Deep Reinforcement Learning for Flow Control

Petros Koumoutsakos
ETH Zürich

with:

M. Gazzola (UIUC), A. Tchieu (Space-X), W. van Rees (MIT), S. Verma (FAU), D. Alexeev (NVIDIA)

G. Novati, P. Vlachas, P. Weber

Artificial Intelligence:? Computational ability to achieve goals

John McCarthy

From the movie “Microcosmos”

Learning
to
Optimize

STOCHASTIC OPTIMIZATION for FLUID MECHANICS

Black Box scenario

➡In Fluid Mechanics gradients are not available or not useful

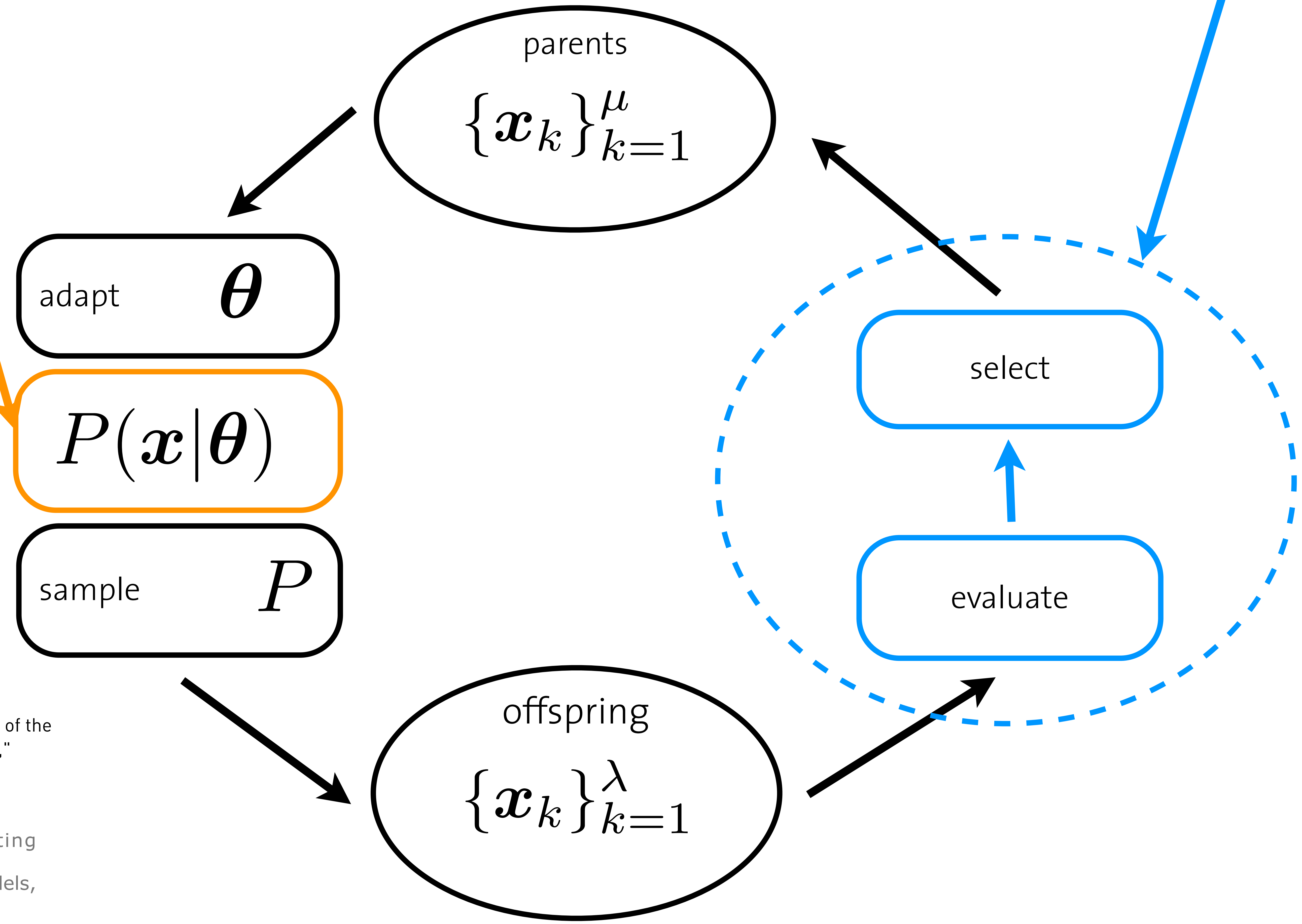
➡Commercial(Black box) solvers, Experimental Set-ups

➡Multiple Local Minima, Noisy Output

Surrogates and Covariance Matrix Adaptation ES

Model to render search more efficient than random sampling

meta-models to replace expensive evaluations in the selection process

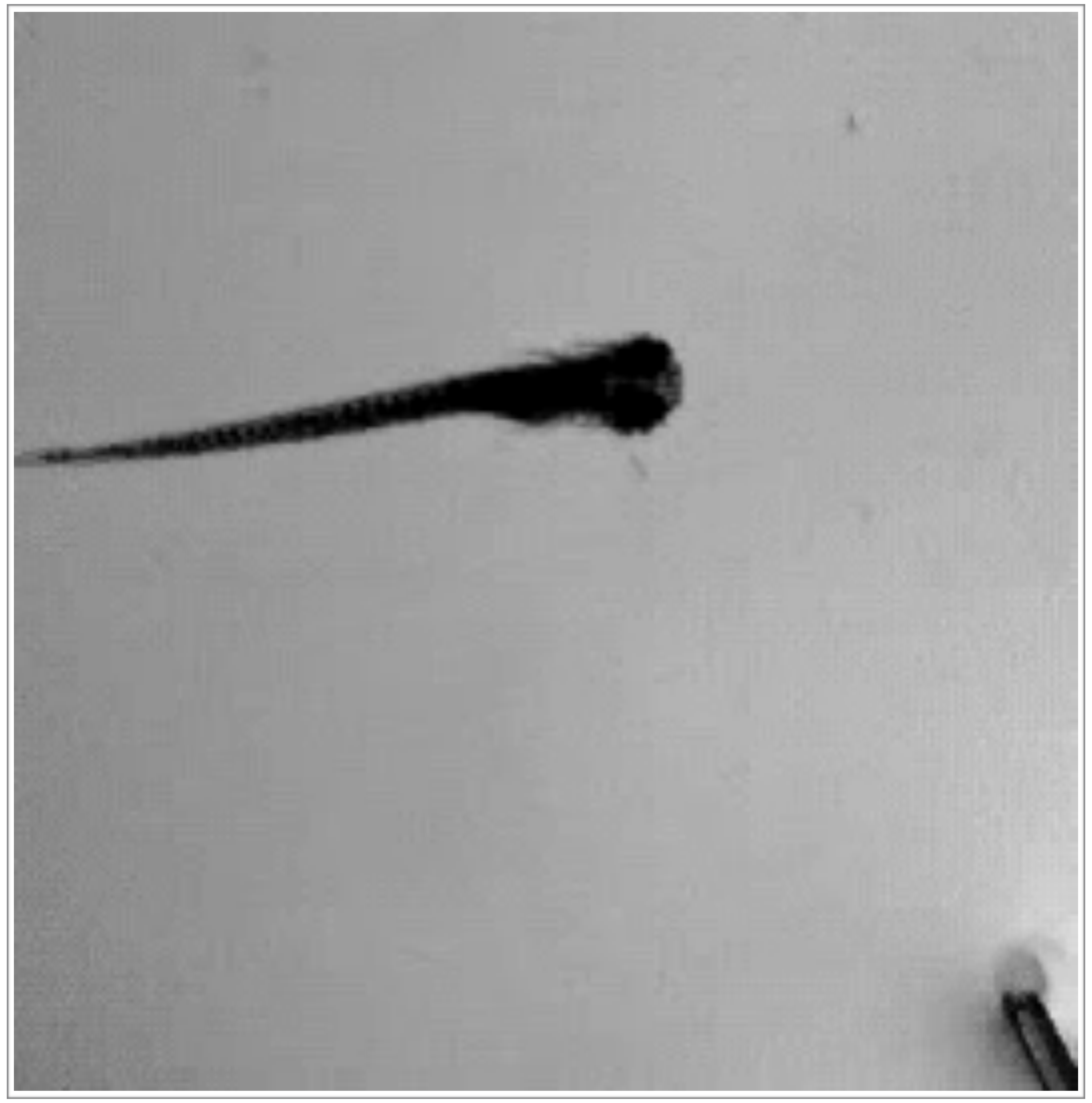


N. Hansen, S. D. Müller, and P. Koumoutsakos, "Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES)," ***Evolutionary computation***, **vol. 11, iss. 1, p. 1–18, 2003.**

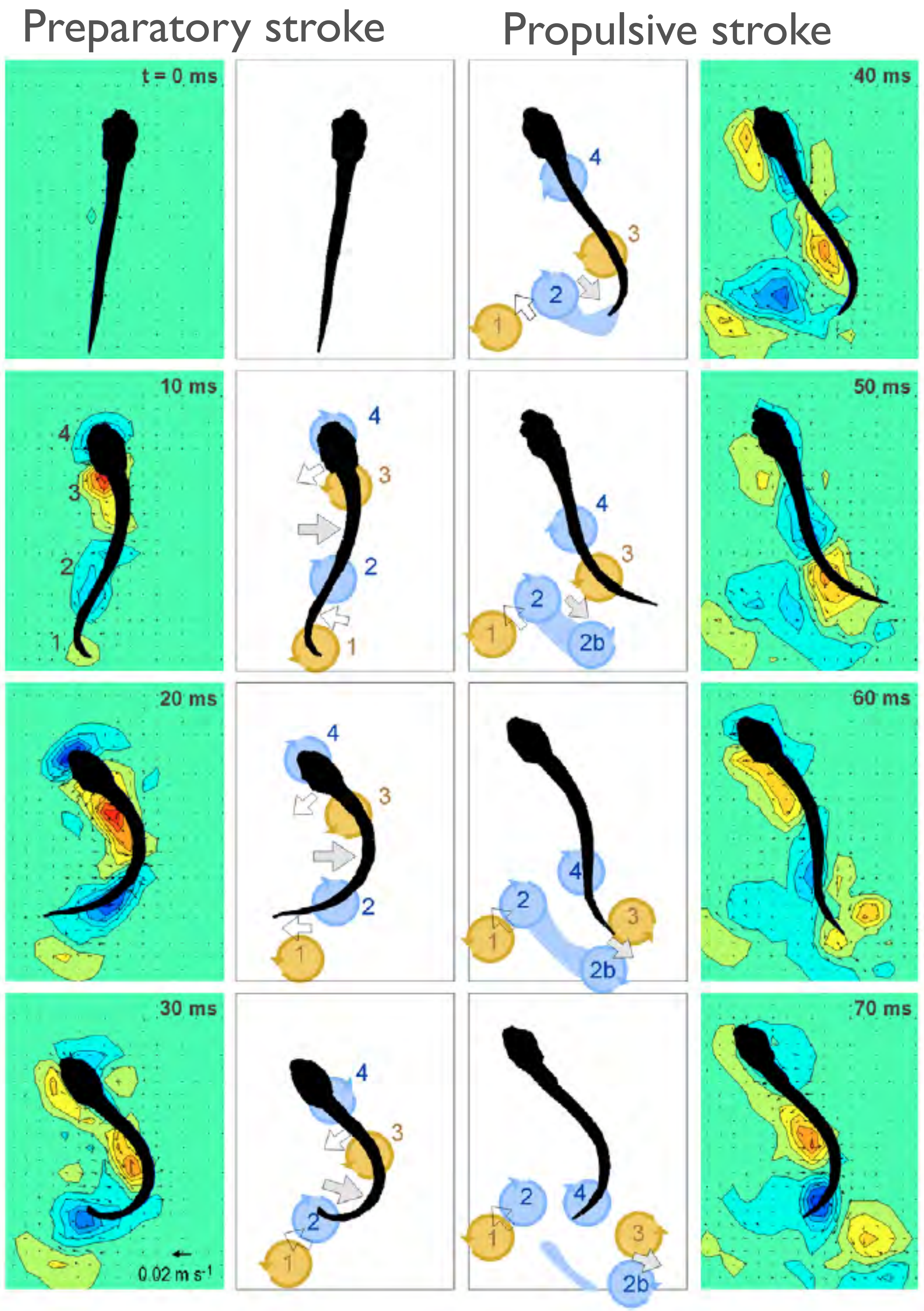
D. Bueche, N. Schraudolph, P. Koumoutsakos, Accelerating evolutionary algorithms with Gaussian process fitness function models, **IEEE Trans. on Systems, Man and Cybernetics**,, 35,, **2005**

C-start is an escape pattern

Is C-start optimal?

