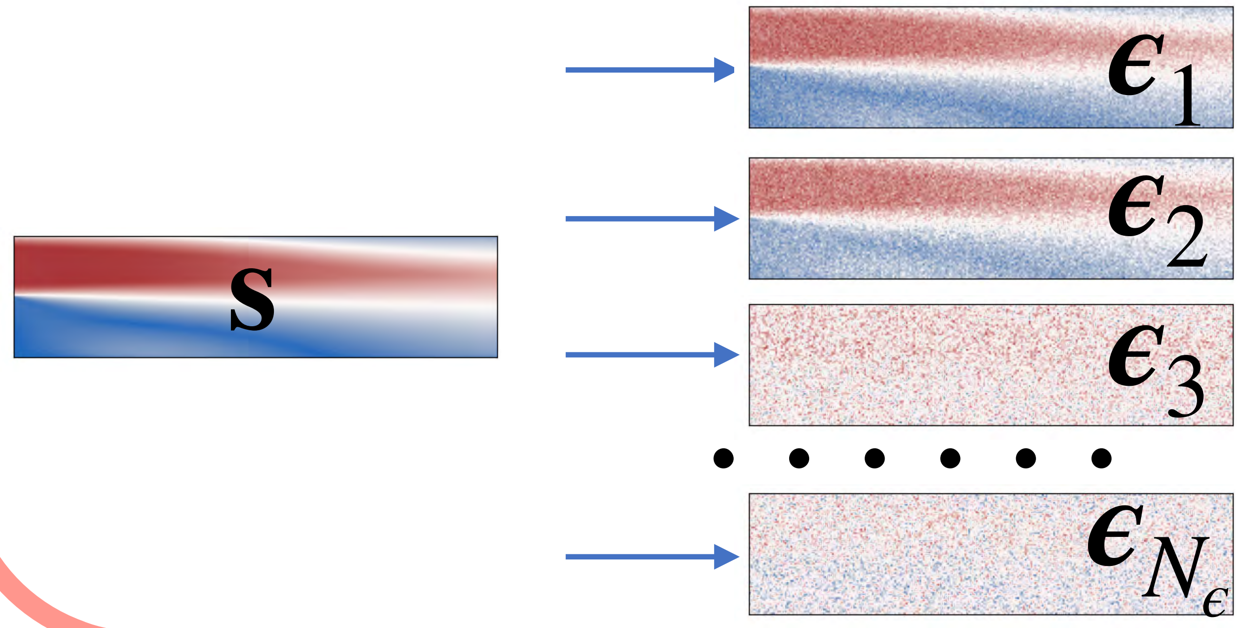
Sequence net can be RNN, LSTM, Transformer And En/ Decoder can be variational, PCA or other

Initial Condition (**Space**)


FORECASTING
Diffusion Models
s_1
Micro level
v_2
v_n
$x_0 = (v_1, v_2, \ldots, v_n)$
$z = (z_1, z_2, \ldots, z_n)$
Physics Information
$p_w(x_0 | x_1; z)$
$p_w(x_{t-1} | x_t; z)$
$p_w(x_t | x_{t+1}; z)$
$p_w(x_{T-1} | x_T; z)$
Guided diffusion model
z_0
TRANSFORMER
z_1
TRANSFORMER
z_2
TRANSFORMER
z_n
Macro level
Downsampler
v_0
Micro level

How to train the Diffusion model ?

(Forward process: adding noise to the training data)

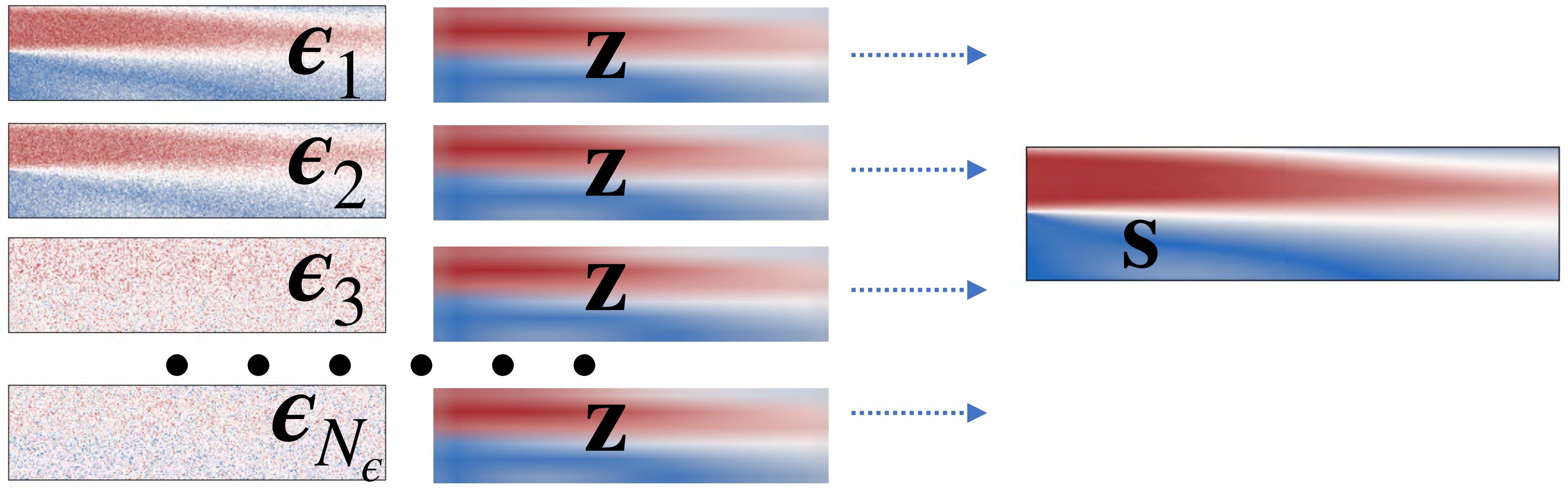


$$\boldsymbol{\epsilon}_i \sim q_i(\boldsymbol{\epsilon}_i \,|\, \mathbf{s}) := \mathcal{N}(\mathbf{s}, \sigma_i^2 \mathbf{I})$$

And σ_{N_ϵ} is larget enough $\mathrm{s.t.}\ \boldsymbol{\epsilon}_{N_\epsilon} \sim \mathcal{N}(\mathbf{s}, \sigma_{N_\epsilon}\mathbf{I}) \approx \mathcal{N}(0, \sigma_{N_\epsilon}\mathbf{I})$

$$\rho_{\text{inv}} := \frac{1}{\rho}, \quad \sigma_i = \left(\sigma_{\text{max}}^{\rho_{\text{inv}}} + \frac{N_\epsilon - i}{N_\epsilon - 1} \left(\sigma_{\text{min}}^{\rho_{\text{inv}}} - \sigma_{\text{max}}^{\rho_{\text{inv}}} \right) \right)^{\rho} \quad \text{for } i = 1, 2, \ldots, N_\epsilon,$$

(Train DNN to remove noise added to the training data)



The training is supervised, $\mathrm{DNN}_\theta : (\boldsymbol{\epsilon}_i, \mathbf{z}, i) \mapsto \mathbf{s}$,

A DNN is trained to denoise $\mathbf{s} \approx \mathrm{DNN}_\theta(\boldsymbol{\epsilon}_i, \mathbf{z}, i)$,

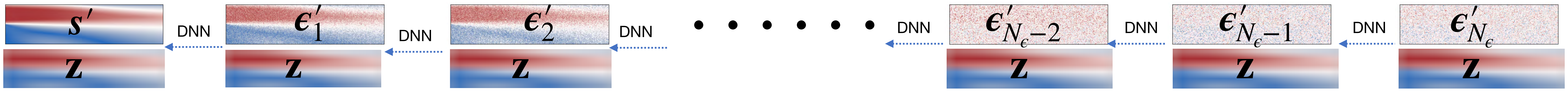
DNN is a 3-D UNet, and $\mathbf{z}$ the latent state

Decoding using the Diffusion Model:

(Reverse process)

Firstly, sample a white noise as a starting point: $\boldsymbol{\epsilon}'_{N_\epsilon} \sim \mathcal{N}(0, \sigma_{N_\epsilon}\mathbf{I})$

Then, from N_ϵ to 1, apply $\boldsymbol{\epsilon}'_i \sim p(\boldsymbol{\epsilon}'_i \,|\, \boldsymbol{\epsilon}'_{i+1}, \mathbf{z}, i) := \mathcal{N}\left(\frac{\sigma^2_{i+1} - \sigma^2_i}{\sigma^2_{i+1}} \underbrace{\mathrm{DNN}(\boldsymbol{\epsilon}'_{i+1}, \mathbf{z}, i)}_{\text{denoised } \mathbf{s}' \text{ given } \mathbf{z}} + \frac{\sigma^2_i}{\sigma^2_{i+1}} \boldsymbol{\epsilon}'_{i+1}, \frac{(\sigma^2_{i+1} - \sigma^2_i)\sigma^2_i}{\sigma^2_{i+1}} \mathbf{I}\right)$



During predictions, $\boldsymbol{\epsilon}'_{N_\epsilon}$ is a new white noise and the sampling is stochastically achieved as

$$\boldsymbol{\epsilon}'_i \sim p(\boldsymbol{\epsilon}'_i \,|\, \boldsymbol{\epsilon}'_{i+1}, \mathbf{z}, i) := \mathcal{N}\left(\frac{\sigma^2_{i+1} - \sigma^2_i}{\sigma^2_{i+1}} \underbrace{\mathrm{DNN}(\boldsymbol{\epsilon}'_{i+1}, \mathbf{z}, i)}_{\text{denoised } \mathbf{s}' \text{ given } \mathbf{z}} + \frac{\sigma^2_i}{\sigma^2_{i+1}} \boldsymbol{\epsilon}'_{i+1}, \frac{(\sigma^2_{i+1} - \sigma^2_i)\sigma^2_i}{\sigma^2_{i+1}} \mathbf{I}\right)$$

Since the reverse diffusion process is trained on $\mathbf{z}$, the decoding process has a low variance compared to diffusion models in computer vision.

What about known physical constraints (or partial information) ?

Incorporate as (virtual) observables via Gradient Guidance !

* S.Kaltenbach and P.-S. Koutsourelakis: Incorporating physical constraints in a deep probabilistic machine learning framework for coarse-graining dynamical systems, J. Comp. Physics, 2020

1: Formulate physical information as a residual $\boldsymbol{R}(\boldsymbol{s}_t)$.
In case we have some equation with governing dynamics $\hat{\boldsymbol{R}}(\boldsymbol{s}_t) = 0$ then we can also formulated a residual such as virtually observed with $\hat{\boldsymbol{R}}(\boldsymbol{s}_t) = 0$ *

2: Use Bayes Law to condition the current state of the diffusion process on the residual. An estimate for $\boldsymbol{s}_t$ is obtained based on z_t using the trained NN.

3: Use the modified gradient in the reverse diffusion process

$$p(\boldsymbol{\epsilon}_i | \hat{\boldsymbol{R}} = 0, \mathbf{z}) = \frac{p(\hat{\boldsymbol{R}} = 0 | \mathbf{z}, \boldsymbol{\epsilon}_i) p(\boldsymbol{\epsilon}_i | \mathbf{z})}{C}$$

$$= \nabla_\epsilon \log p(\hat{\boldsymbol{R}} = 0 | \boldsymbol{\epsilon}_i, \mathbf{z}) + \nabla_\epsilon \log p(\boldsymbol{\epsilon}_i | \mathbf{z})$$

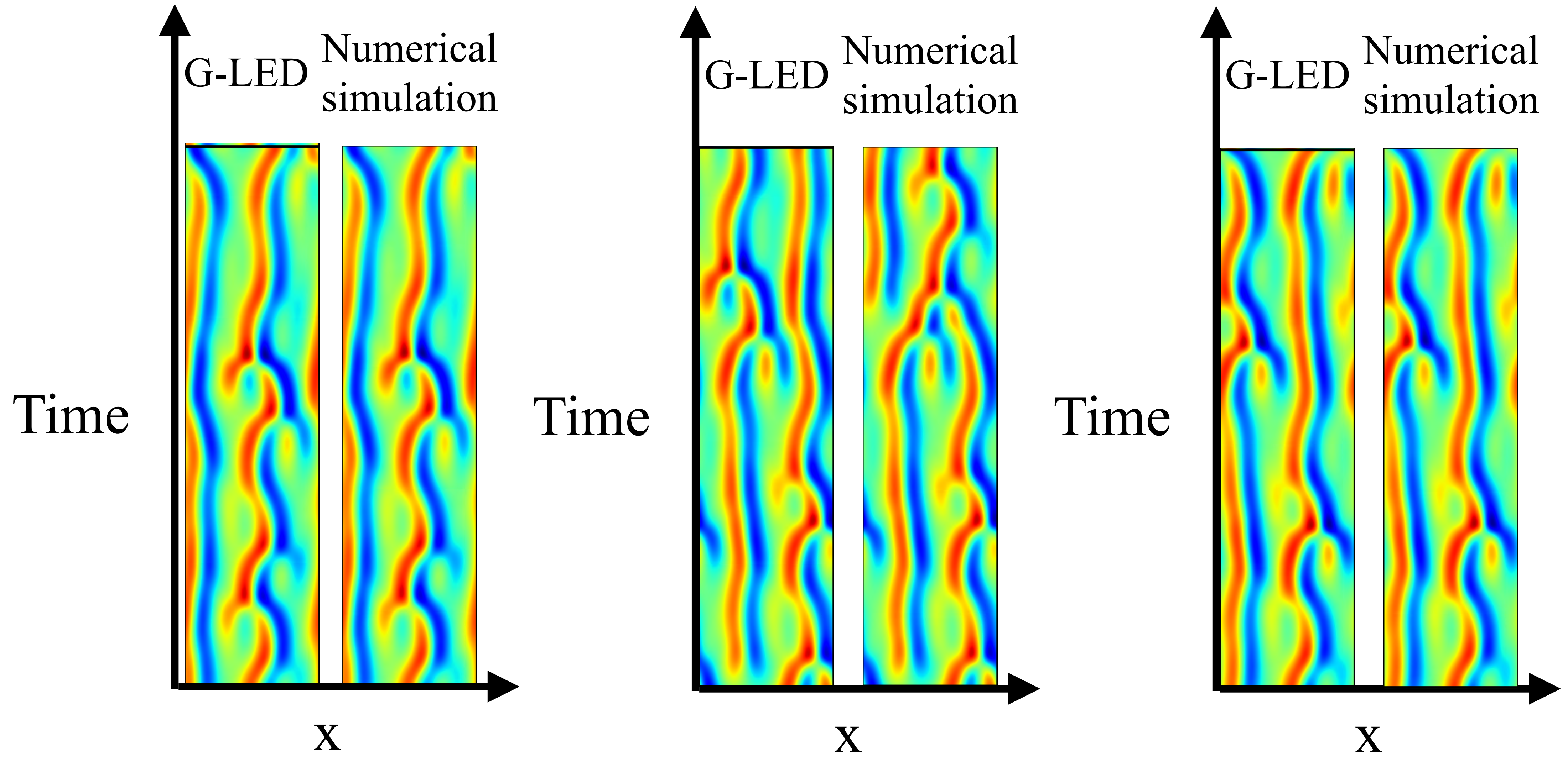
New term due to physical information


Micro level
s_1
s_2
s_n
Noise
Noise
Noise
Physics is incorporated as virtual observables via gradient guidance
Spatiotemporal Bayesian diffusion model
Macro level
z_0
Transformer
z_1
Transformer
z_2
Transformer
z_n
Forecasting in time
Downsampler
Spatiotemporal systems
s_0
Micro level
Initial Condition (space)