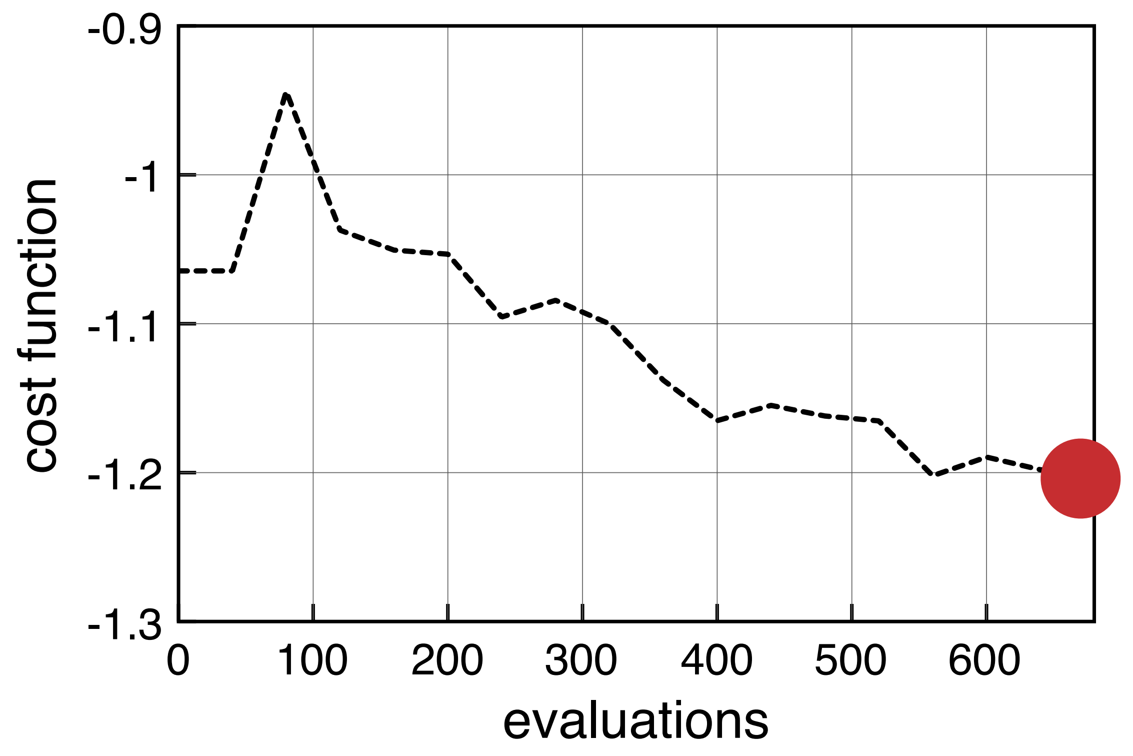


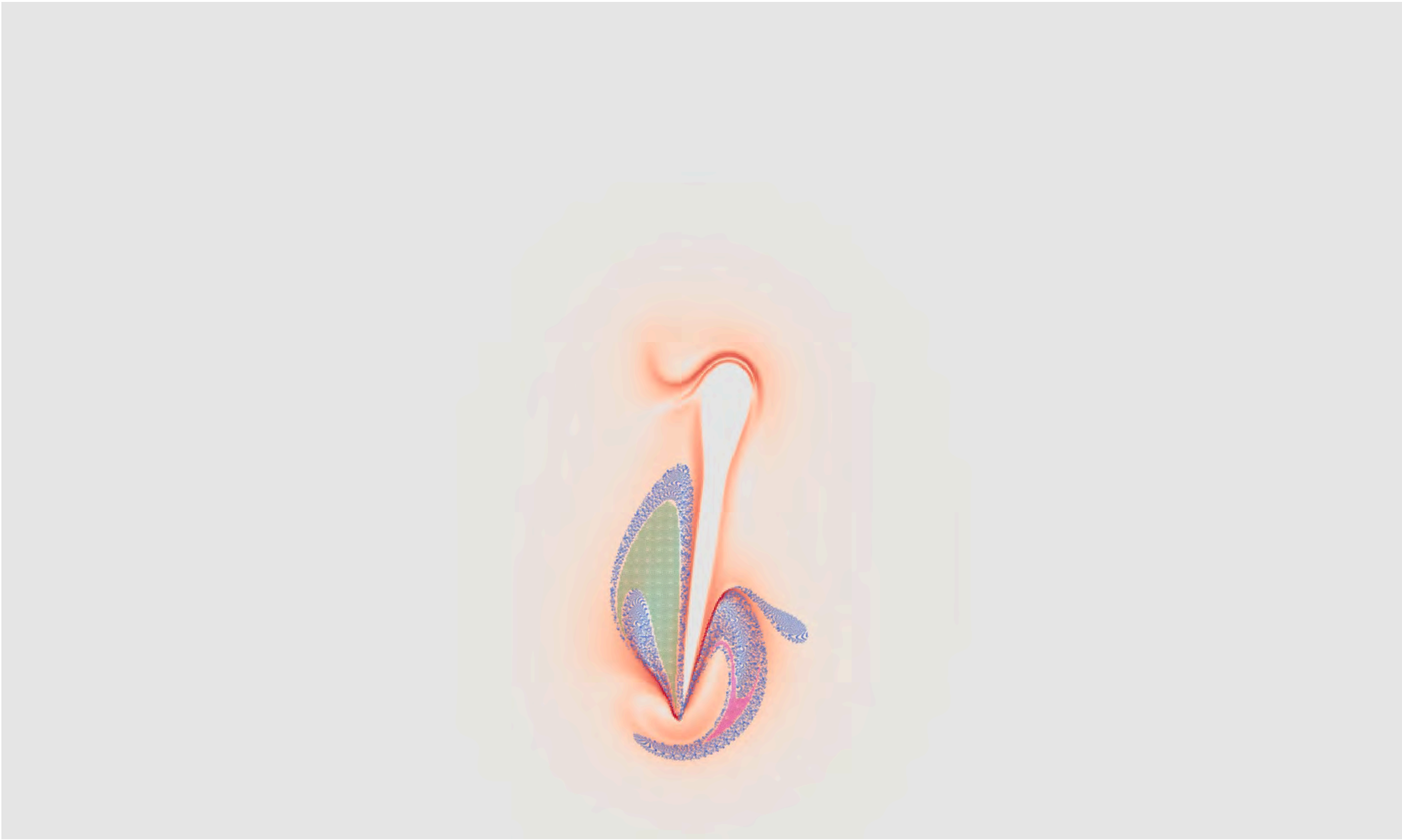
Liao Lab's Channel - YouTube

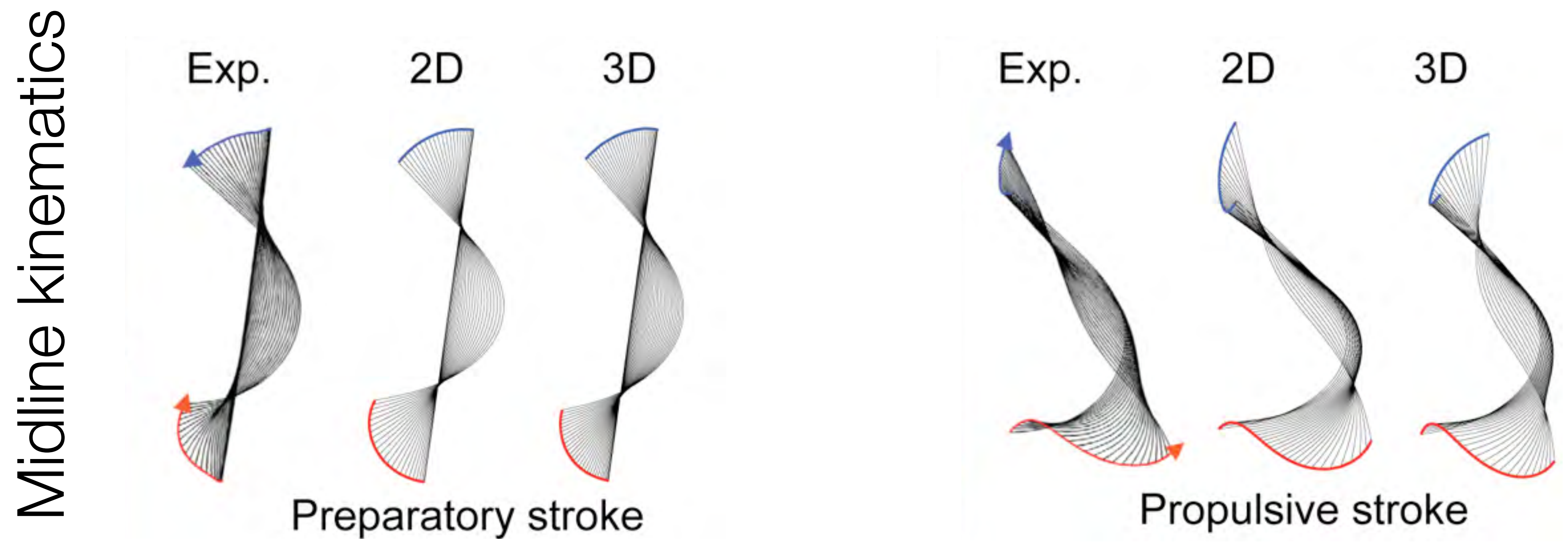


Muller, van den Boogaart, van Leeuwen. J. of Exp.Biology, 2008.

C-start is **OUTCOME** of optimization







1. Fluid region trapped by C-shape
2. Acceleration by propulsive stroke


STOCHAST
MACHINE
LEARNING
U.S. Patent
Jul. 4, 2000
Fig.3A
CREDIT SUISSE

Learning

to

Control

FISH SCHOOLING

- Behavioral Traits - Vortex Dynamics
- Energetic benefits ?