G-LED Results: 1-D Kuramoto–Sivashinsky equation

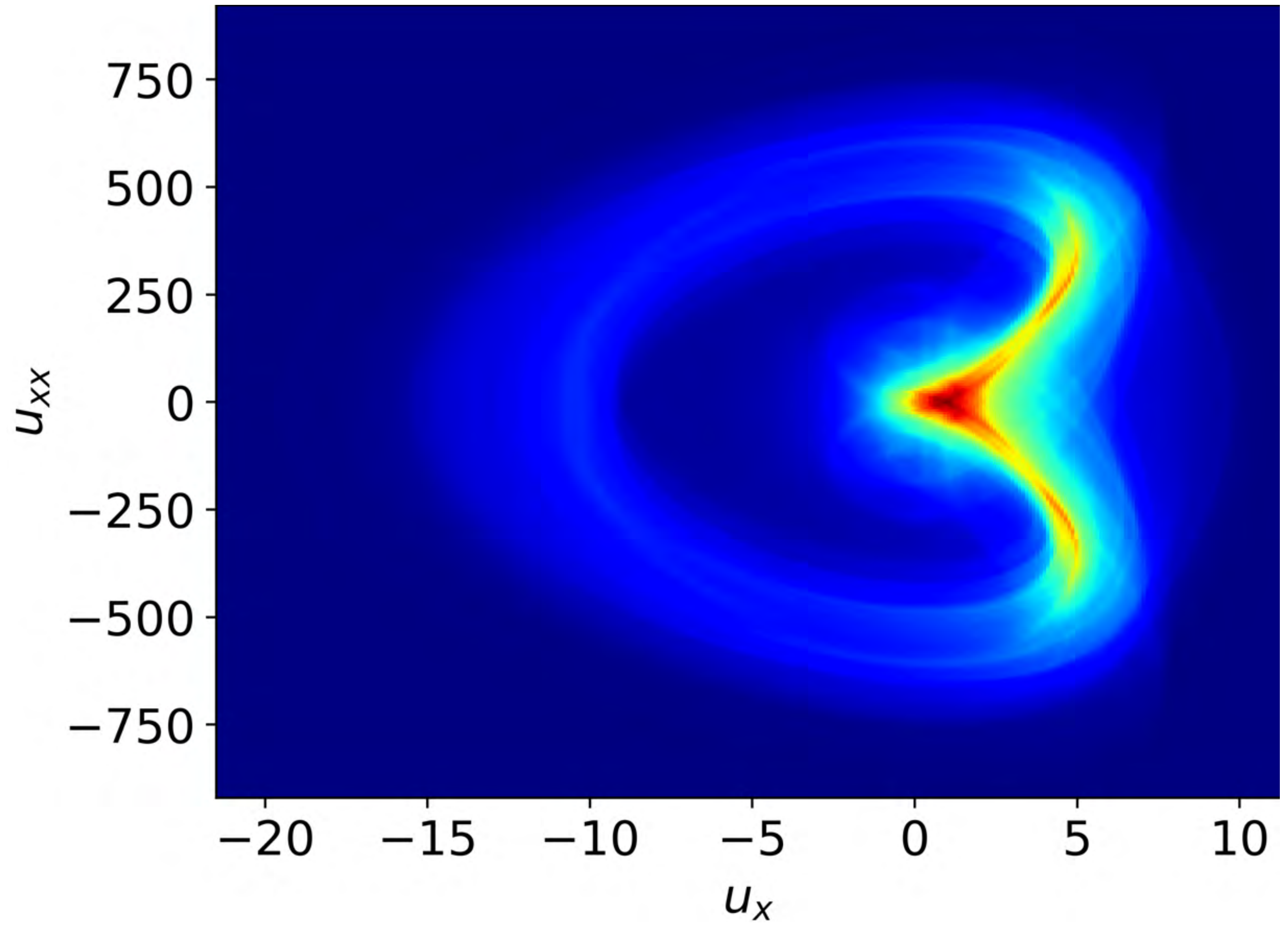
$$\frac{\partial u}{\partial t} = -\nu \frac{\partial^4 u}{\partial t^4} - \frac{\partial^2 u}{\partial x^2} - u\frac{\partial u}{\partial x},$$

$$\Omega = [0,22] \quad \text{with} \quad u(0,t) = u(L,t) \quad \text{and} \quad \nu = 1$$

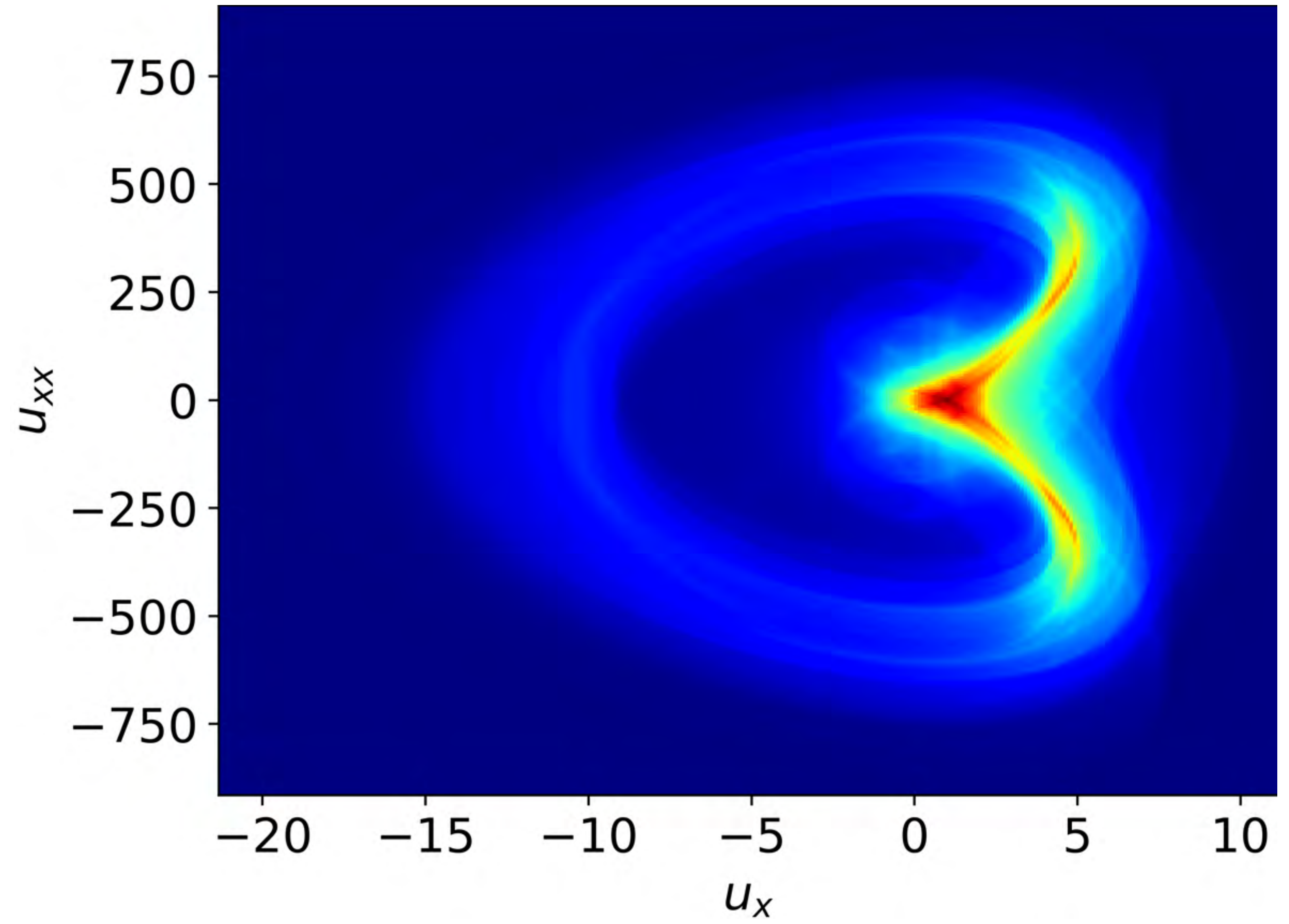


test trajectories with new initial condition.
The vertical direction depicts the time t from 0 to 96s where the first 16s used an initial conditions for warm-up. 21K trajectories are used for training and 19K for testing.

G-LED: Manifold

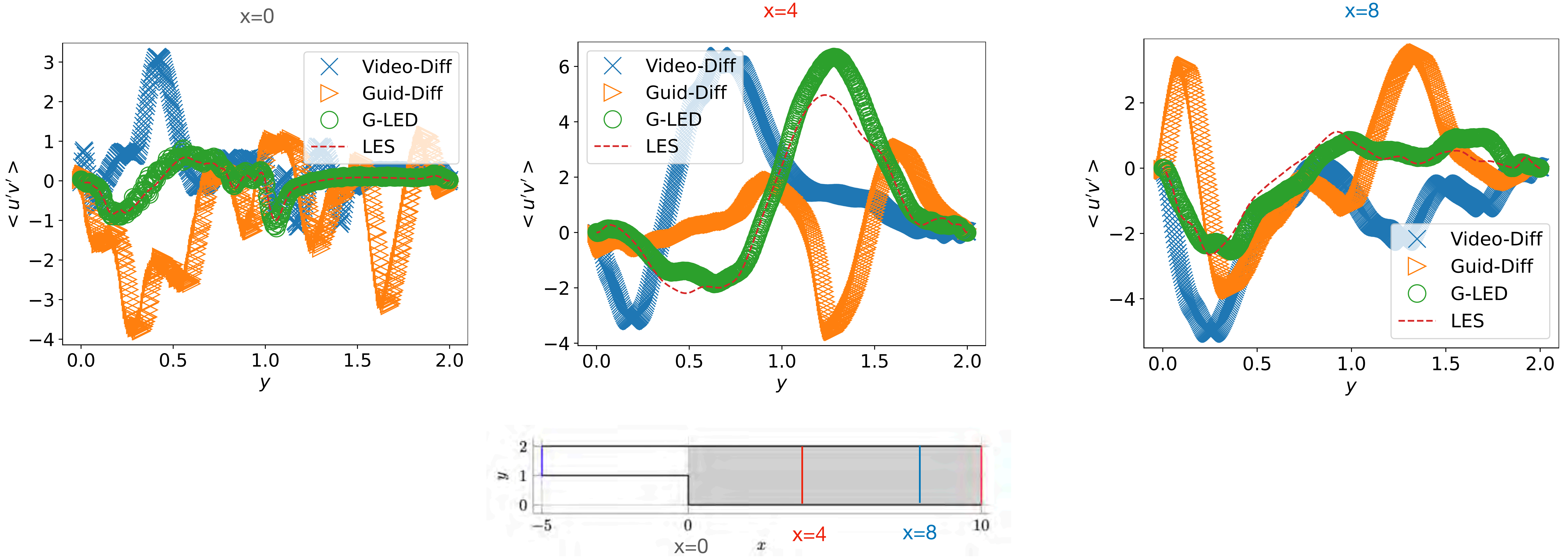


Numerical simulation: Manifold



G-LED Results: Outperform different diffusion modeling methods

Mean stress of streamwise-wallnormal velocity



Video-Diff: Video diffusion models. Advances in Neural Information Processing Systems, 35, 8633-8646.

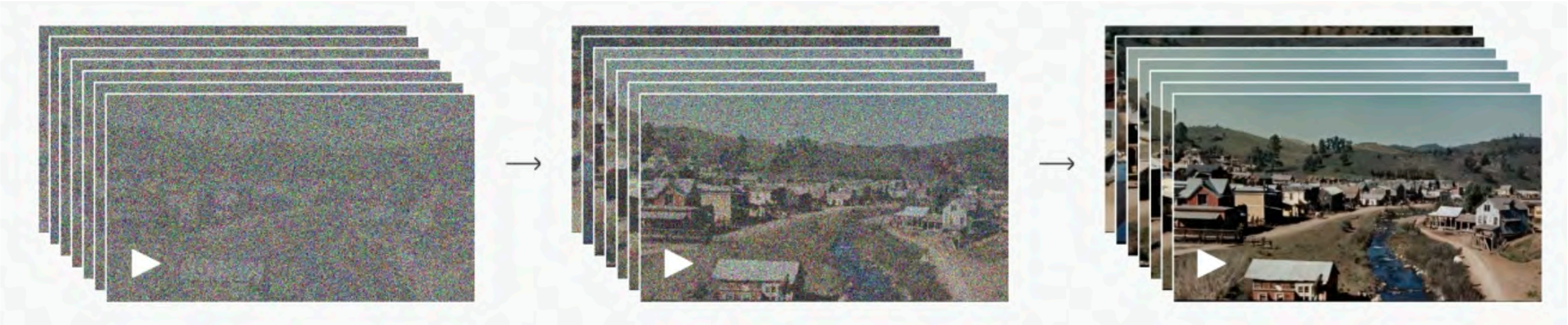
Guidance-Diff: Bayesian conditional diffusion models for versatile spatiotemporal turbulence generation. *Computer Methods in Applied Mechanics and Engineering*, *427*, 117023.

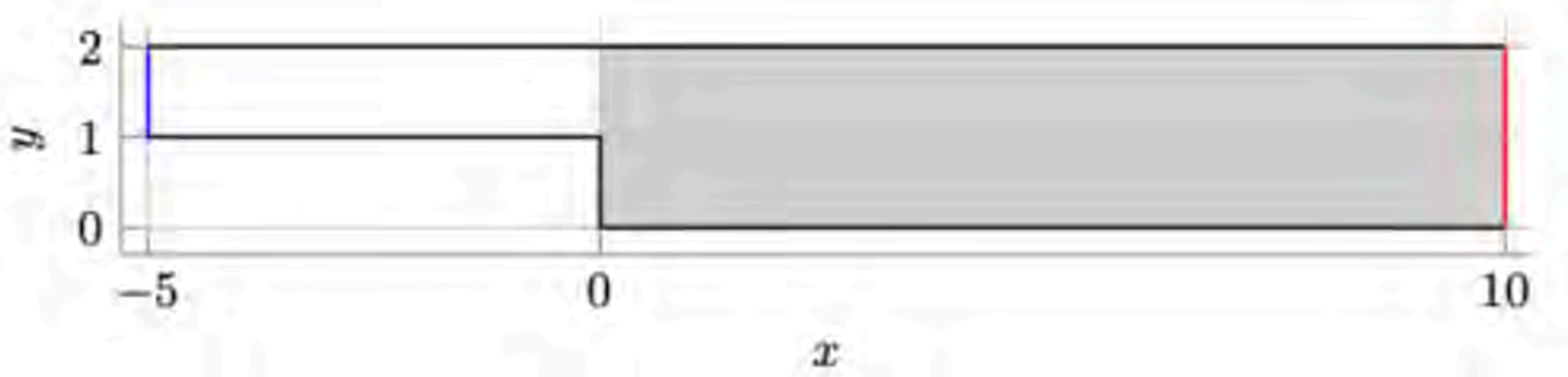
Forward and reverse processes in G-LED: Summary

KEY ISSUES :

1. Diffusion models are associated with large variations in the generated samples.
   - In G-LED **sequence of snapshots are correlated by the underlying physical process** via macro sequences.
   - **Condition the denoising process on the latent states**.

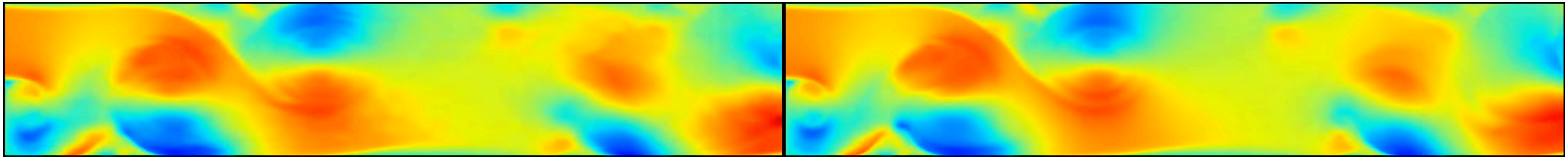
2. G-LED decodes multiple consecutive macro states together as a **batch** (similar to Sora) to **enhance temporal coherence and increase temporal smoothness** in the results.





Geometry of flow domain (solid lines), area of interest (shadowed zone)

Streamwise velocity from t=0s to 1.25s

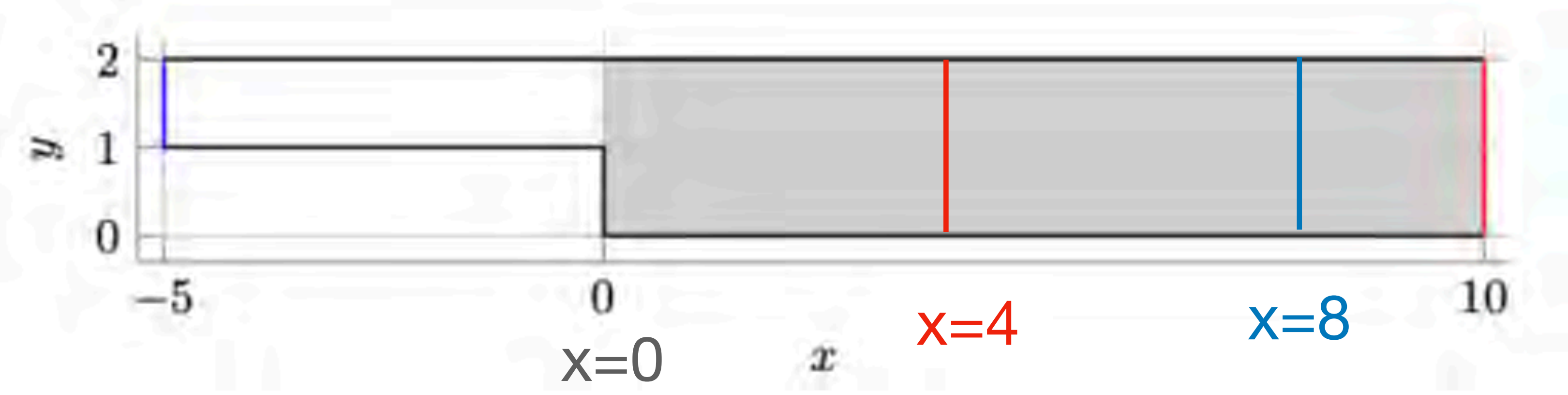


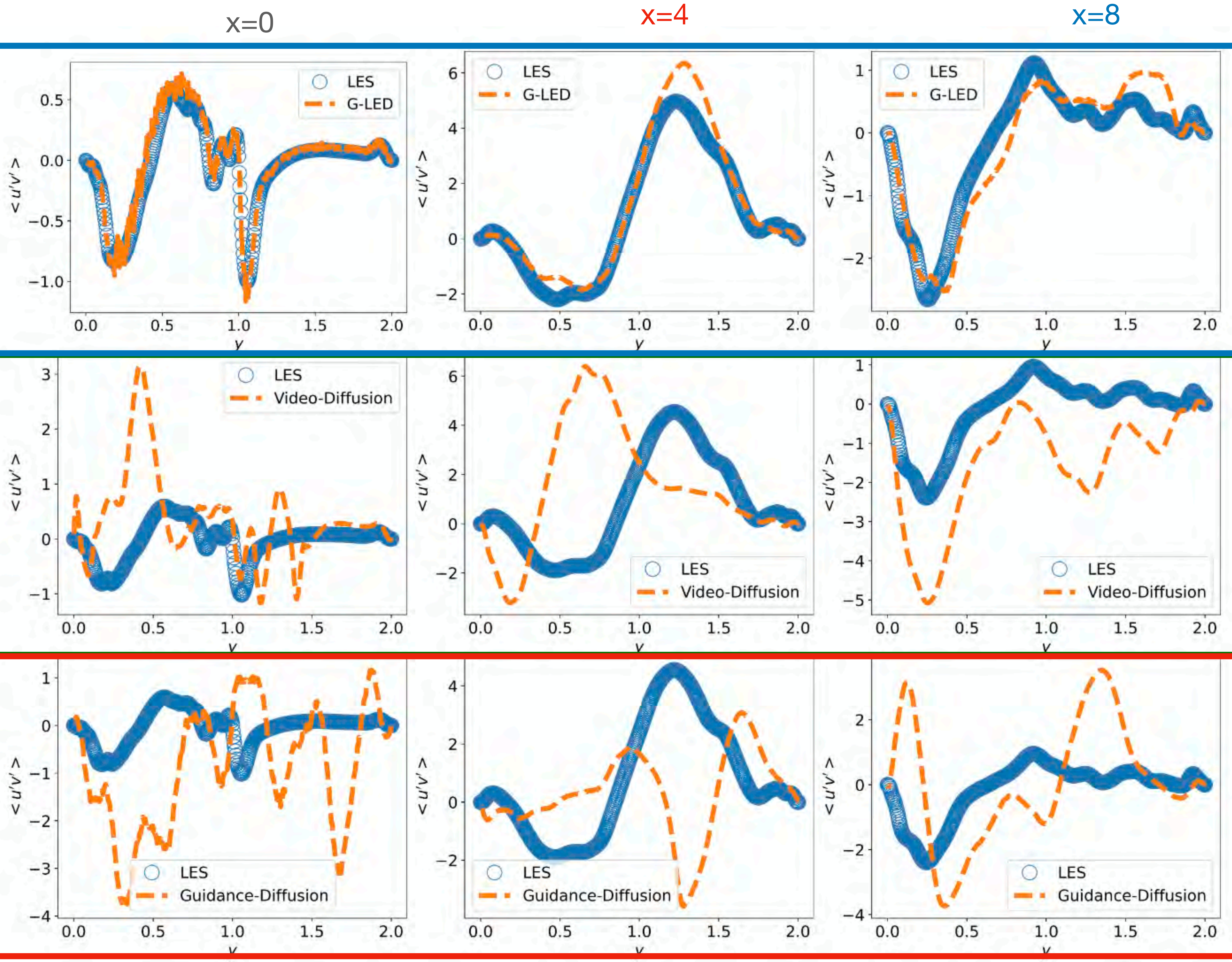
G-LED LES

Wallnormal velocity from t=0s to 1.25s

The spatial domain is discretized with a 512x512 uniform grid the snapshots are subsampled with a larger time step Dt = 0.05. 8000 snapshots are used for training, 500 snapshots are for validation and 1500 snapshots are for the testing.

Mean stress of streamwise-wallnormal velocity





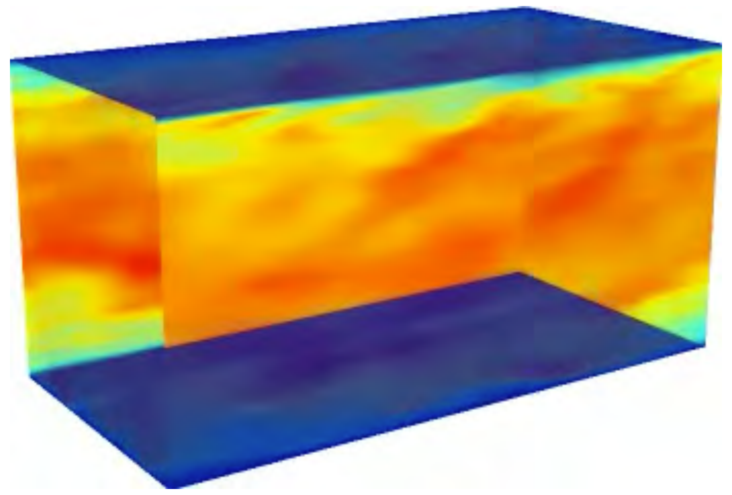
G-LED

Video-Diffusion: Video diffusion models. Advances in Neural Information Processing Systems, 35, 8633-8646.

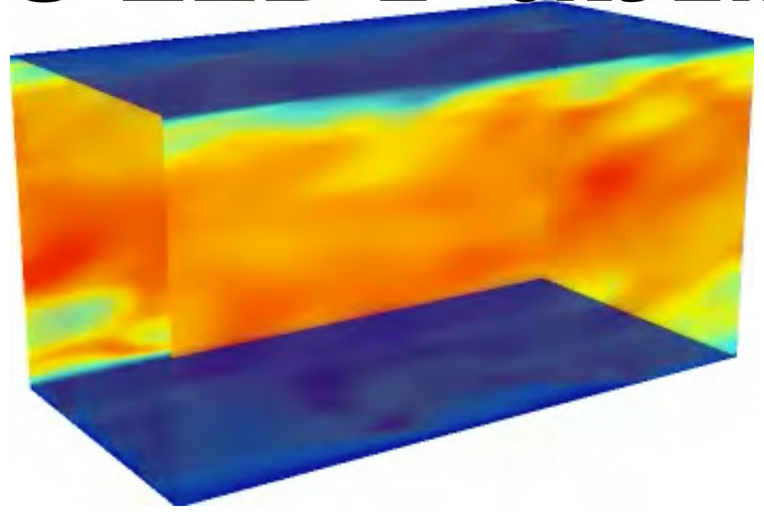
Guidance-Diffusion: Bayesian conditional diffusion models for versatile spatiotemporal turbulence generation. *Computer Methods in Applied Mechanics and Engineering*, *427*, 117023.