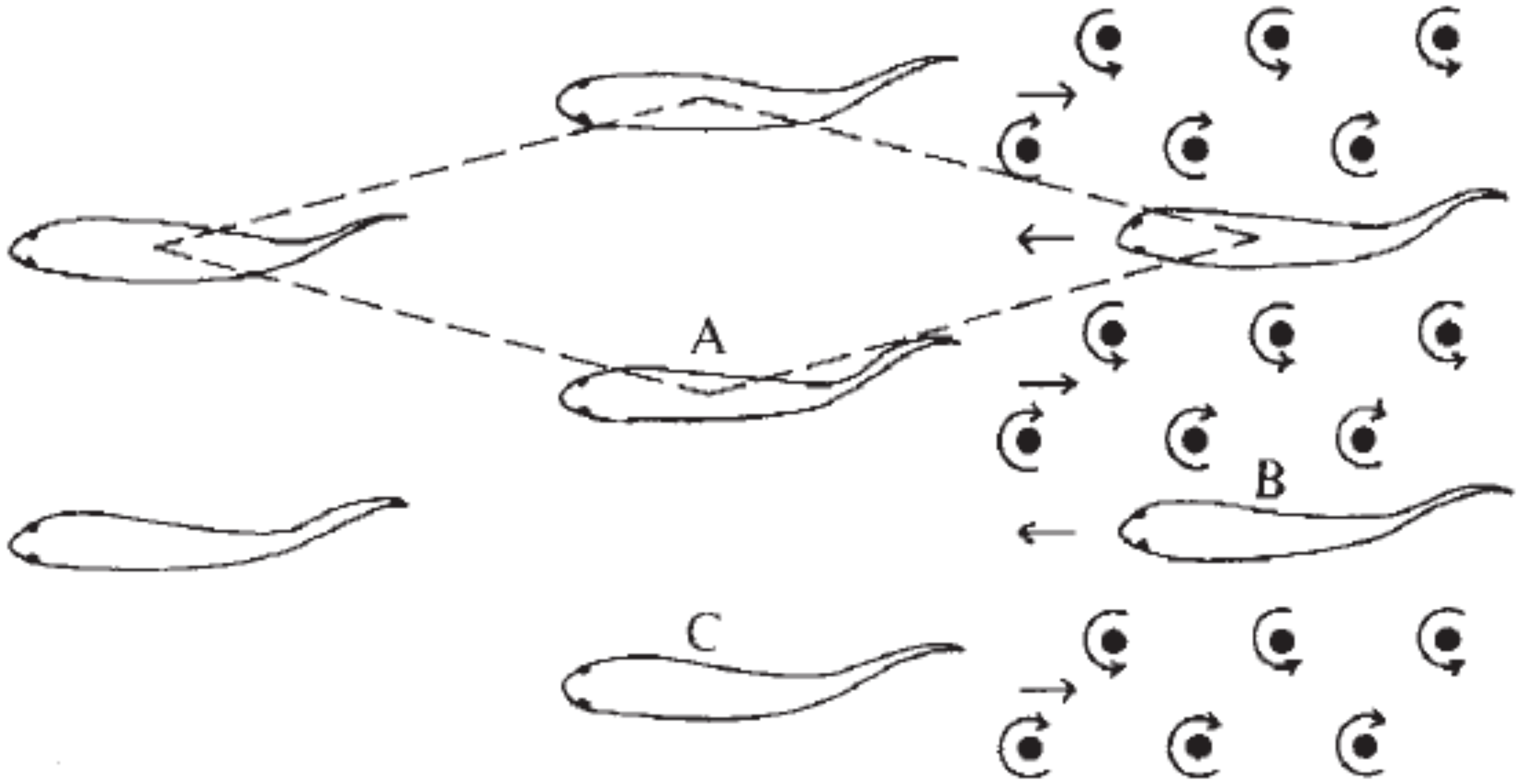
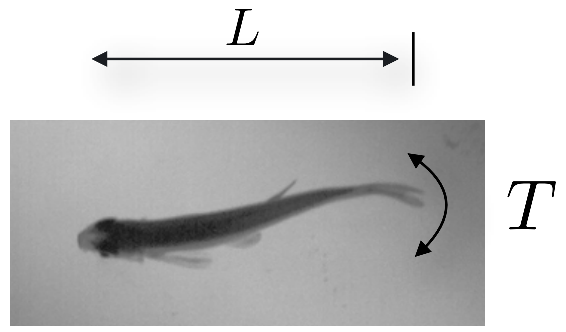


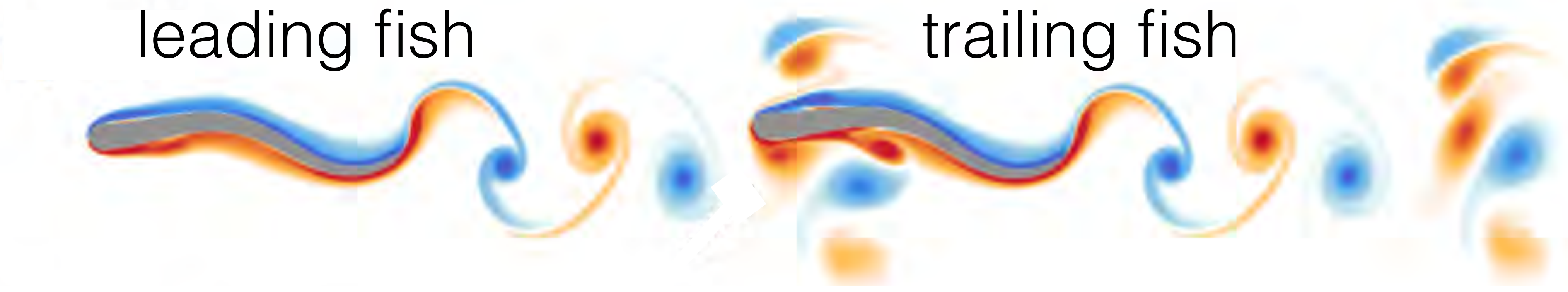
Fig. 1 Part of a horizontal layer of fish in a school, from above. Arrows near vortex streets show direction of induced flow relative to the vortices. The dotted line shows a "diamond" pattern.

- Simple model of fish schooling: a leader and follower

$$Re = \frac{L^2/T}{\nu} = 5000$$



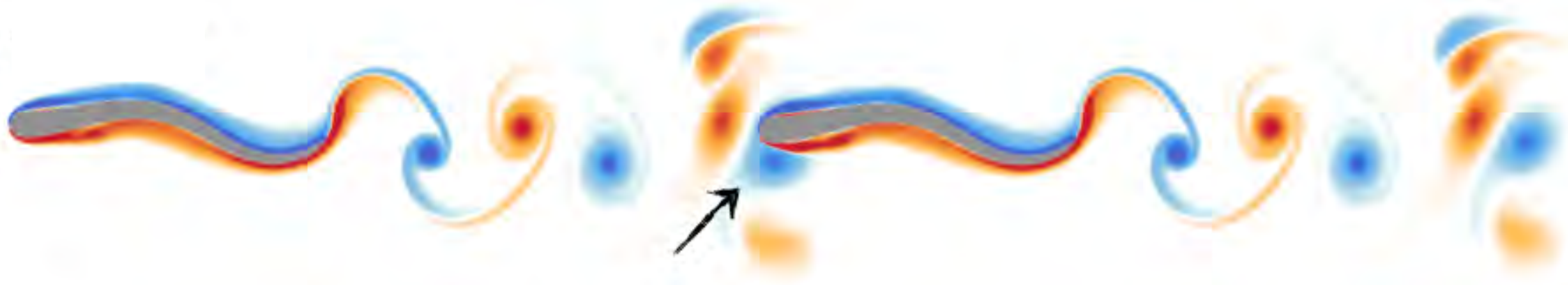
Trailing



Initial tail-to-head distance = 1 L

- Swimming in a wake can be **beneficial** **or detrimental**

Trailing fish's head intercepts **negative vorticity**: velocity decreases

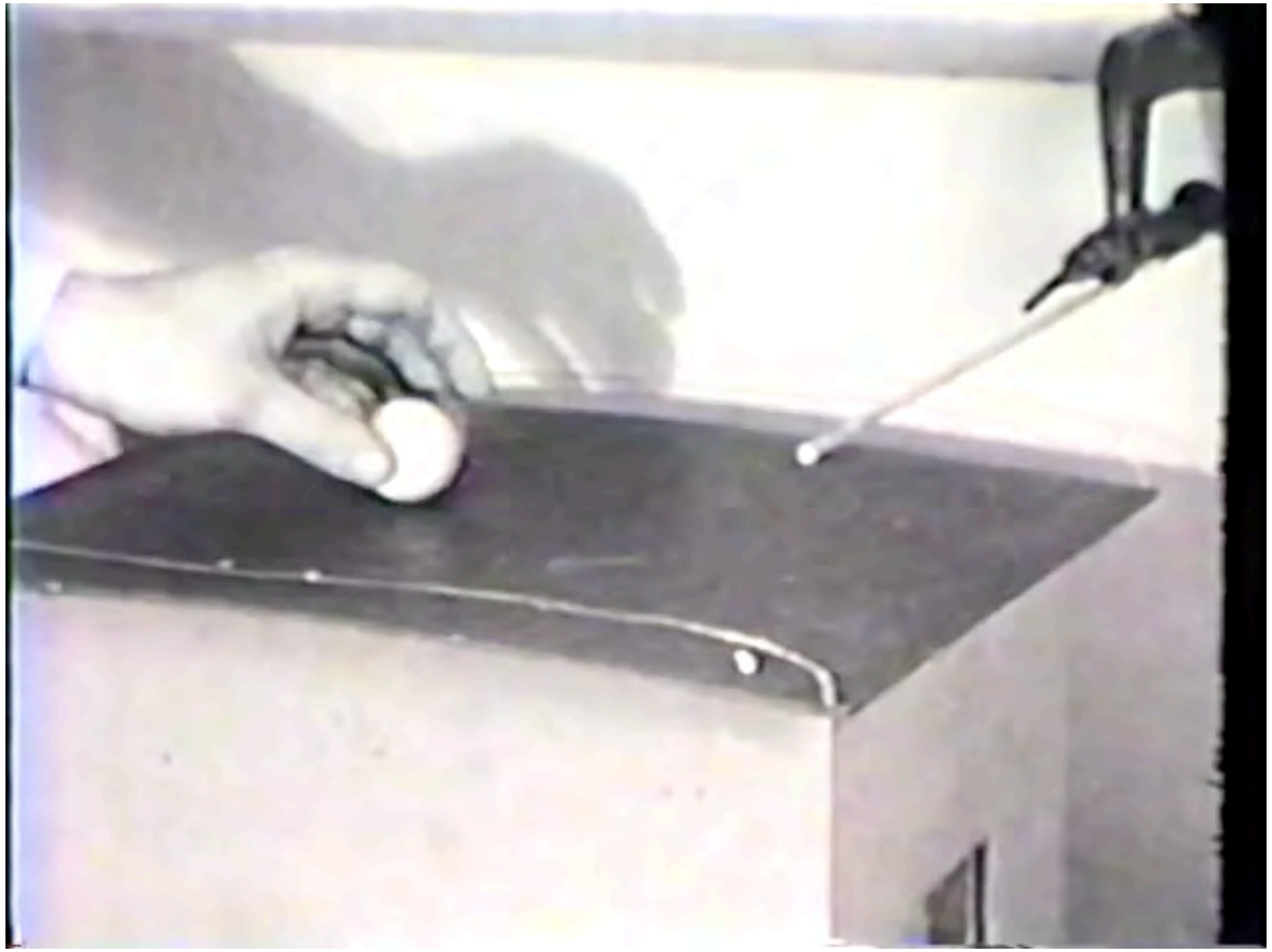


Initial tail-to-head distance = 1.25 L

Reinforcement Learning

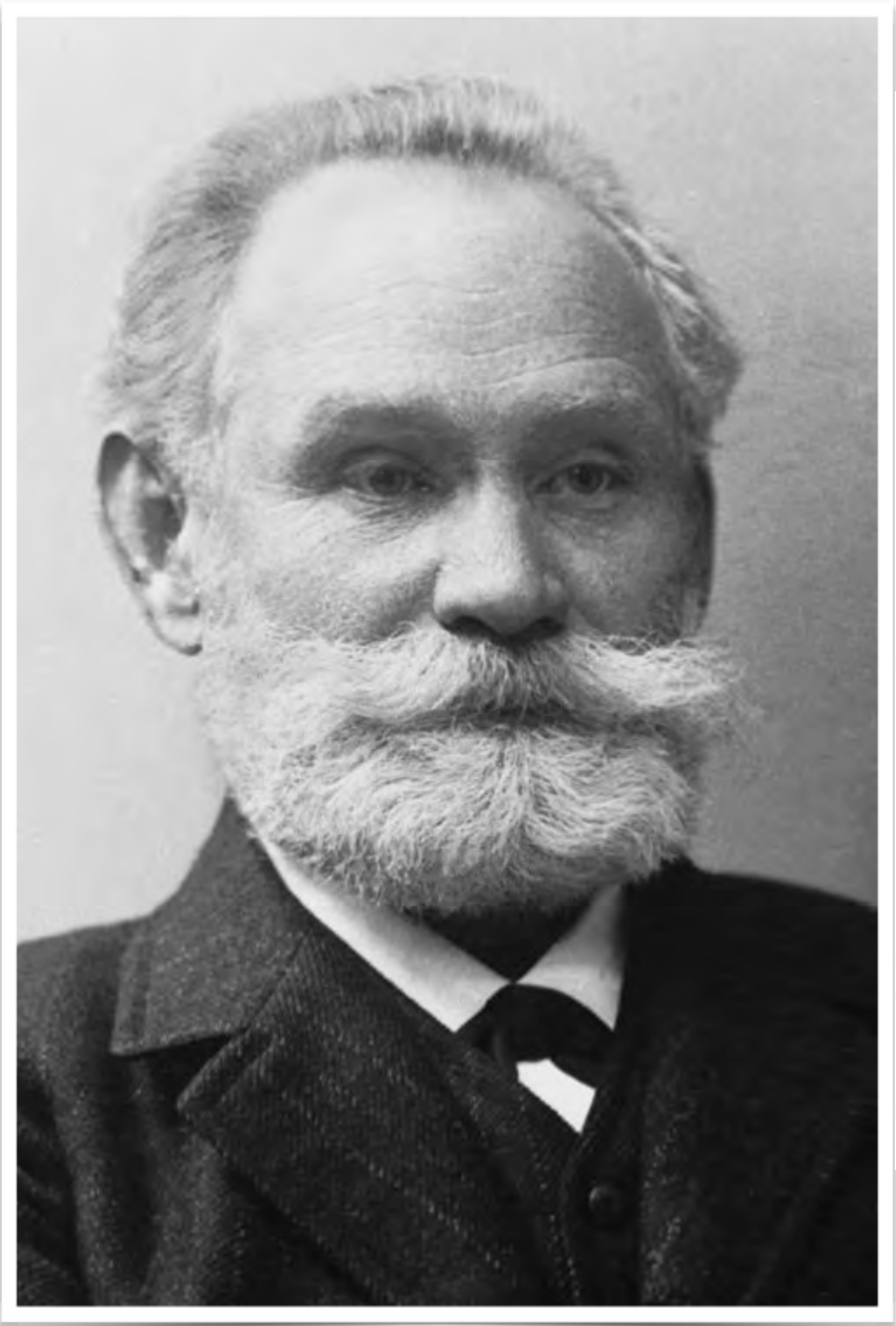
Learning: Behavioral changes due to Experiences (Action, Stimulus, Reward)

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



CREDIT: B.F. Skinner Foundation

RL in Psychology: Conditioning



Ivan Pavlov - 1890

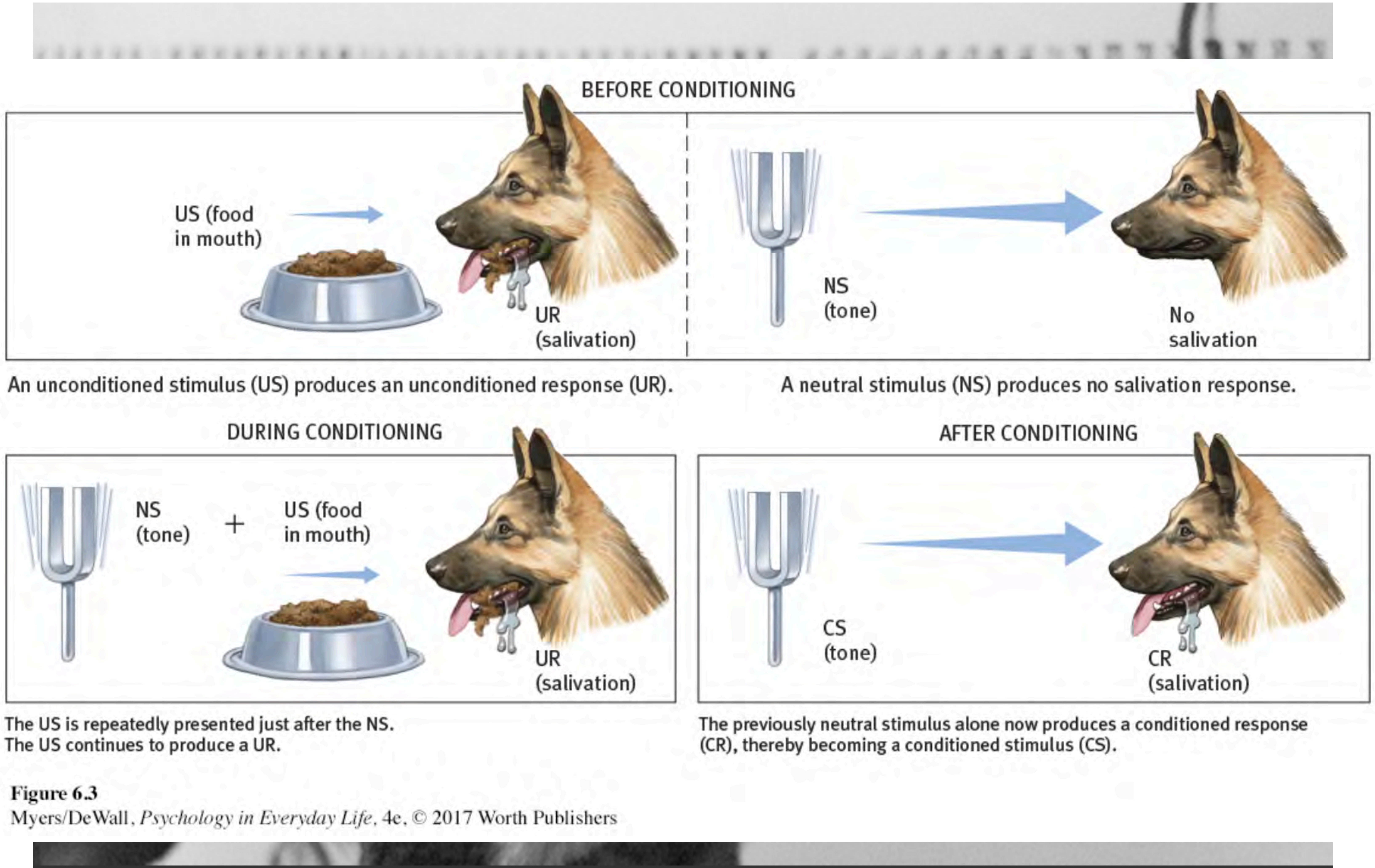


Image: Rklawton

Reinforcement Learning: Over 150 years of history

- **Psychology** - Behaviorism and Decision Making - *I. Pavlov, R.F. Skinner*
- **Mathematics** - Dynamic Programming - *P.J. Werbos, D. Bertsekas, J. Tsitsiklis*
- **Economics** - Game Theory - *John von Neumann*
- **Computer Science** - Algorithms and Deep Networks - *A. Barto, R. Sutton,* `DeepMind`
- …

I. DYNAMIC PROGRAMMING ->REINFORCEMENT LEARNING

Markov Decision Processes

action

$$X_{t+1} = F(X_t, \alpha_t, \epsilon_t)$$

THE FLOW SOLVER

Observation/State

state

$$s_t = \mathcal{G}(X_t, \eta_t)$$

THE DATA

Expected Utility

reward

$$J = \left\langle \sum_{t=1}^{T} \gamma^t \mathcal{R}(s_t, \alpha_t) \right\rangle$$

THE COST FUNCTION

REINFORCEMENT LEARNING:

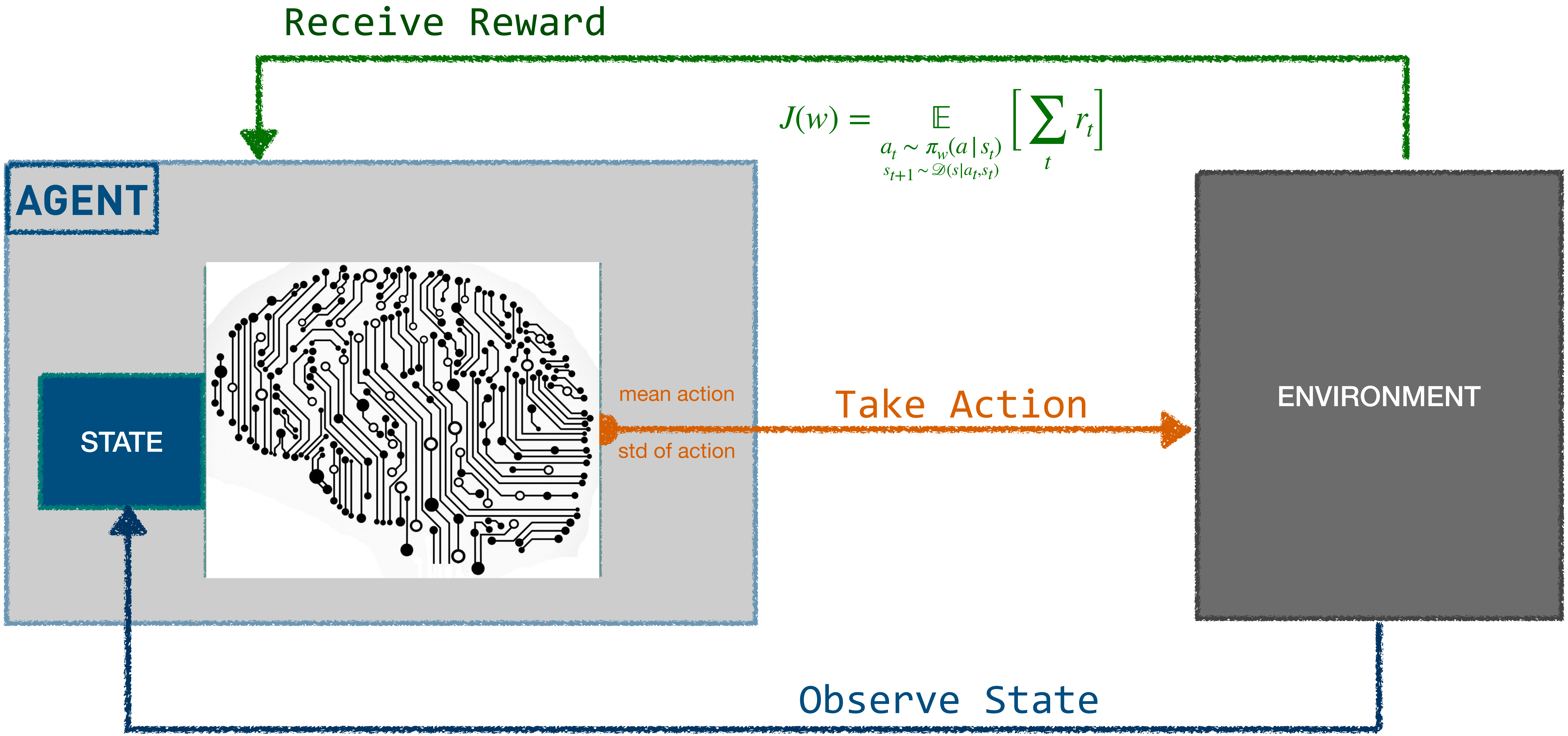
$D = (\mathcal{G}, \mathcal{R})$ observed

BUT

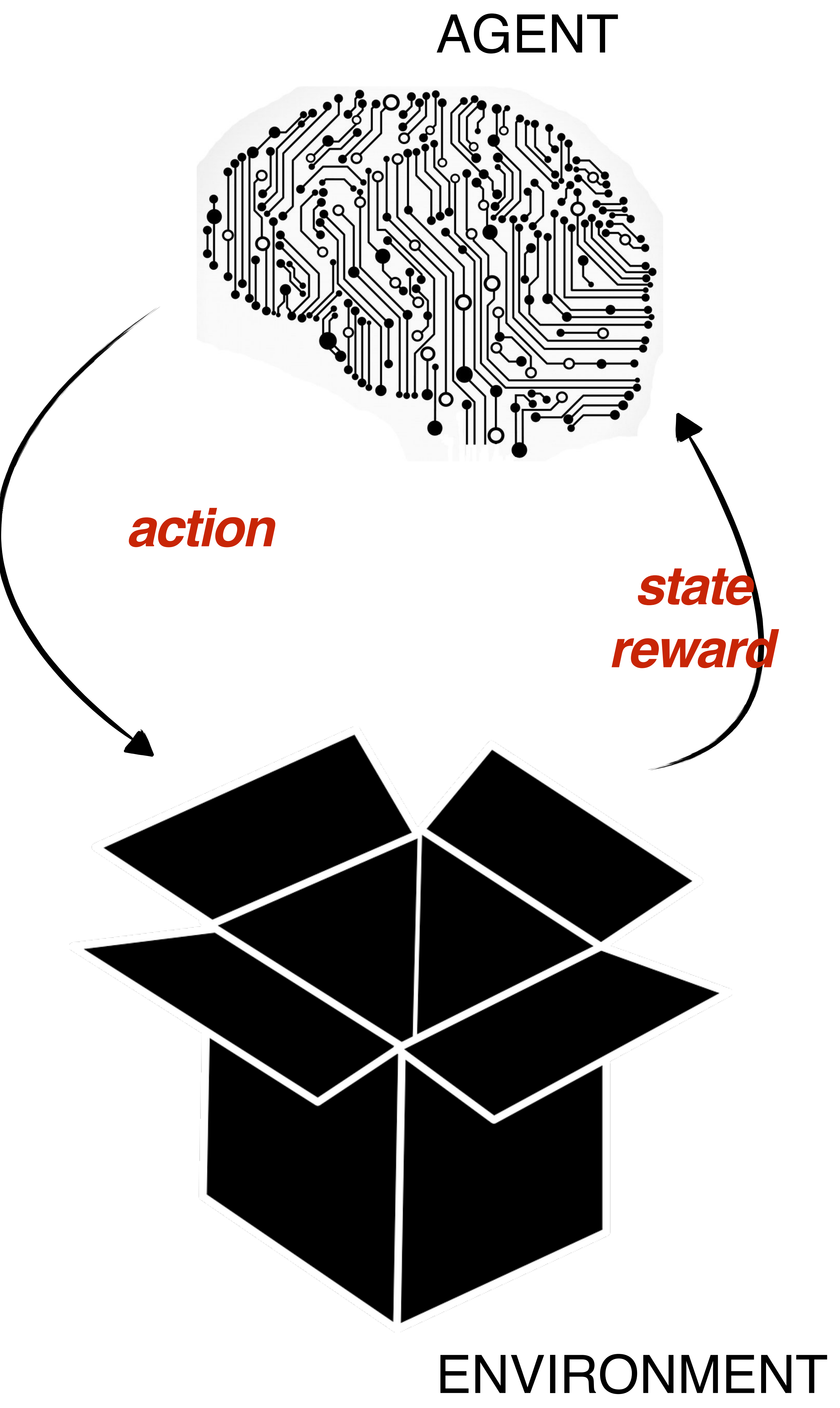
not known -

SAMPLING

II. Reinforcement Learning: Find Policy to Maximize Long Term Reward



REINFORCEMENT LEARNING : An **agent** learning an **action policy** trough **rewards**