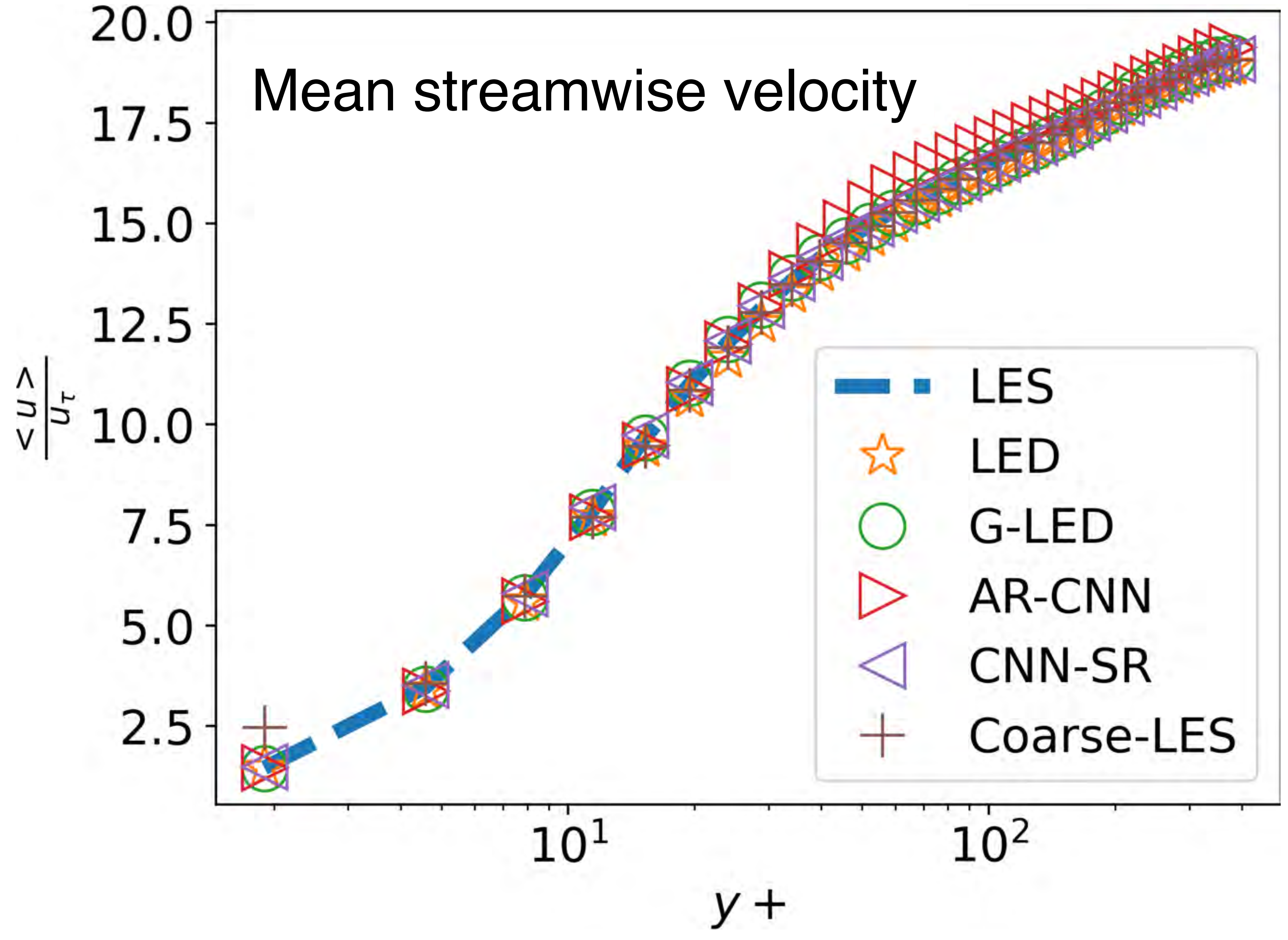
Turbulent channel flow $Re_\tau = 395$

LES = 40x50x30



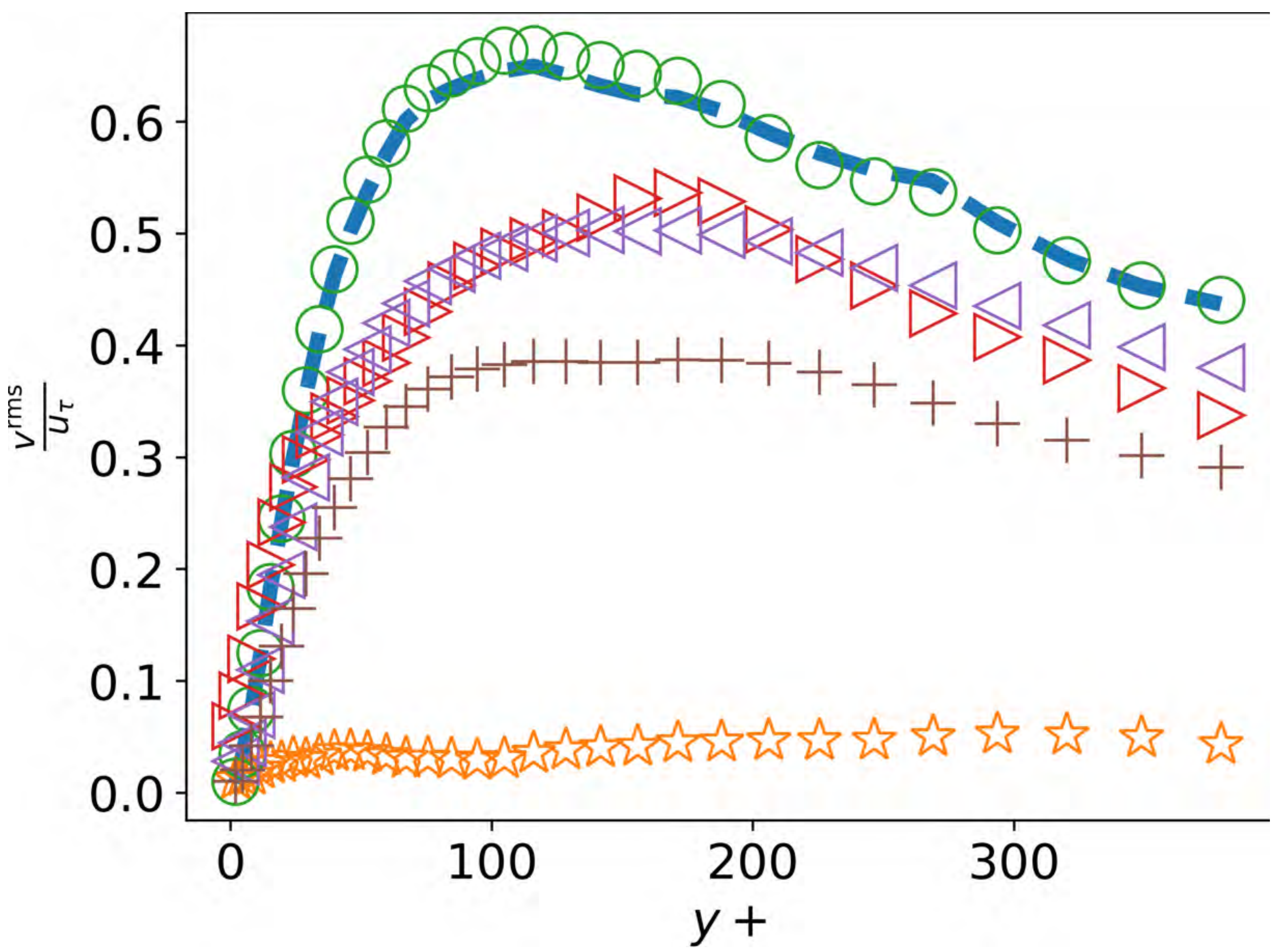
G-LED z=8x32x8



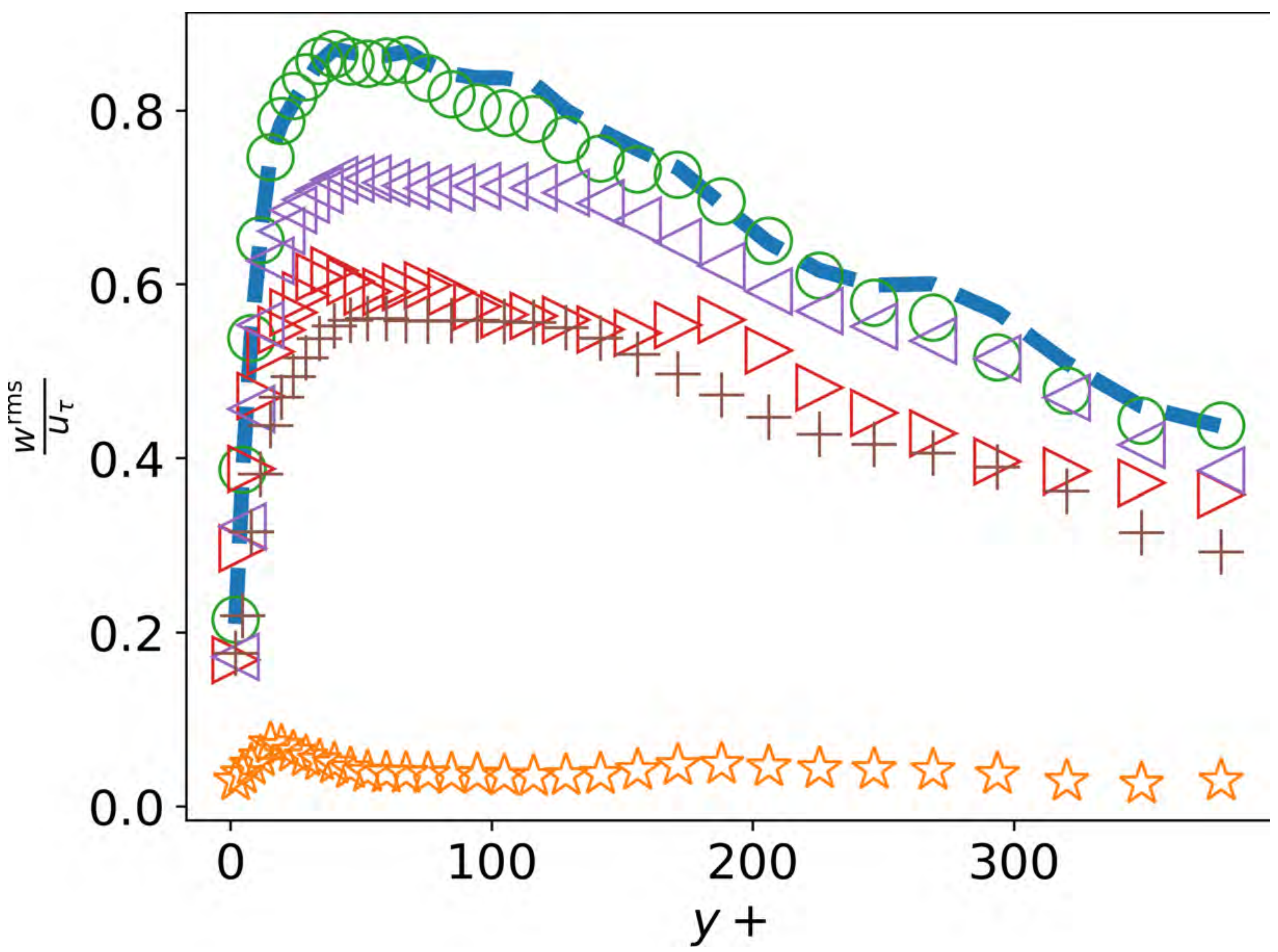


LED	Vlachas, P. R., Arampatzis, G., Uhler, C., & Koumoutsakos, P. (2022). Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence*, *4*(4), 359-366.
LES	Nicoud, F., & Ducros, F. (1999). Subgrid-scale stress modelling based on the square of the velocity gradient tensor. *Flow, turbulence and Combustion*, *62*(3), 183-200.
AR-CNN	Geneva, N., & Zabaras, N. (2020). Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, *403*, 109056.
CNN-SR	Ren, P., Rao, C., Liu, Y., Ma, Z., Wang, Q., Wang, J. X., & Sun, H. (2023). PhySR: Physics-informed deep super-resolution for spatiotemporal data. *Journal of Computational Physics*, *492*, 112438.

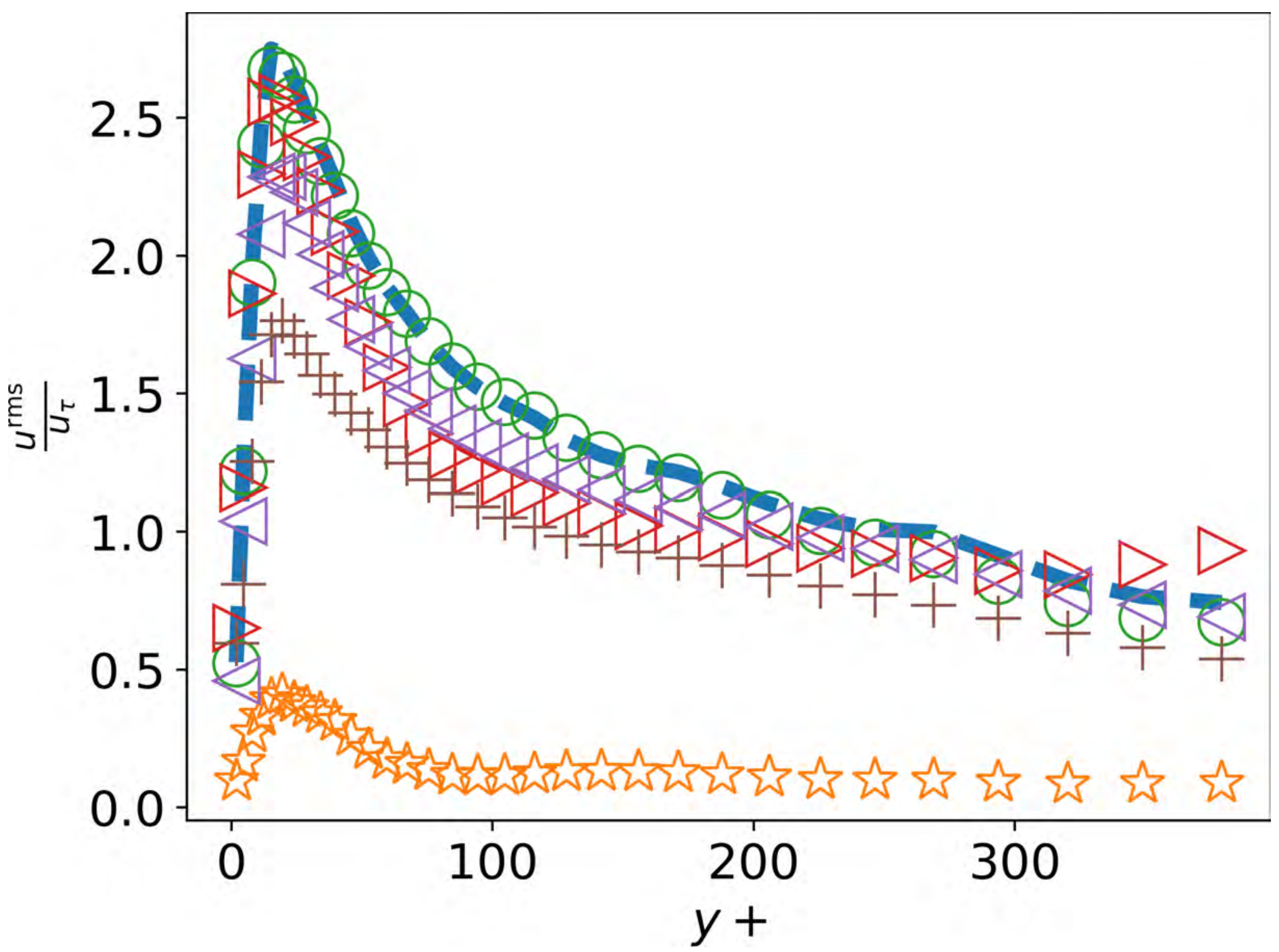
wall normal fluctuations,



spanwise fluctuations,



streamwise fluctuations,



SUMMARY of G-LED

- **A** (surprisingly powerful) **generative framework for forecasting complex systems and forecast their statistics.**
- In G-LED:
  - Bayesian diffusion model is trained on high dimensional simulations and integrates physical information in its prior knowledge.
  - A flexible attention model that evolves the latent space dynamics.
  - The generative model projects the latent space dynamics to high dimensional spaces.

LEARNING TO SOLVE PROBLEMS

ALGORITHMS

What is Intelligence ?


John McCarthy

Intelligence is
the computational part
of the ability to achieve
goals in the world.

A system having a goal or not, is not a property of the system itself. It is in the **relationship between the system and an observer.**

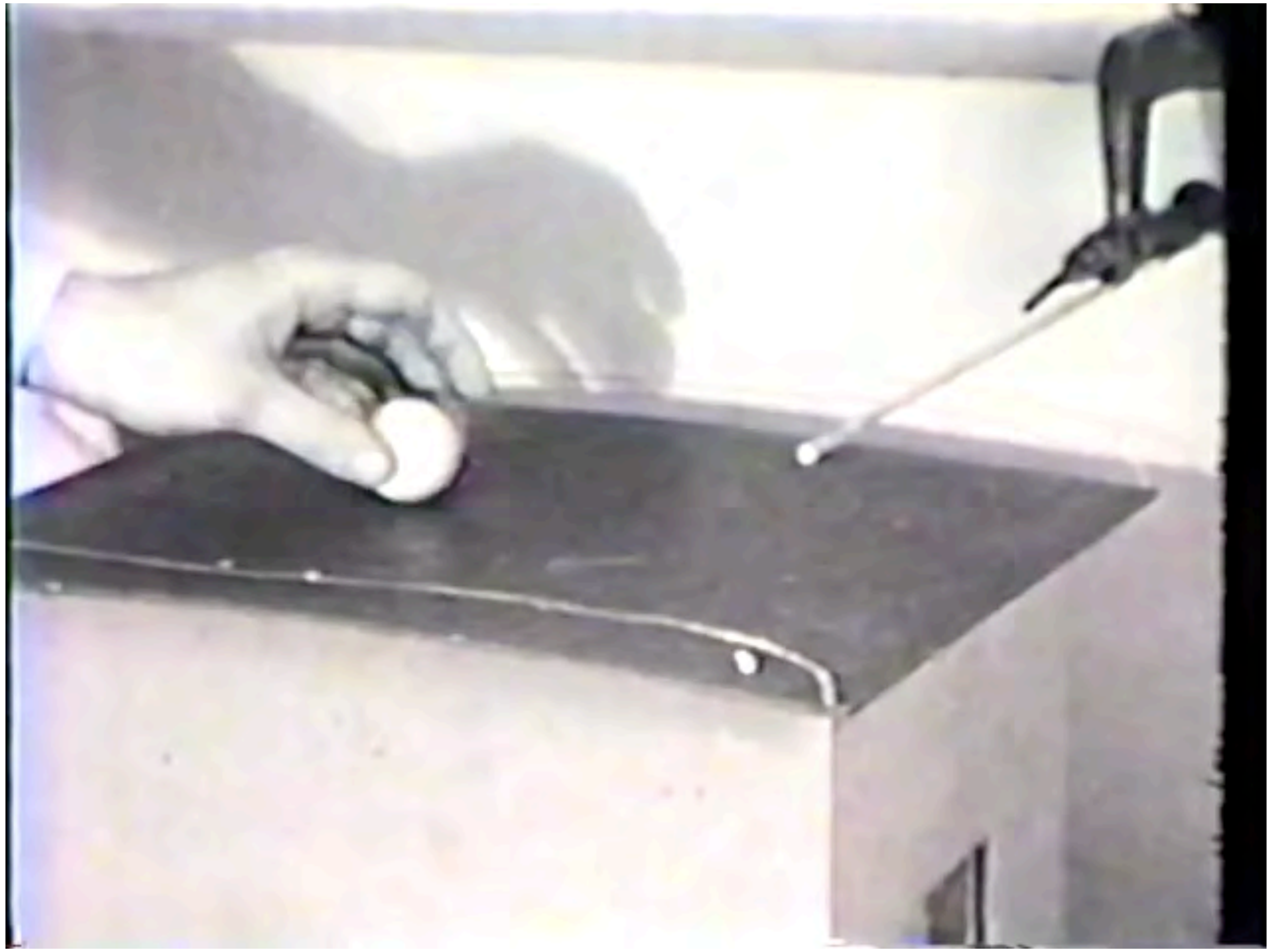
The system is most usefully understood/predicted/controlled in terms **of its outcomes rather than its mechanisms.**