Goal: Maximize the **value function**

$$V_{\pi_w}(s) = \underset{\substack{a_k \sim \pi_w(a \mid s_k) \\ s_{k+1} \sim \mathscr{D}(s|a_k, s_k)}}{\mathbb{E}} \left[\sum_{k=0}^{\infty} \gamma^t r_{t+k+1} \mid s_t = s \right]$$

➡ **The celebrated**
- **Bellman Equation:**

$$\Longrightarrow \quad V_{\pi_w}(s) = \sum_a \pi_w(a \mid s) \sum_{s', r} D(s', r \mid s, a) \left[r + \gamma \; v_{\pi_w}(s') \right]$$

value-action function

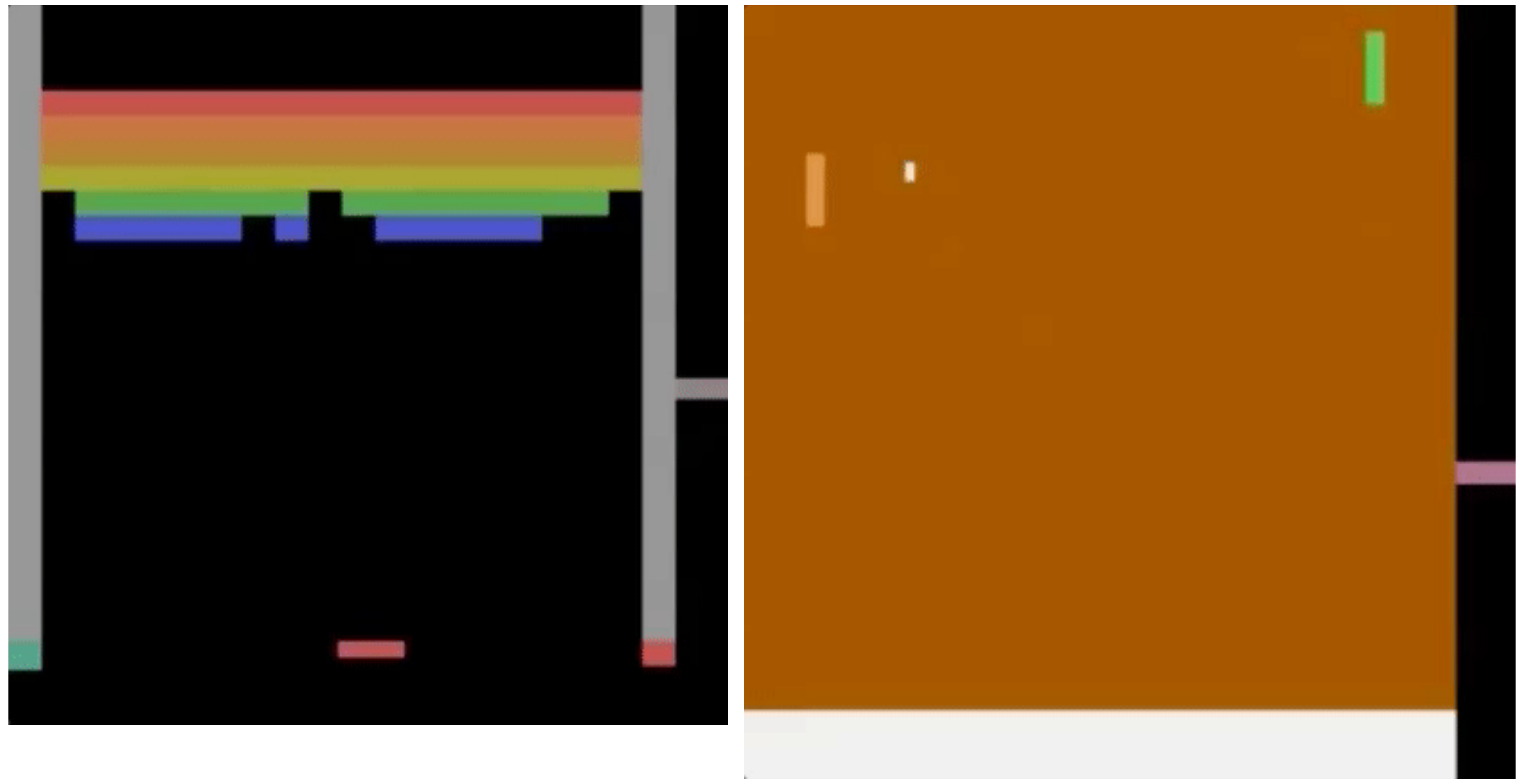
$$Q^{\pi}(s,a) = E_{\pi} \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k} | s_t = s, a_t = a \right\}$$

‣ **Bellman Equation:**

$$Q^*(s,a) = \sum_{s'} T(s,a,s') \left(R(s,a,s') + \gamma \max_{a'} Q^*(s',a') \right)$$