http://jmc.stanford.edu/artificial-intelligence/what-is-ai/index.html

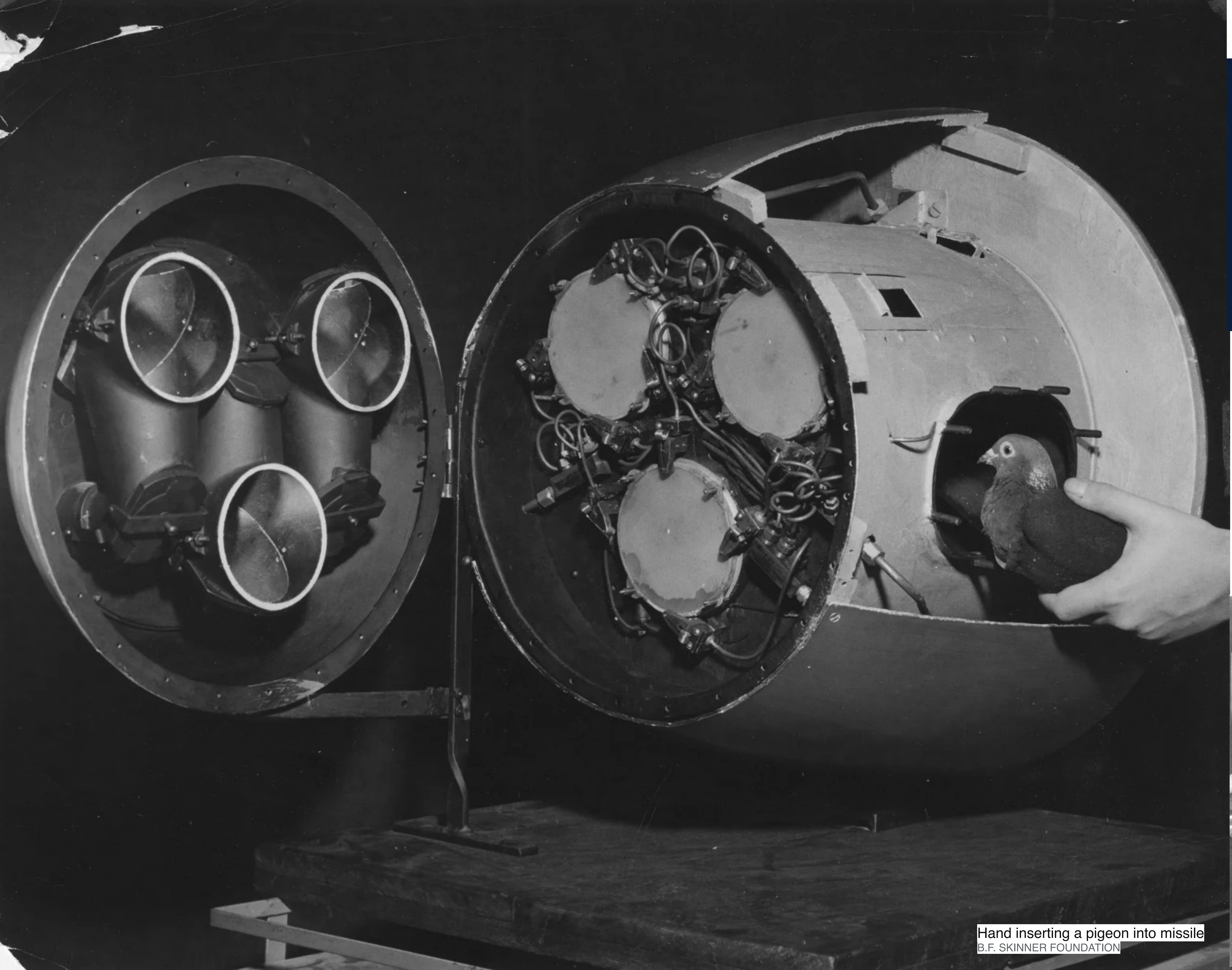
Reinforcement Learning

Learning: Behavioral changes due to Experiences (Action, Stimulus, Reward)

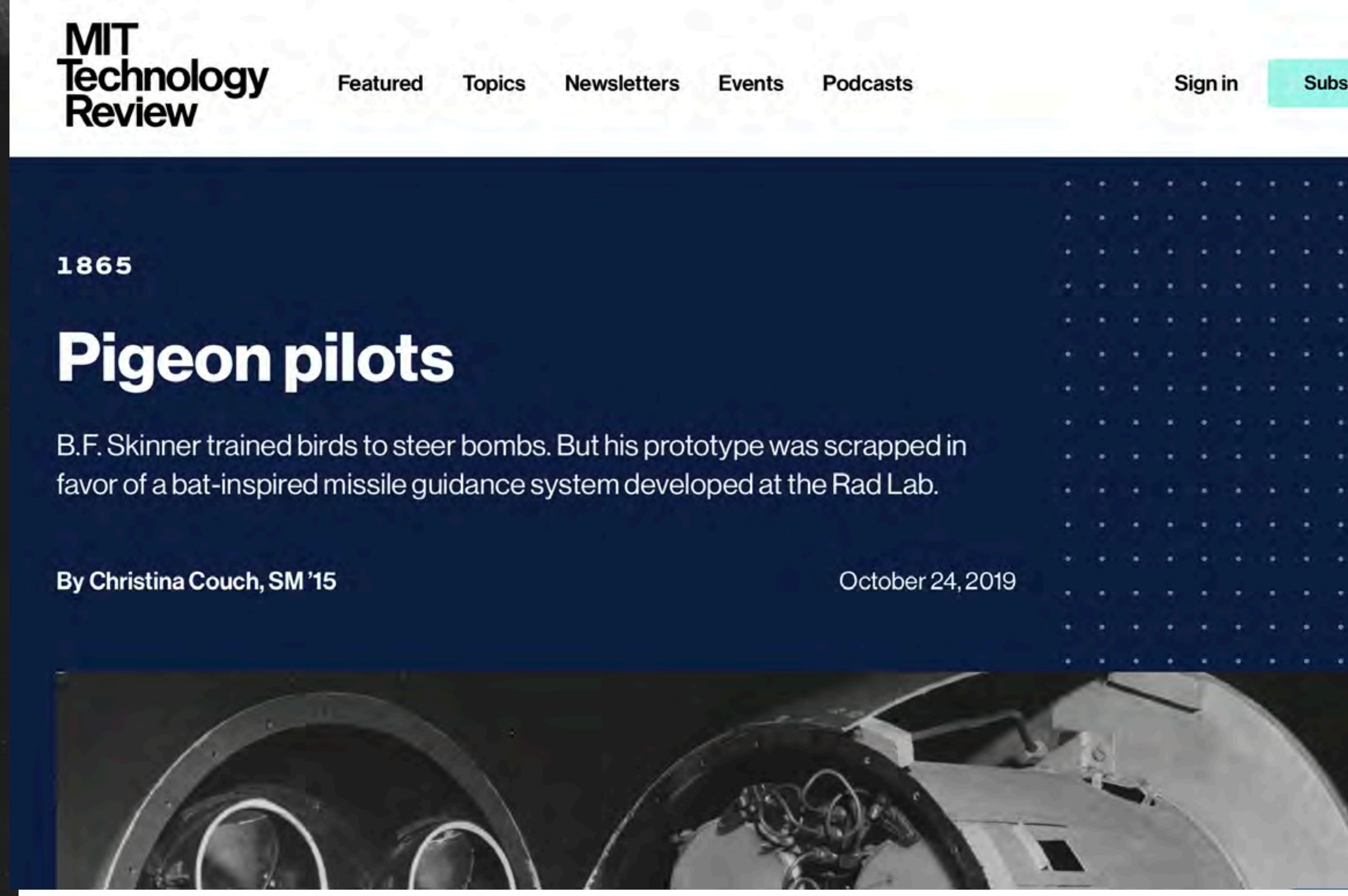
Reinforcement: **stimulus-action** pattern is **rewarded ->** actor is conditioned to a behavior.



CREDIT: B.F. Skinner Foundation



Hand inserting a pigeon into missile
B.F. SKINNER FOUNDATION



MIT Technology Review

Featured Topics Newsletters Events Podcasts Sign in Subs

1865

Pigeon pilots

B.F. Skinner trained birds to steer bombs. But his prototype was scrapped in favor of a bat-inspired missile guidance system developed at the Rad Lab.

By Christina Couch, SM '15 October 24, 2019

In the early 1940s, as part if the war effort, the Brelands assisted Skinner in his famous Project Pigeon, in which they taught pigeons how to guide bombs. They did this work atop a General Mills grain elevator in Minneapolis, pictured below. Both Keller and Marian left the University of Minnesota without doctorates, planning to apply the powerful procedures they had learned under Skinner to animal behavior. In 1961, when they published their most famous work, they playfully entitled it, The Misbehavior of Organisms.

https://www3.uca.edu/iqzoo/History/bf_skinner.htm

$$J(w) = \mathop{\mathbb{E}}_{\substack{a_t \sim \pi_w(a \mid s_t) \\ s_{t+1} \sim \mathcal{D}(s \mid a_t, s_t)}} \left[\sum_t r_t \right]$$

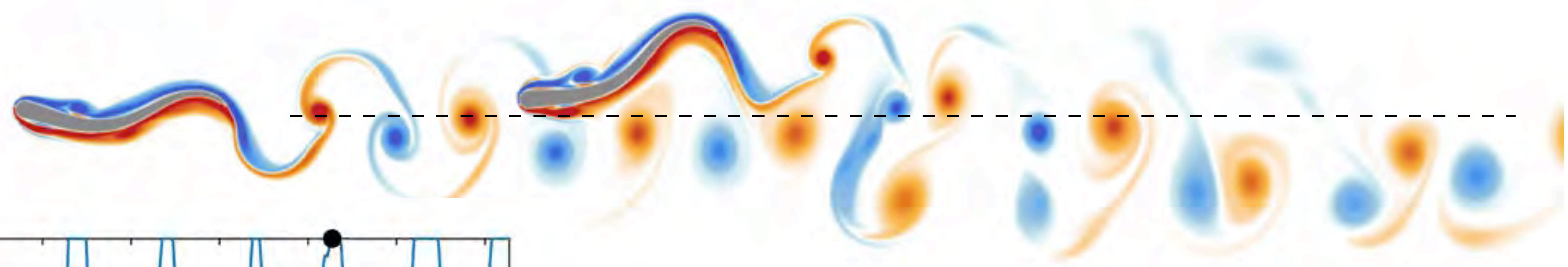
A World Model, from Scott McCloud's *Understanding Comics*. [1, 2]