- **ATARI Games:**
  - State from pixels

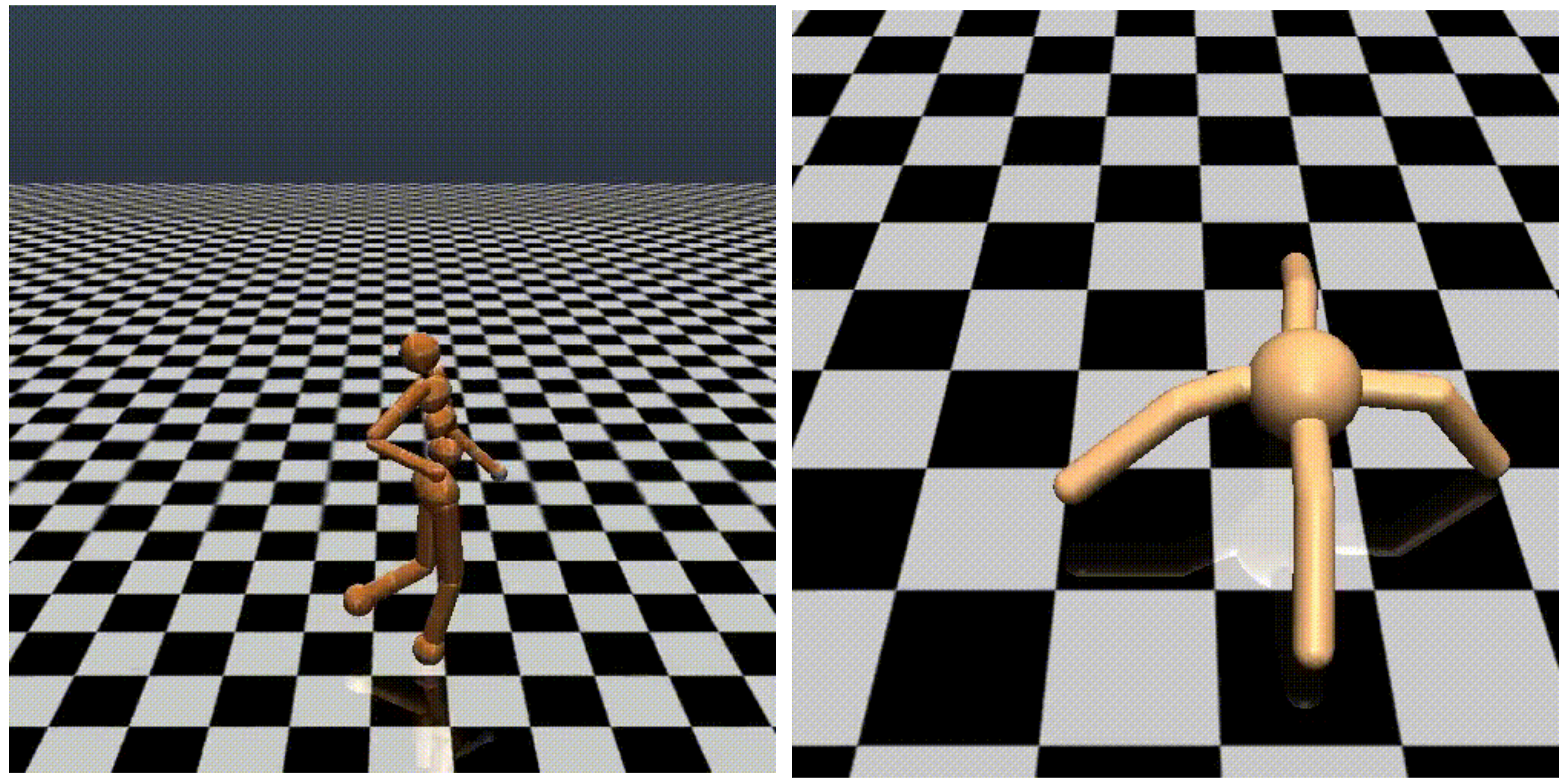
Mnih 2015



- **Chess and GO**
  - Breadth of game-state dimensionality

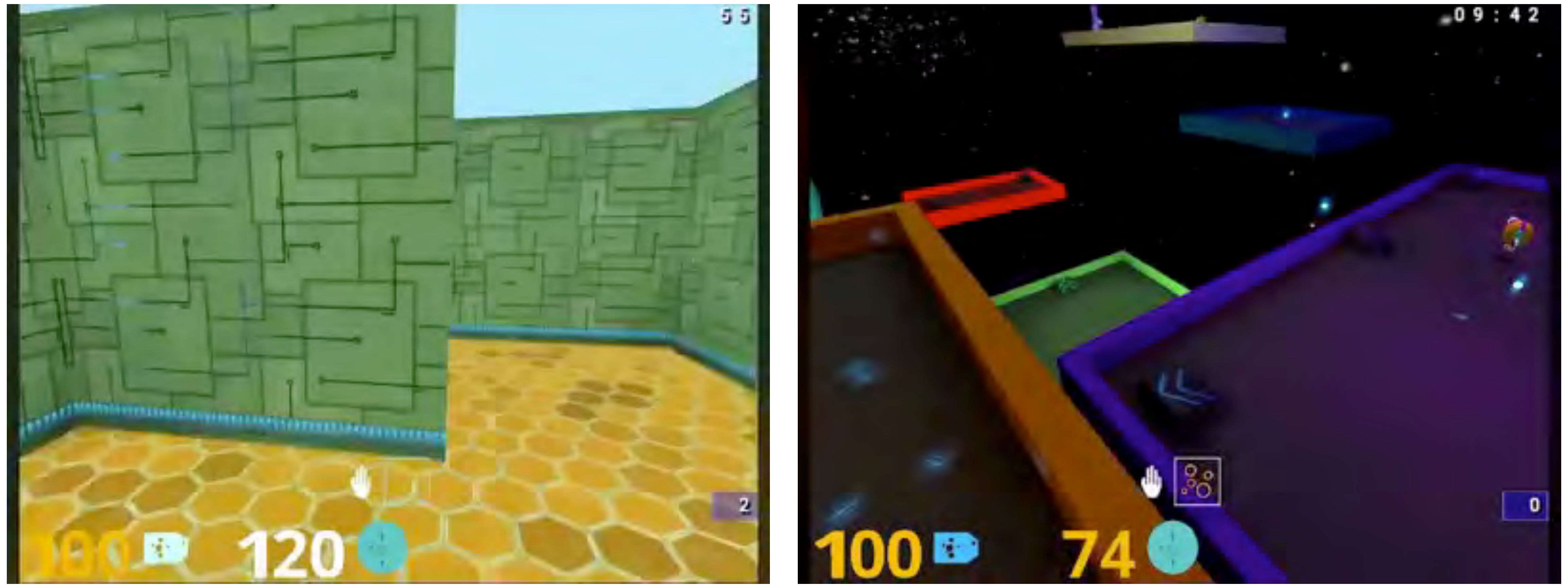
Silver 2016

- **Robotic Benchmarks:**
  - Continuous actions, Precise controls



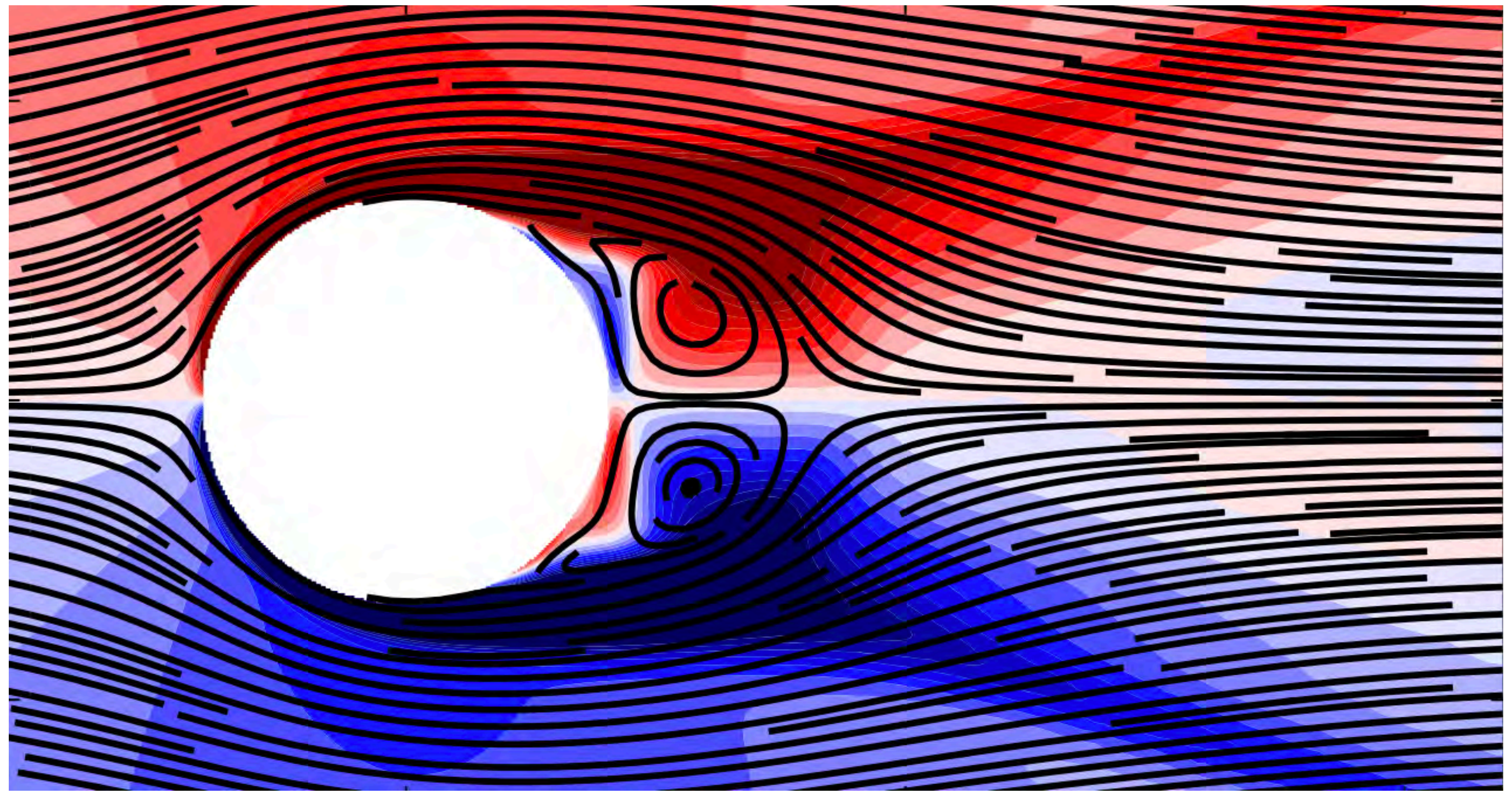
Schulman 2015

- **Advanced games**
  - Generality, Partial observability

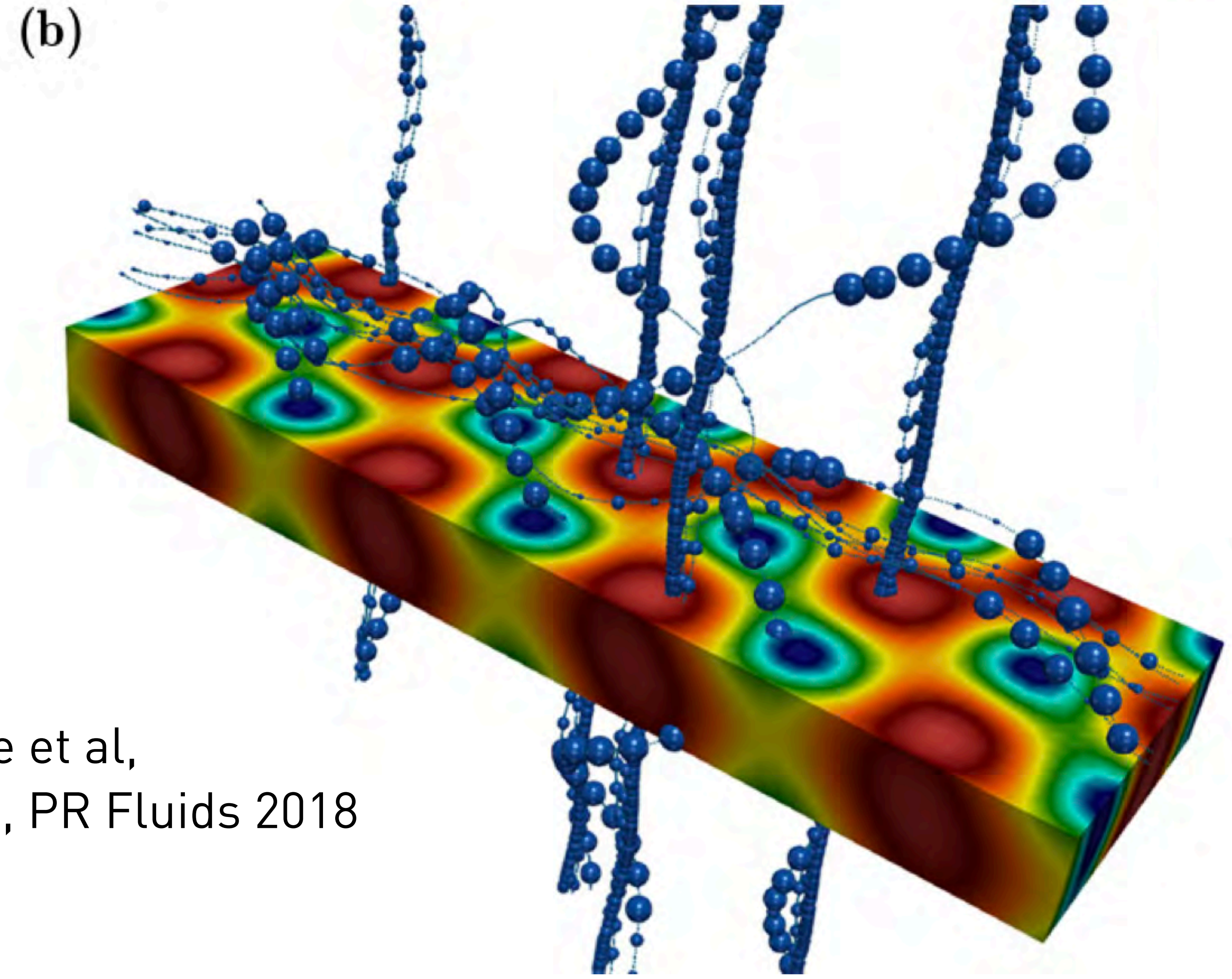


Jaderberg 2016

RL+ Fluids



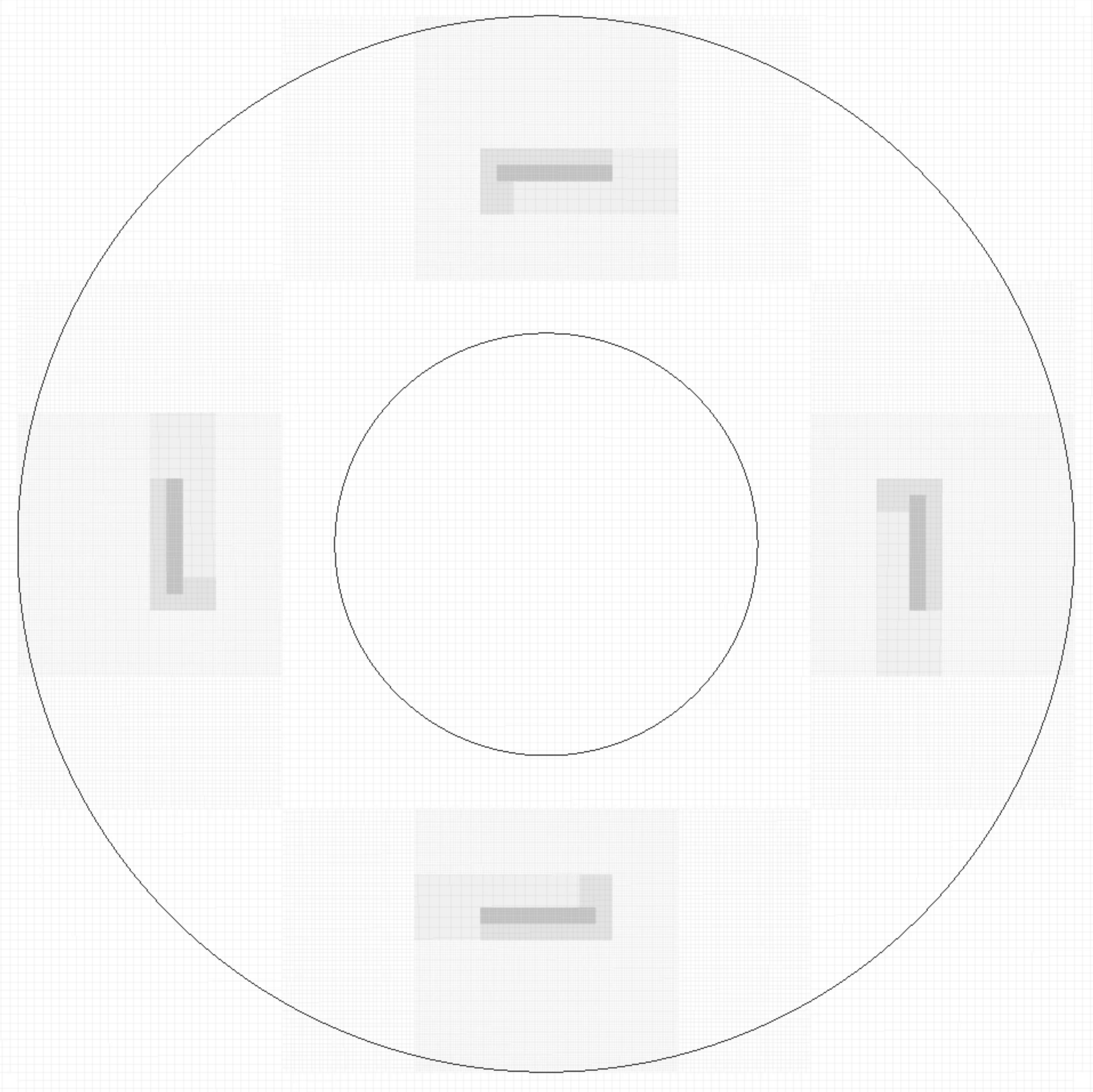
Gueniat et al,
Theor. Comp. Fluid Dyn., 2016