GOAL II : maximising efficiency

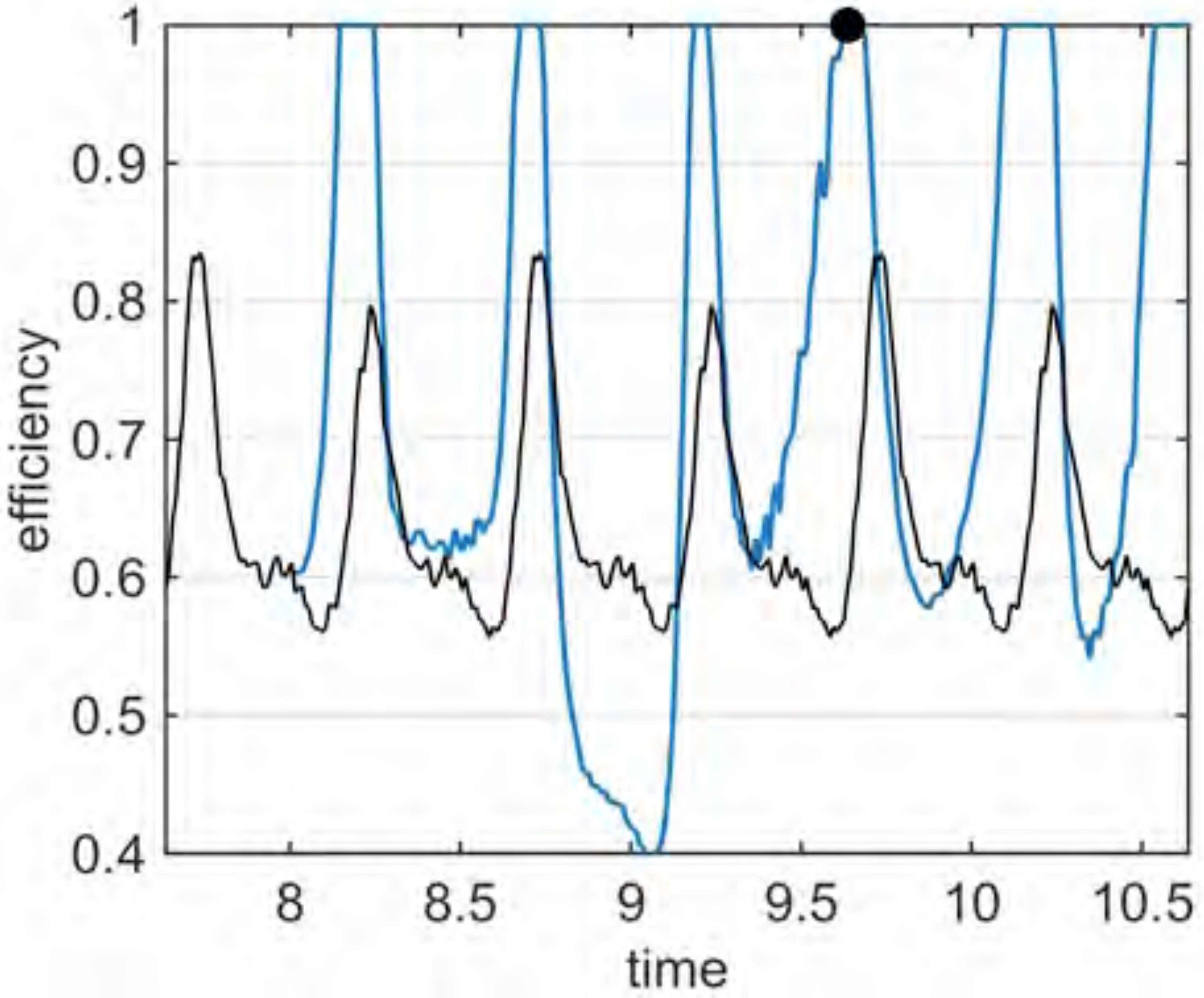
No distance-based constraints specified

EARLY STAGES OF LEARNING

GOAL II : MAX EFFICIENCY



No distance constraints specified

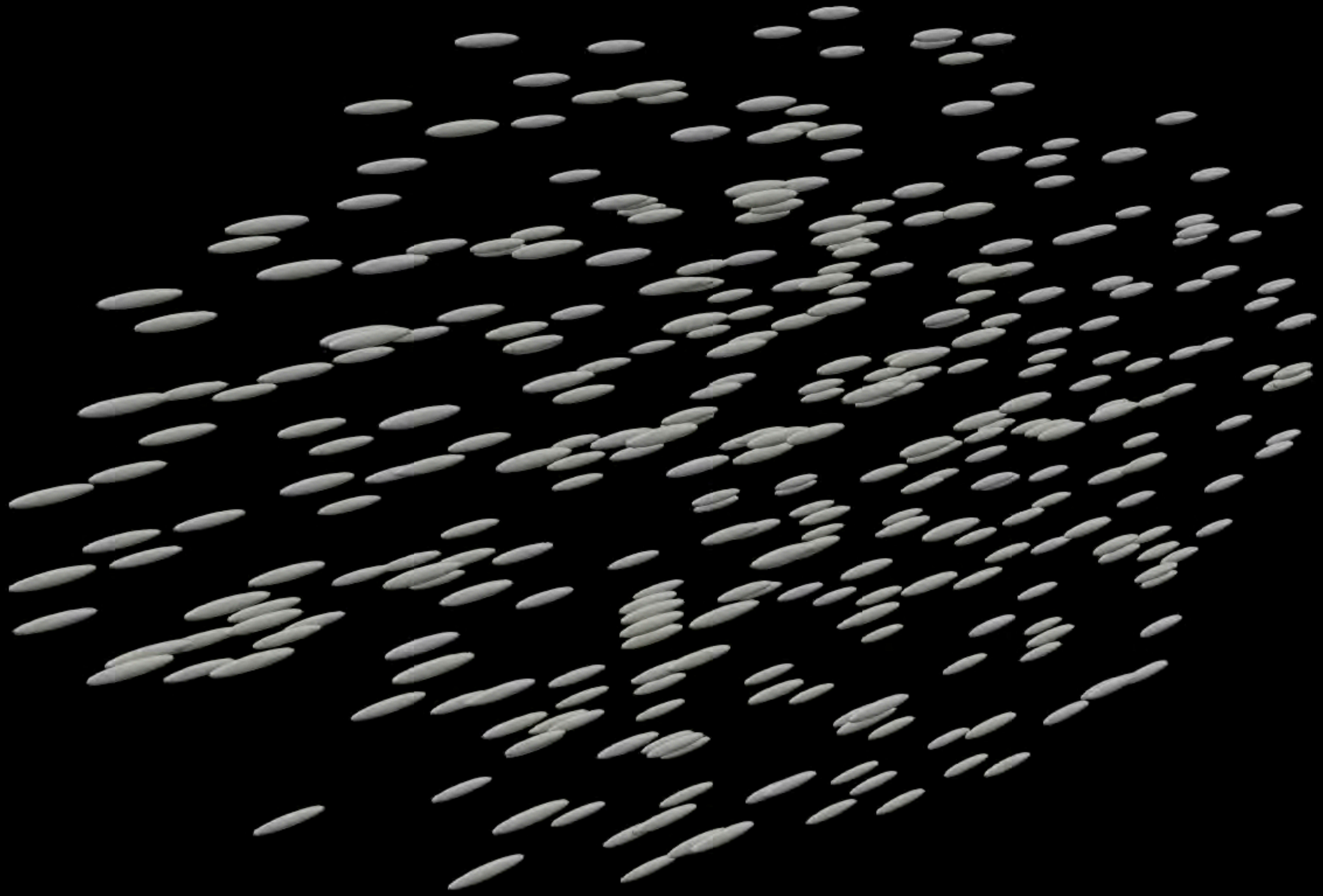


$$R_\eta = \frac{Tu}{Tu + max(P_{def}, 0)}$$

- Follower ***opts*** to interact with wake-vortices
  - *Overall, 28% gain in average efficiency*

CONTROL

S. Verma, G. Novati, and P. Koumoutsakos, “Efficient collective swimming by harnessing vortices through deep reinforcement learning," *P. Natl. Acad. Sci.*, p. 201800923, 2018.



ABF catching a circulating cancerous cell

Reinforcement Learning for Flow **Control/Modeling**

$$F(\mathbf{x}, t) = 0$$

Governing Equation

$$\hat{F}(\mathbf{x}, t) + \pi(\mathbf{s}(\mathbf{x}), \mathbf{a}(\mathbf{s})) = 0$$

Model
Control

RL: find a policy $\pi(\mathbf{s}, \mathbf{a})$ for the actions of an agent that learns to optimize their long-term consequences on the environment.

$$u_{ijk}^{n+1} = F(u_{ijk}^{n})$$

F : from numerics

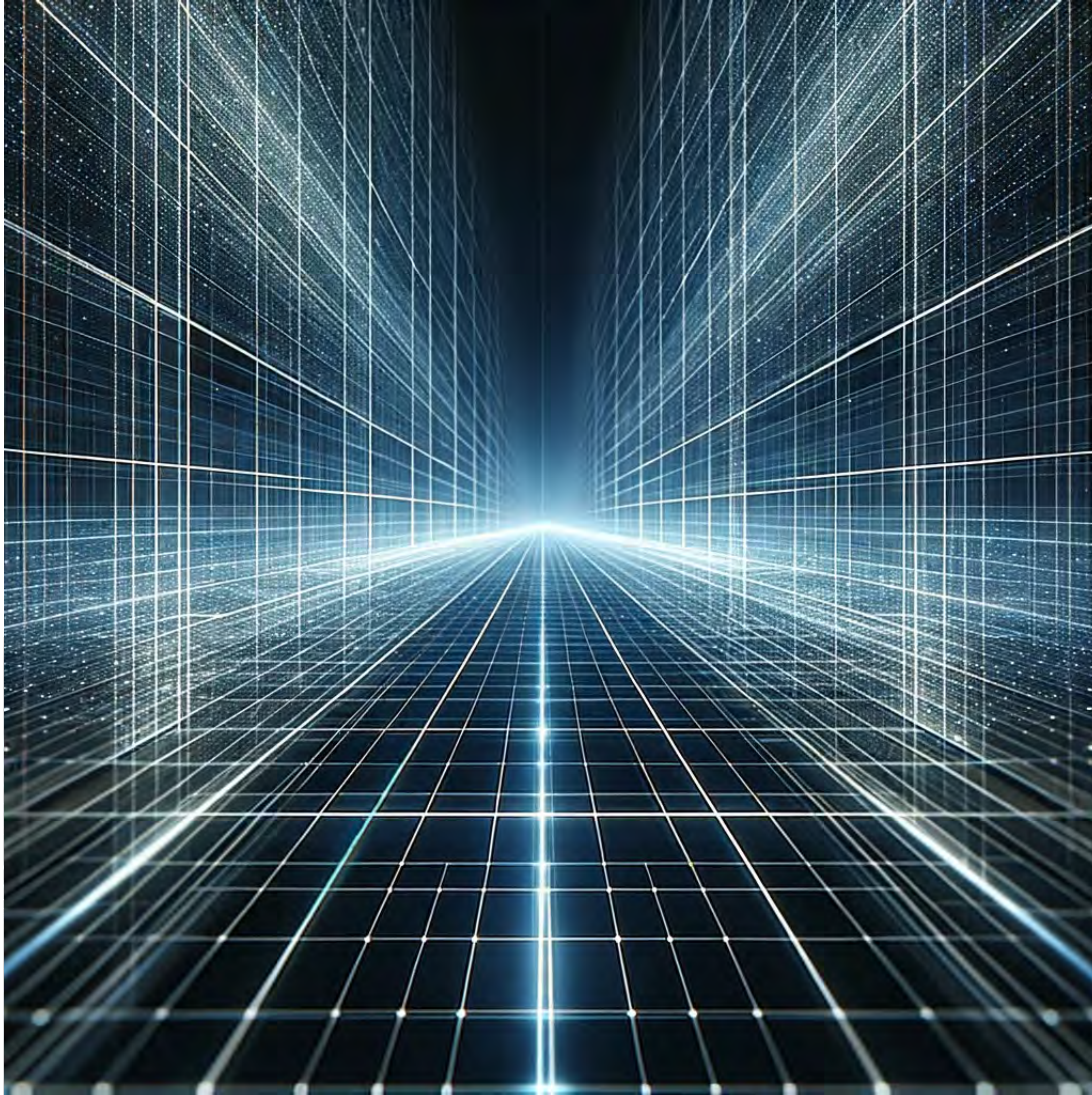
+

F : through RL

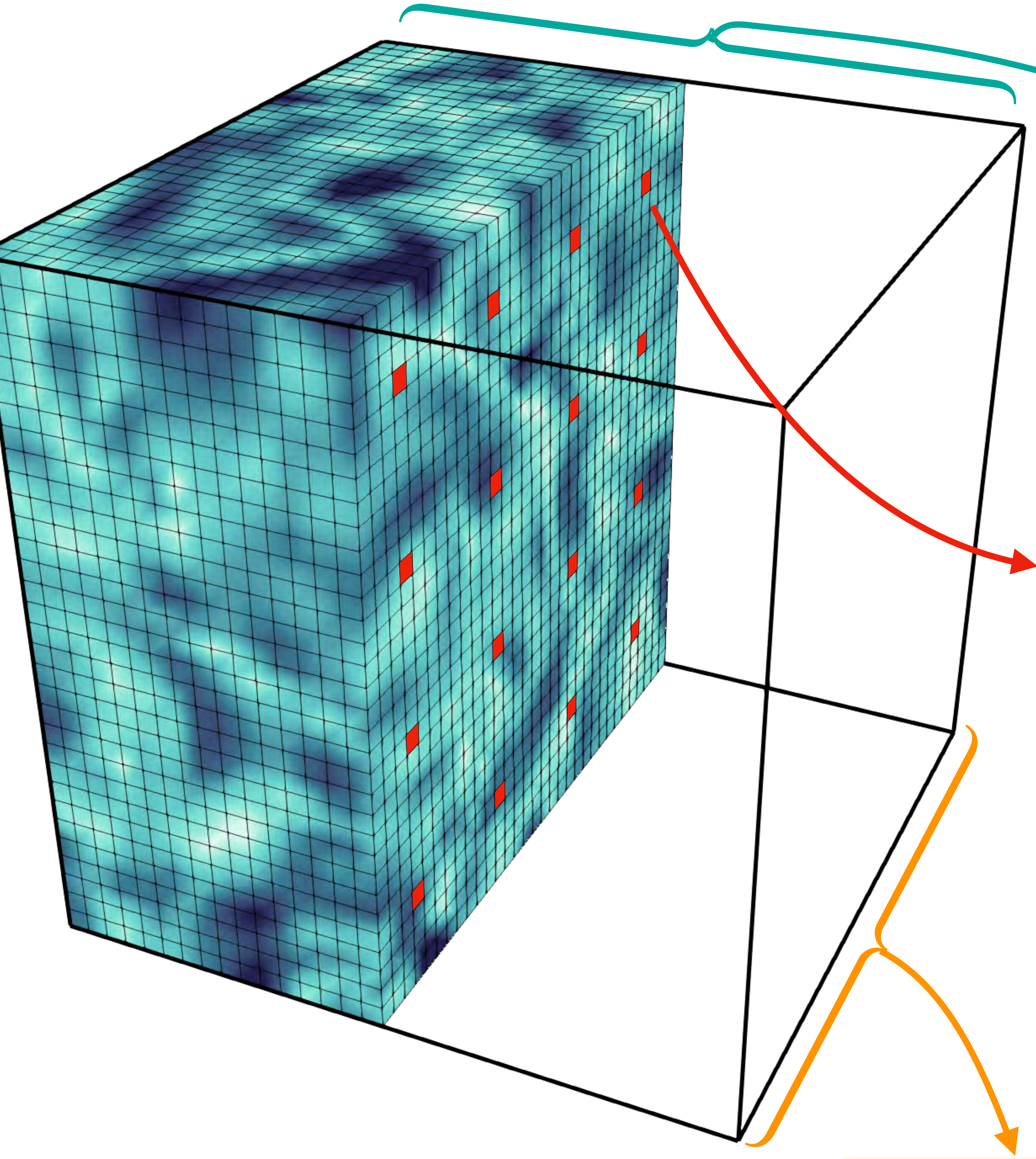
nature machine intelligence
ARTICLES
https://doi.org/10.1038/s42256-020-00272-0
Check for updates

Automating turbulence modelling by multi-agent reinforcement learning

Guido Novati[1], Hugues Lascombes de Laroussilhe[1,2] and Petros Koumoutsakos[1,3]



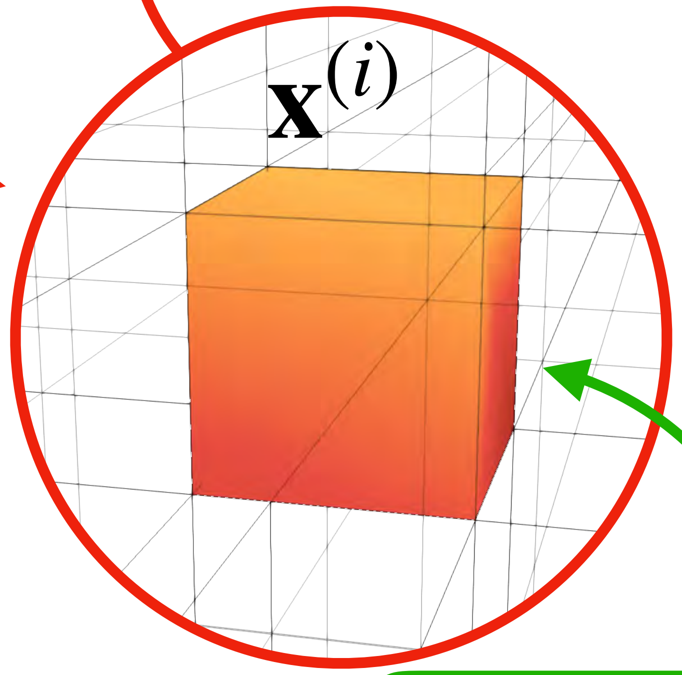
Multi-Agent Deep Reinforcement Learning



local & global state information

$$s^{grid}(t) = \left\{ \epsilon_{visc}(t),\ \epsilon_{tot}(t),\ E(k,t)\Big|_{k=1:15} \right\}$$

$$s\left(\mathbf{x}^{(i)}, t\right) = \left\{ \lambda_k^{\nabla u}\left(\mathbf{x}^{(i)}, t\right)\Big|_{k=1:5},\ \lambda_k^{\Delta u}\left(\mathbf{x}^{(i)}, t\right)\Big|_{k=1:6} \right\}$$



$$\nu_t = (C_s\Delta)^2 ||\mathbf{S}||$$

agent i

$$C_s\left(\mathbf{x}^{(i)}, t\right) = a\left(\mathbf{x}^{(i)}, t\right) \sim \pi_{\mathrm{w}}\left(\cdot \mid s\left(\mathbf{x}^{(i)}, t\right)\right)$$

$$r(t) = -\mathscr{D}(E, E_{DNS})$$

- Common policy
- Agents act locally on (C_s)
- Training on multiple Re_λ

Energy spectra for DNS (solid black line)

Standard Smagorinsky Model (purple), Dynamic Smagorinsky Model (green),

MARL policy π^{LL}, MARL policy π^{G} and MARL policy π^{LL} trained exclusively from data for Re = 111

Training set: $Re_\lambda \in \{65, 76, 88, 103, 120, 140, 163\}$

2D Turbulence: Prototype for atmospheric & oceanic flows *(with Pedram Hassanzadeh-Rice U.)*

Governing equations

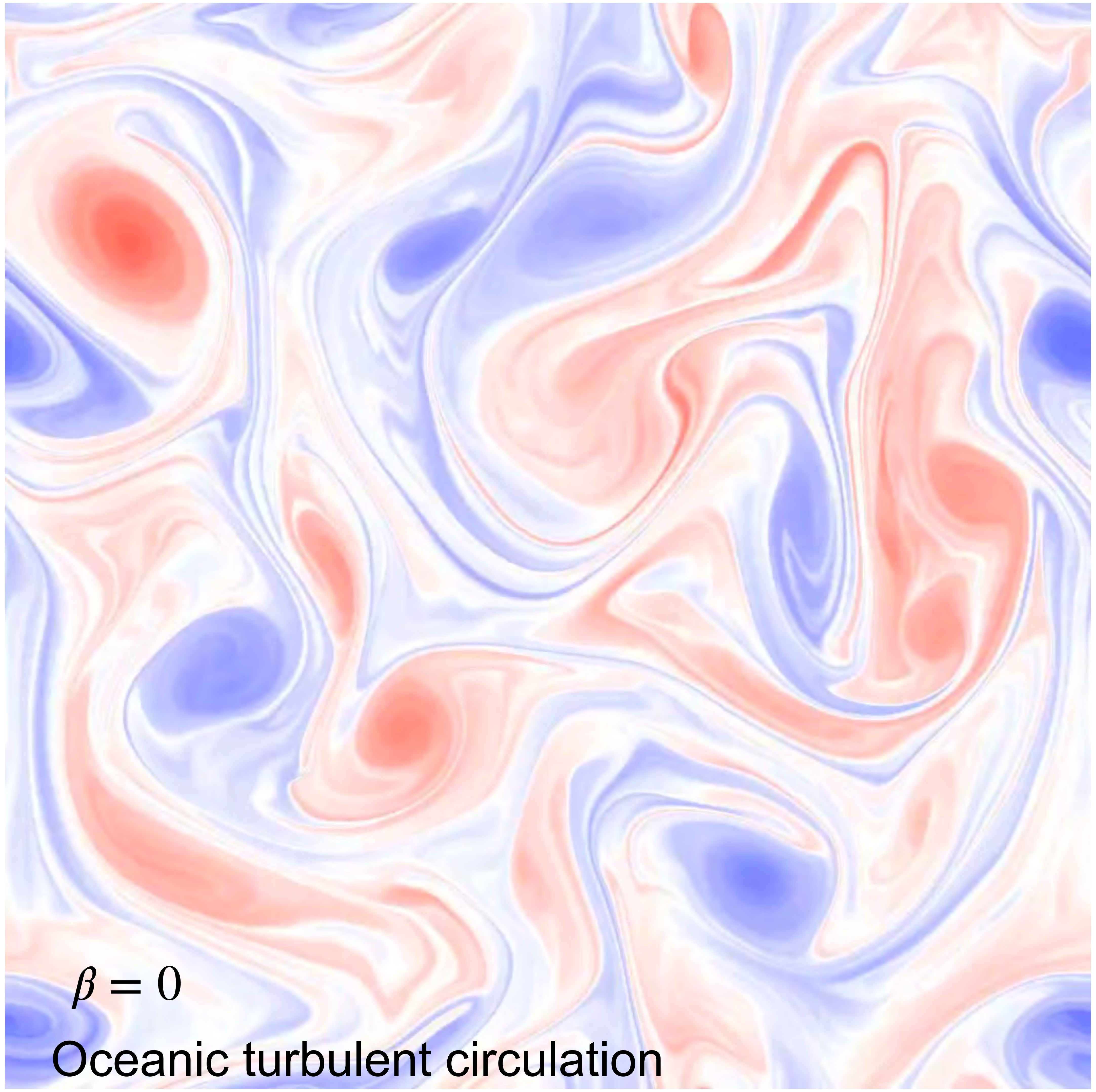
$$\nabla^2 \psi = \omega$$

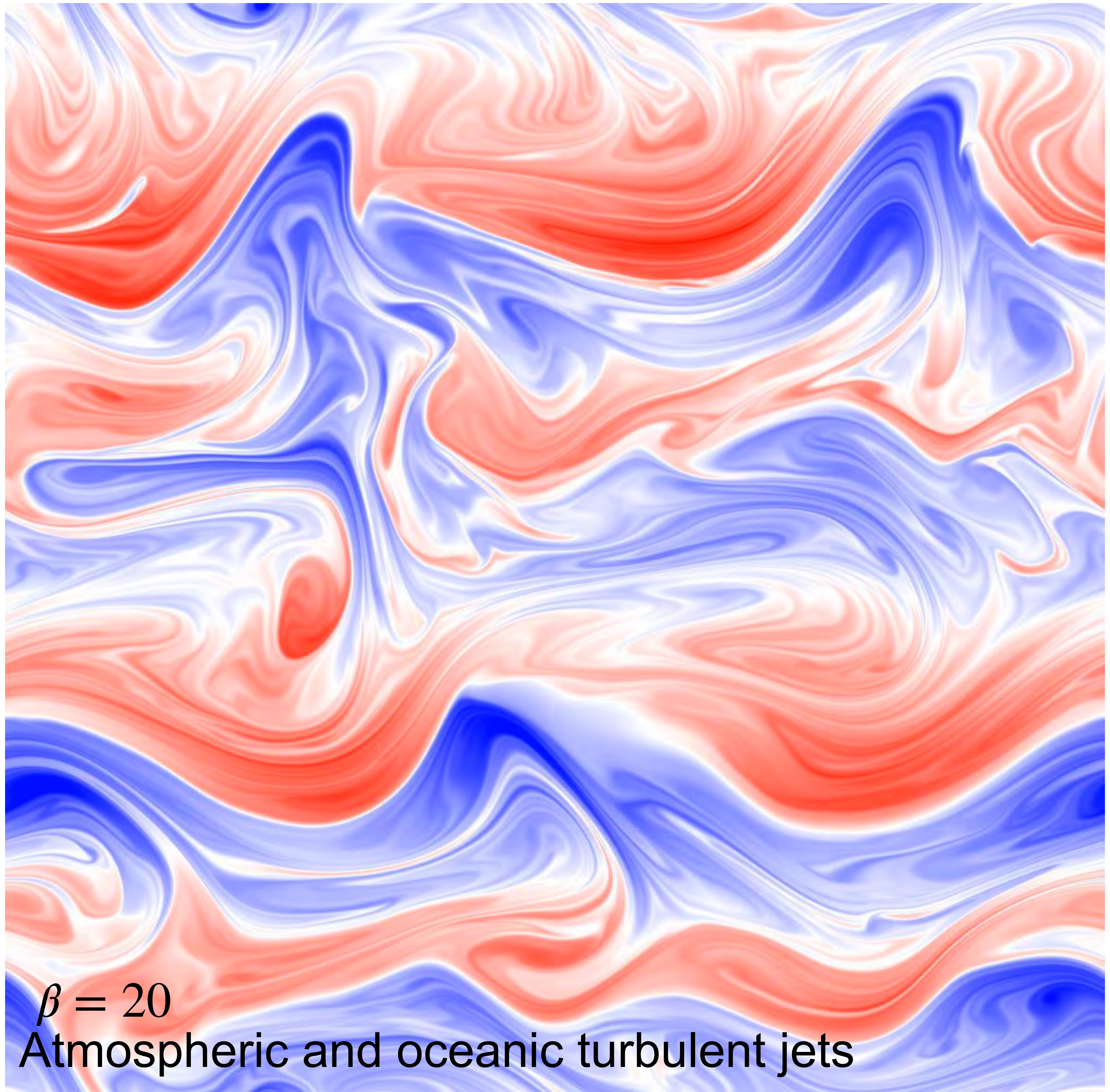
$$\frac{\partial \omega}{\partial t} + N(\omega, \psi) - \beta \psi_x = \frac{1}{Re} \nabla^2 \omega + f - r\omega$$

forcing

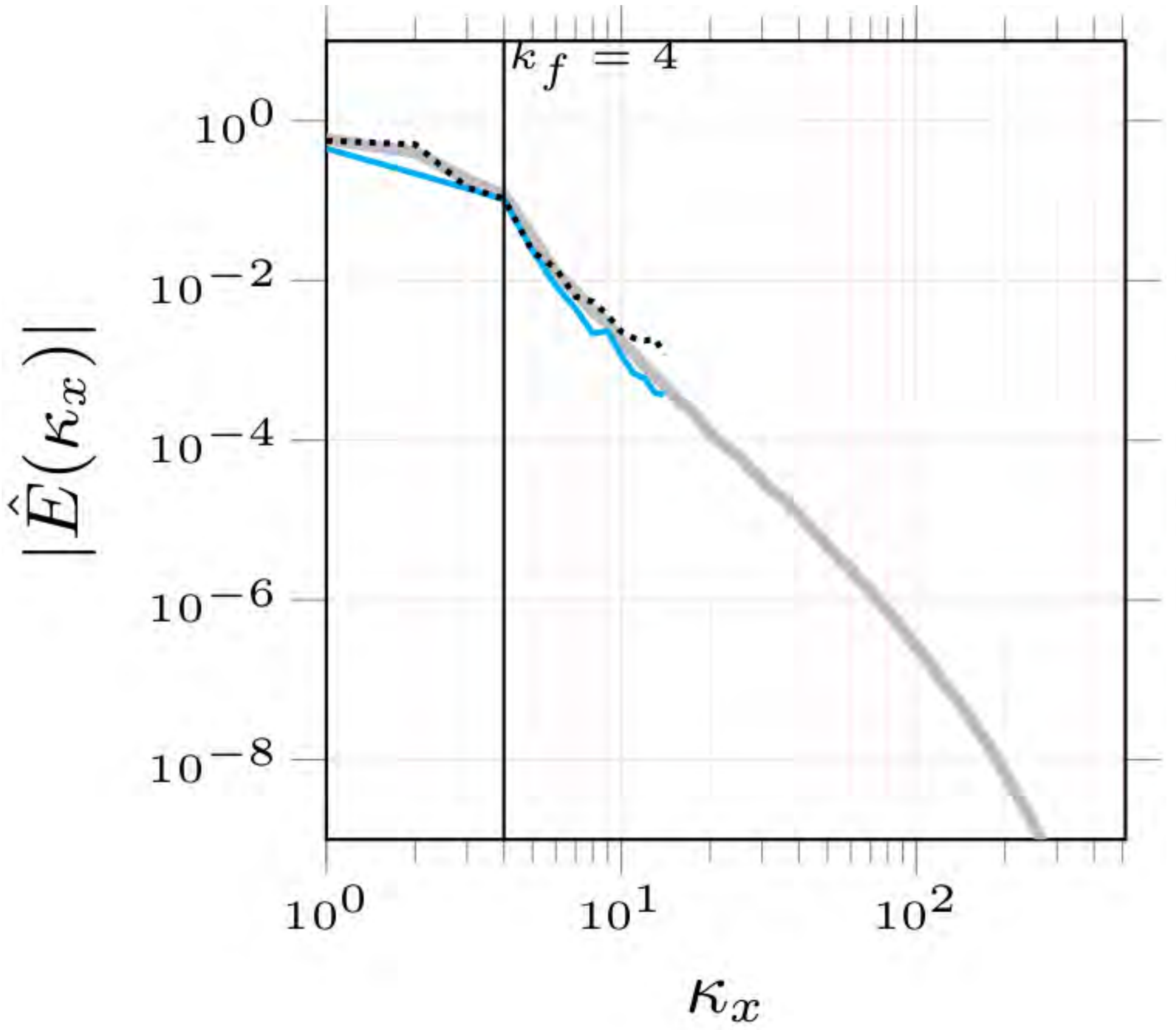
$$f(x, y) = \kappa_f [\cos\left(\kappa_f x\right) + \cos(\kappa_f y)]$$

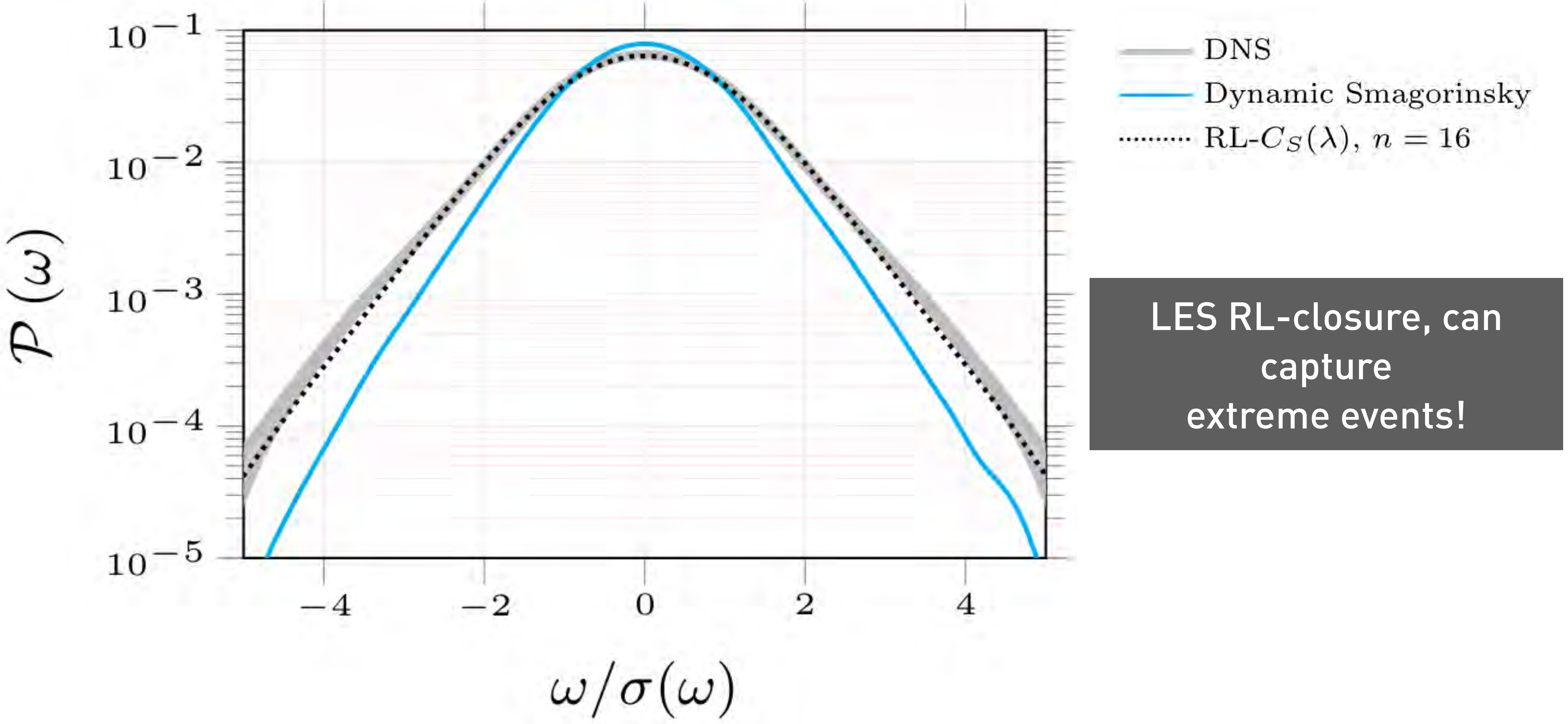
$$N(\omega, \psi) = \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y} - \frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x}$$