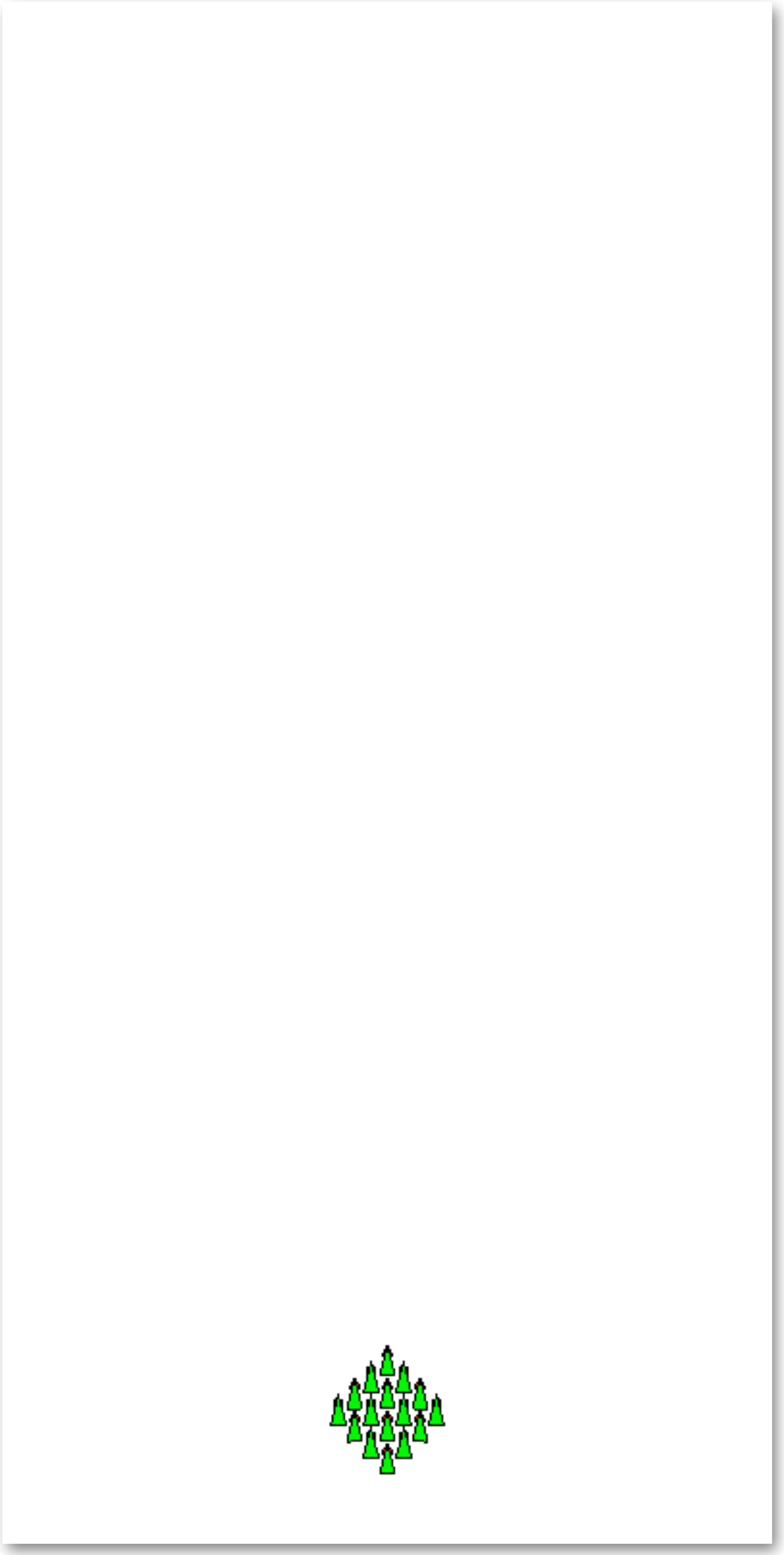
Colabrese et al,
PRL 2017, PR Fluids 2018