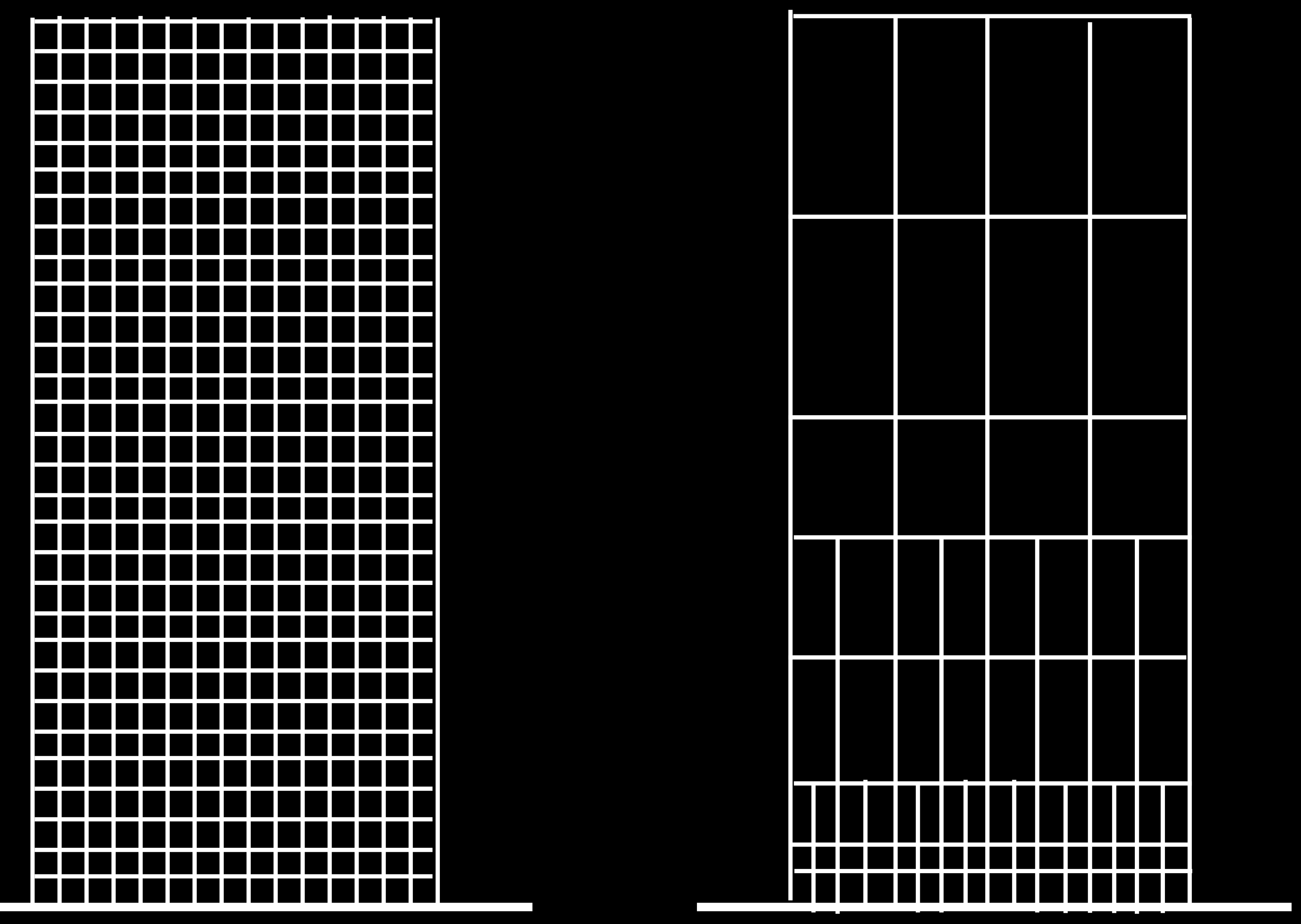
- Reynolds number: Re=20’000 , beta = 0
- **LES:** 32 x 32, 10x coarser in time (~10000x fewer DOFs than)
- **Data:** spectrum from 20 DNS snapshots - **Reward:** enstrophy spectrum - **States;** Local Invariants
- RL: learn $C_s(x,y,t)$ of Smagorinsky closure as a function of resolved flow (16 agents)
- **Tests:** TKE spectrum, PDF of vorticity (weather), including tails (extreme weather)





LES RL-closure, can capture extreme events!

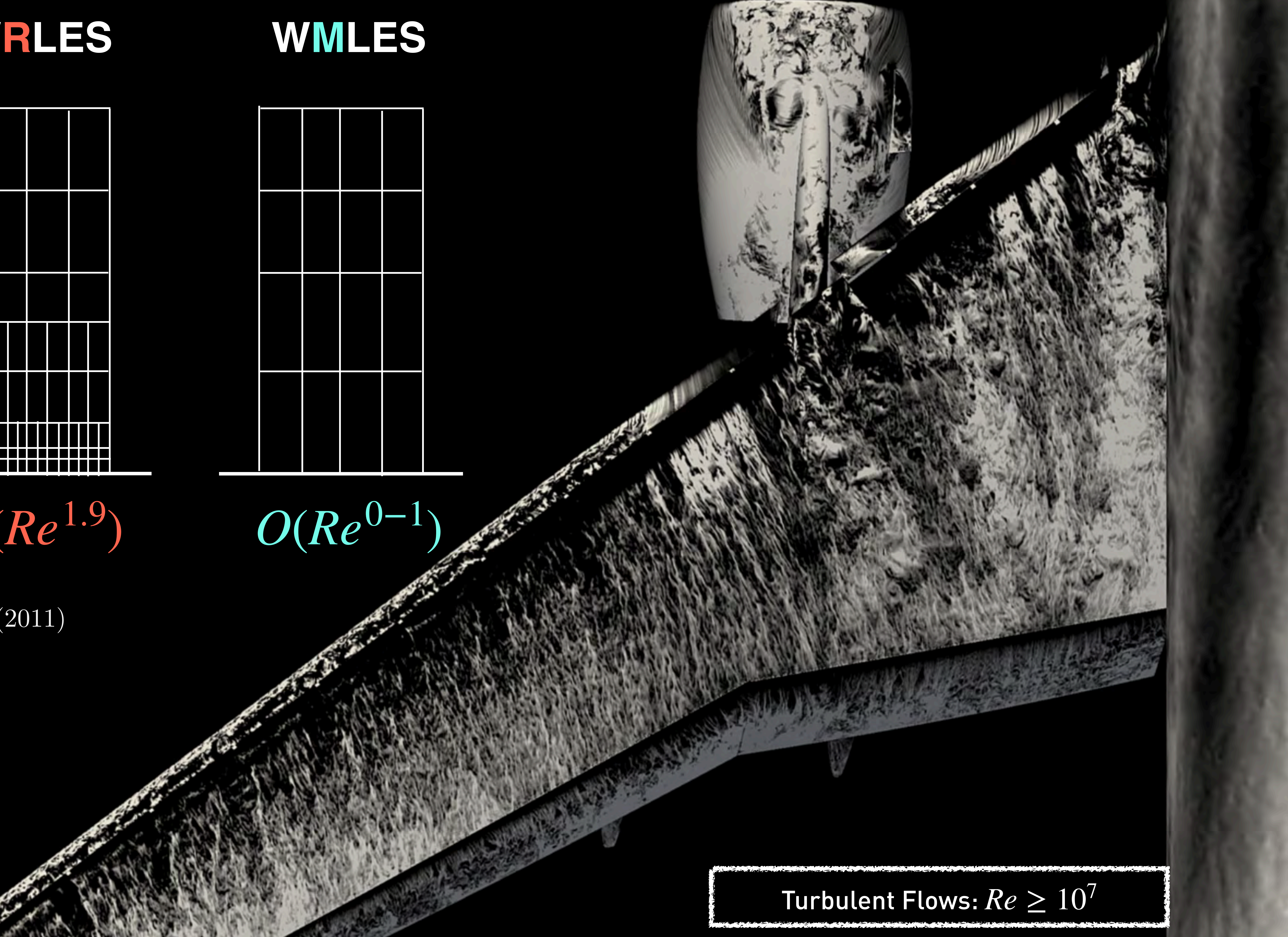
DNS | **WRLES** | **WMLES**



$O(Re^{2.6})$ | $O(Re^{1.9})$ | $O(Re^{0-1})$

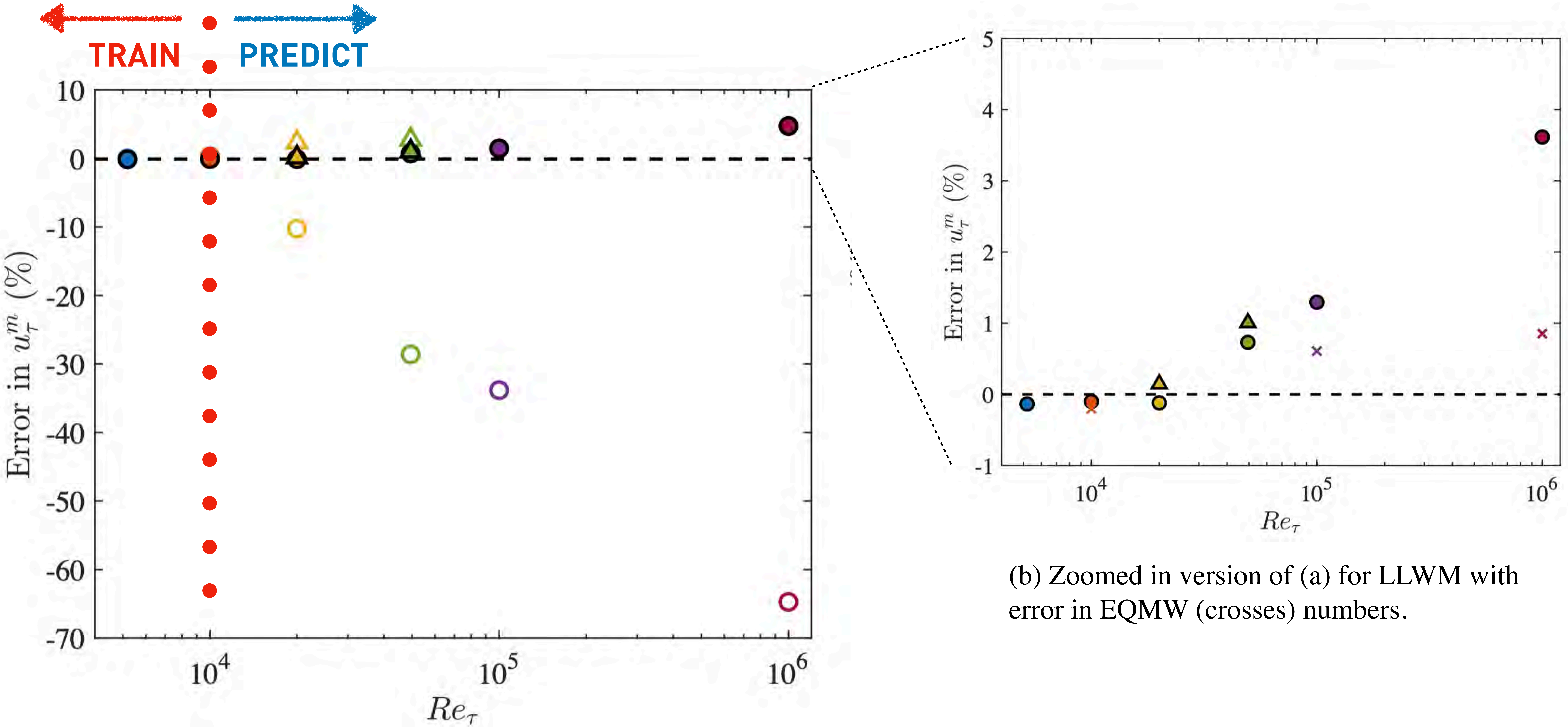
Chapman (1979), Choi & Moin (2011)

How many grid points ?



Turbulent Flows: $Re \geq 10^7$

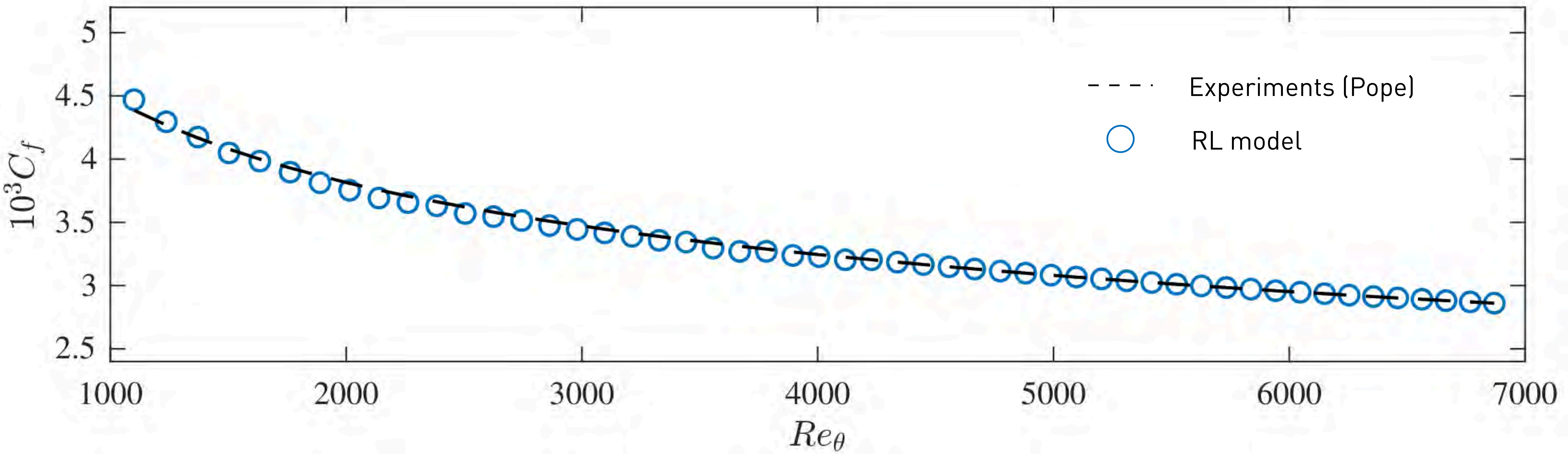
$1mm^3$ on
an airplane wing
nature
COMMUNICATIONS
ARTICLE
Check for updates
https://doi.org/10.1038/s41467-022-28957-7
OPEN
Scientific multi-agent reinforcement learning for wall-models of turbulent flows
H. Jane Bae 1,2 & Petros Koumoutsakos 1,3
WALL
TURBULENCE



Error in time-averaged wall-shear stress obtained from the VWM (empty) and LLWM (filled) for various Reynolds numbers. Circles indicate the standard grid with $\Delta y = 0.05$ and triangles indicate refined cases.

(b) Zoomed in version of (a) for LLWM with error in EQMW (crosses) numbers.

TESTING II: Evolving turbulent boundary layer



CLOSING THOUGHTS

Comment

https://doi.org/10.1038/s42254-024-00726-z

On roads less travelled between AI and computational science

Petros Koumoutsakos

Check for updates

Computational science and artificial intelligence have been drivers and benefactors of advances in algorithms and hardware, each in different ways, and originally with different targets. Petros Koumoutsakos argues that the intellectual space between these two fields is home to exciting opportunities for scientific discovery.

'importance sampling'. There are plenty of opportunities for cross-fertilizing exchanges in algorithms and their applications. Similarly, stochastic and gradient optimization methods have been developed across both communities, but recent works on automatic differentiation indicate that the paths are intersecting again. The emergence and homogenization properties of foundational models that are gaining ground in AI also have counterparts in CoS where emergence is often the outcome of nonlinear differential equations, whereas the concept of homogenization can be recognized for example in particle simulations of phenomena ranging from atoms to galaxies[3]. At the same time the paths of scientific inquiry in AI and CoS may diverge, but I argue that repeated intersection can be exciting. There are many problems where

Nature Reviews Physics, 2024


the future...
CREDIT: THE ATLANTIC

Thank you !