Reddy et al,
PNAS 2017, Nature 2018

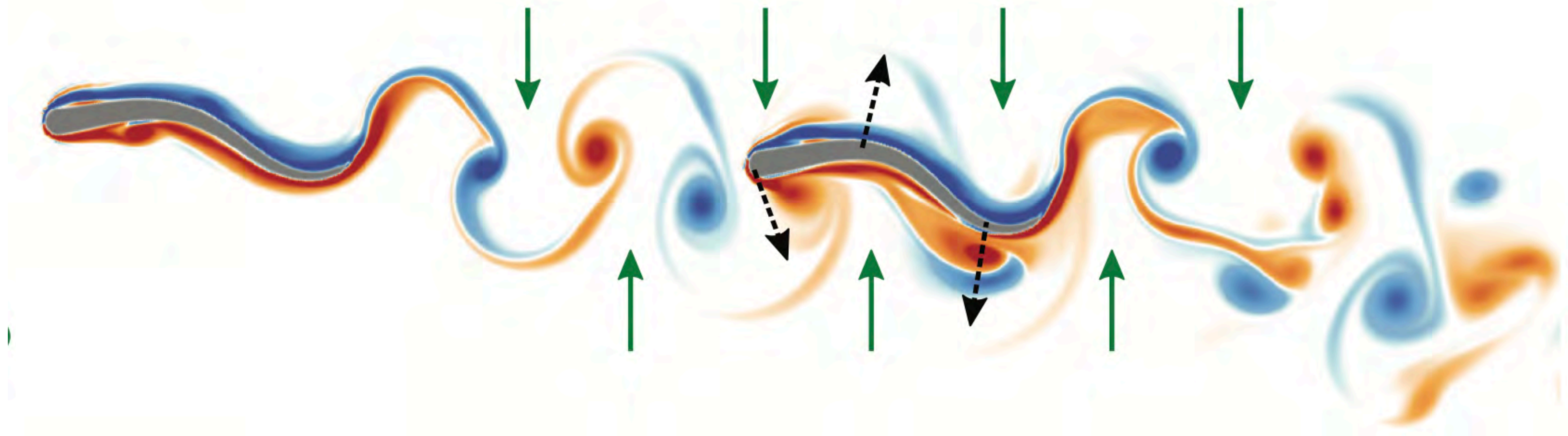
RL + Fluids @ ETHZ



Gazzola et al,
SIAM J. Sci. Comp., 2014

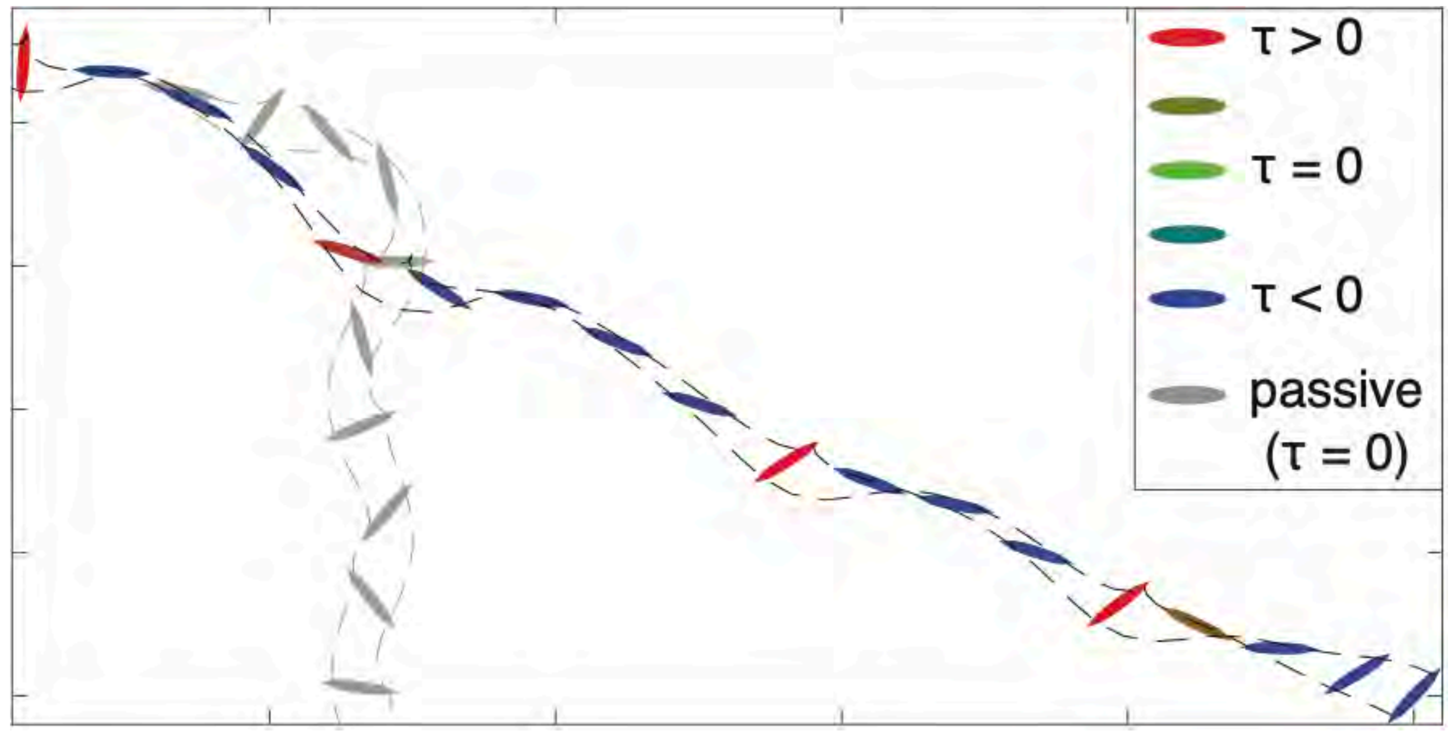


Gazzola et al,
J. Fluid Mech., 2016



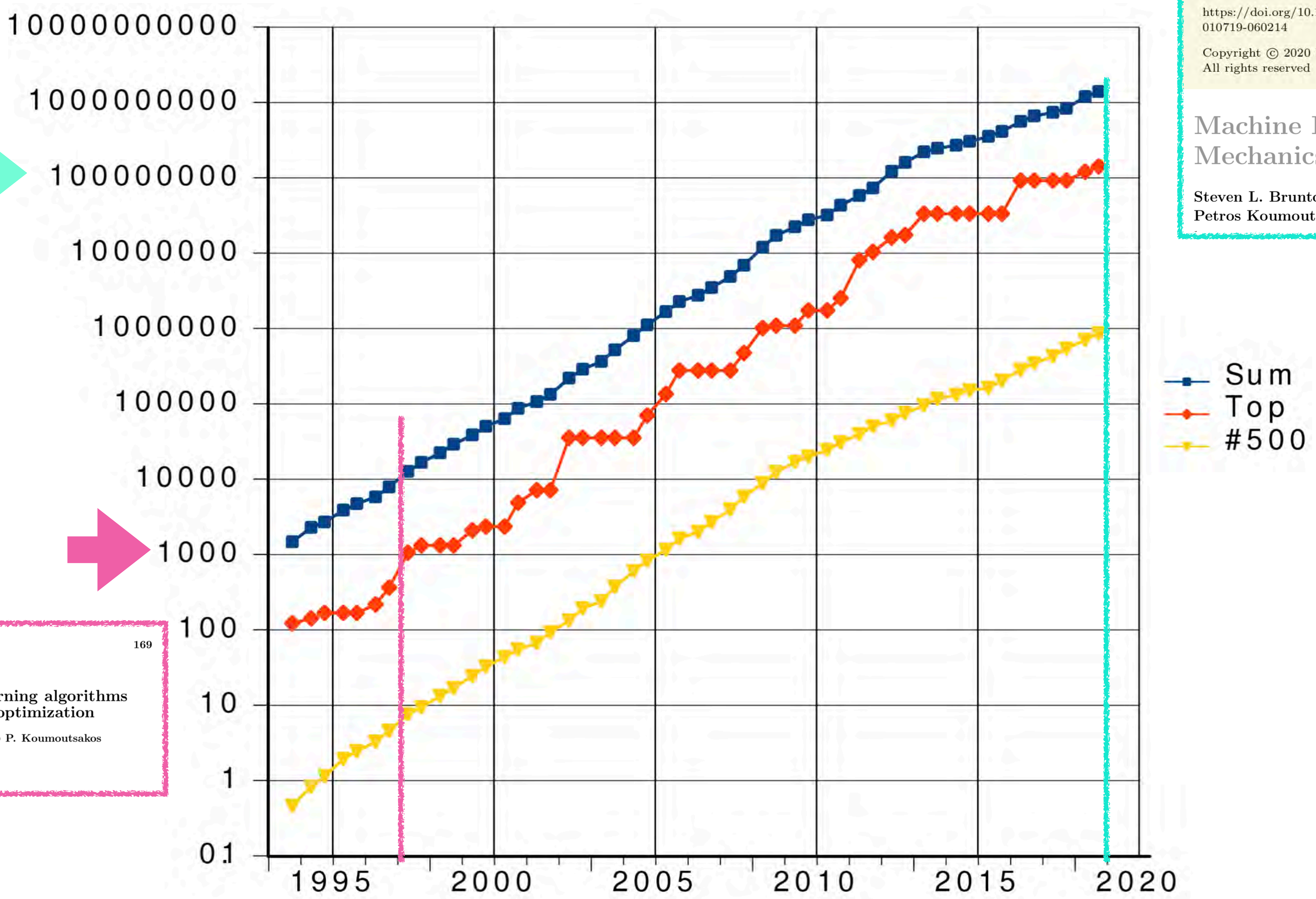
Novati et al,
Bioinsp. Biomim., 2017

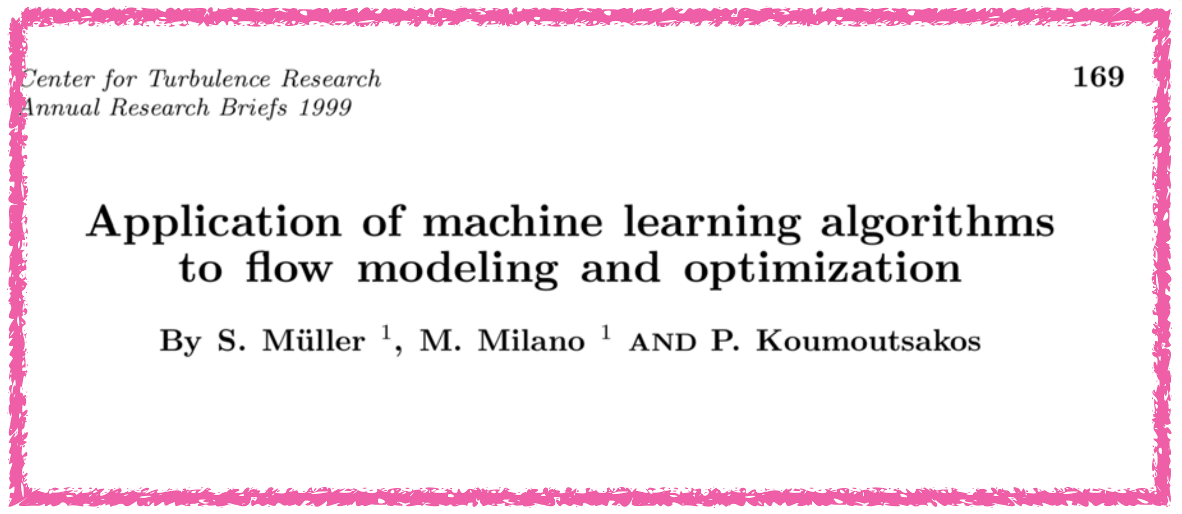
Verma et al,
PNAS, 2018



Novati et al,
PR Fluids, 2019

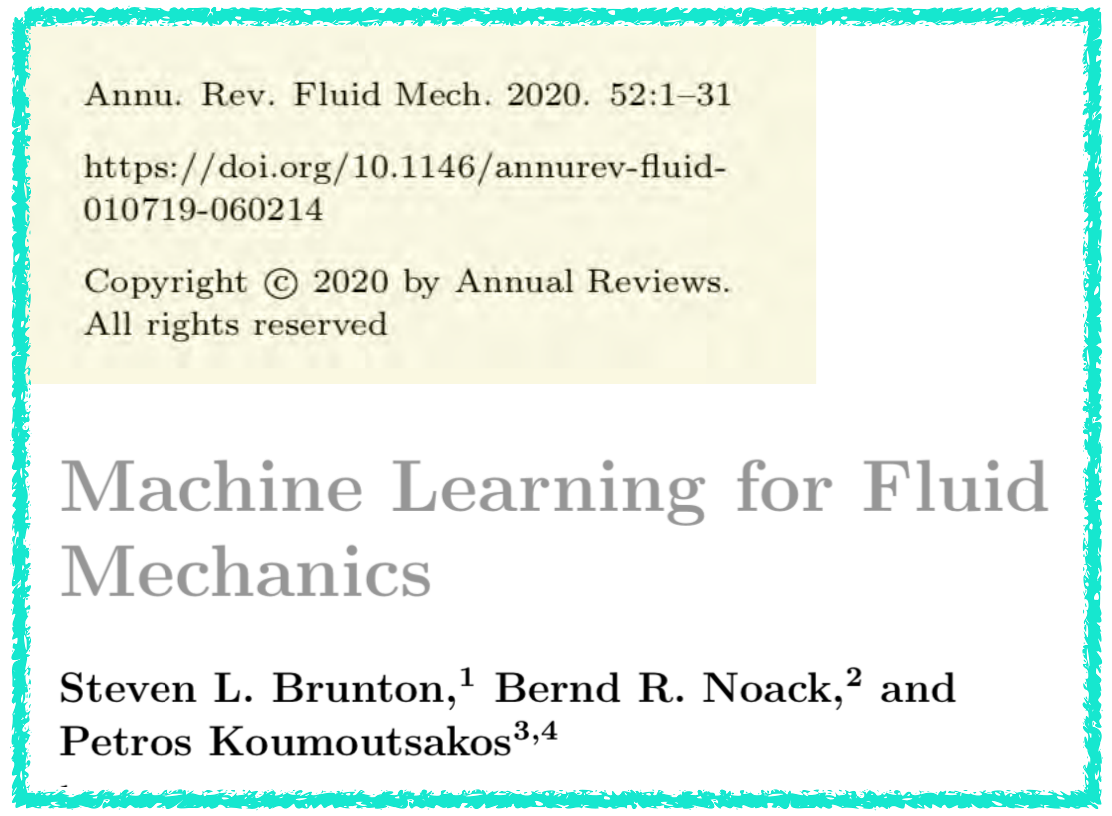
1Y in 1997 ~ 3' in 2019




Center for Turbulence Research
Annual Research Briefs 1999 169

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

Copyright © 2020 by Annual Reviews. All rights reserved

Machine Learning for Fluid Mechanics

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

Reinforcement Learning -Parametric policy

$$\pi_{\mathbf{w}}(a \mid s_t) := \mathcal{N}(a \,|\, \mu(s_t; \mathbf{w}), \sigma^2(s_t; \mathbf{w})) := \mathcal{N}(a \,|\, \mu_{\mathbf{w}}(s_t), \sigma_{\mathbf{w}}(s_t)^2)$$

$$s_0 \sim D_0(s)$$

Environment at random initial condition

$$a_0 \sim \pi_{\mathbf{w}}(a \mid s_0)$$

$$s_1,\ r_1 = D(s_0,\ a_0)$$

Following agent's action, environment advances in time and agent receives reward

$$\vdots$$

$$a_t \sim \pi_{\mathbf{w}}(a \mid s_t)$$

$$s_{t+1},\ r_{t+1} = D(s_t,\ a_t)$$

NOTE: Here assume a deterministic environment

GOAL: Define a cost function $J(\mathbf{w})$

identify optimal $\mathbf{w}$

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

$$\mathbf{w}^\star = \arg\max_{\mathbf{w}} J(\mathbf{w})$$

PARAMETRIZED POLICIES

$$\mathbf{a_t} \sim \pi(\mathbf{a}\,|\,\mathbf{s_t}\,;\mathbf{w}) \;=\; \pi_{\mathbf{w}}(\mathbf{a}\,|\,\mathbf{s_t})$$

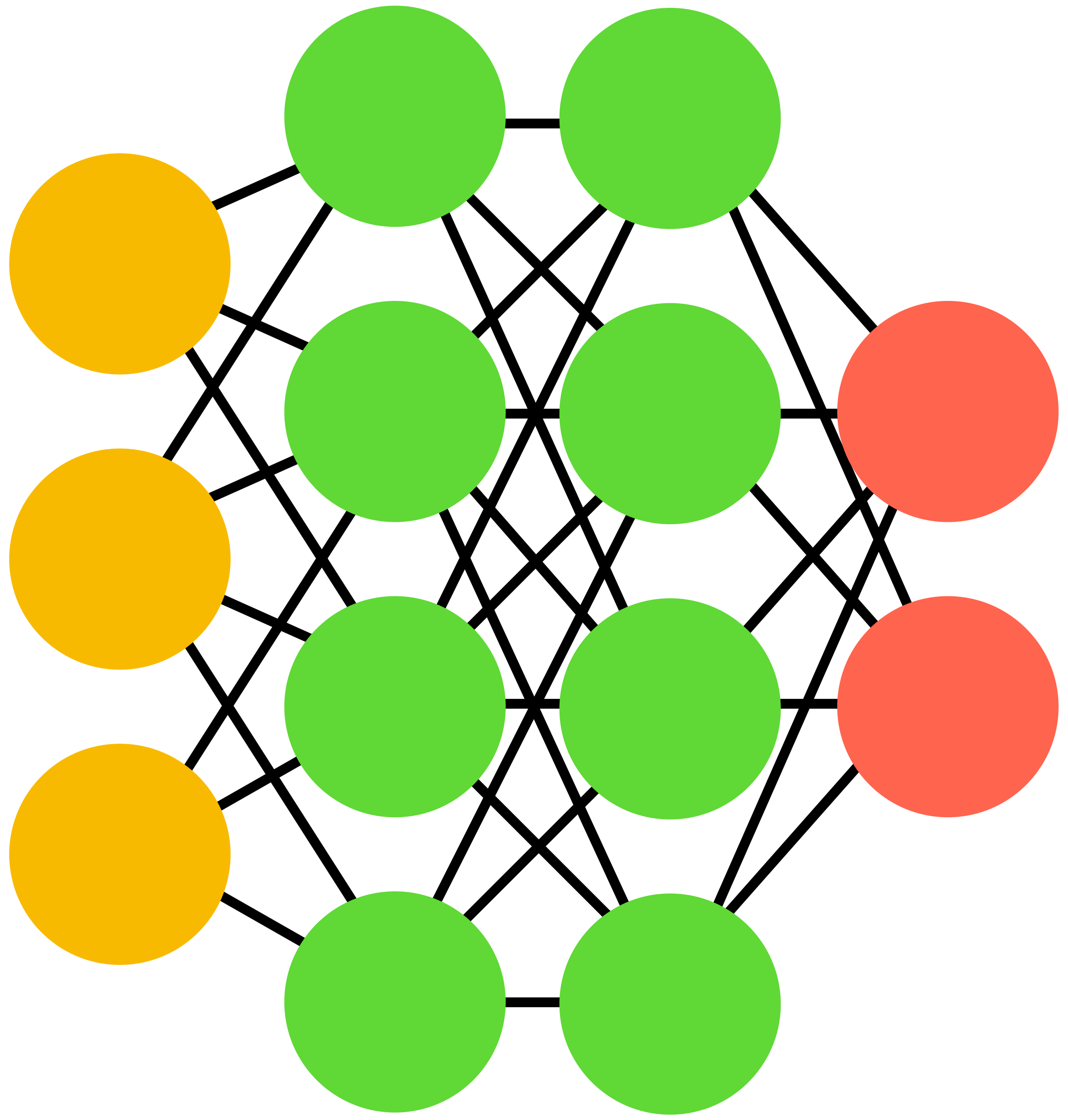
EXAMPLE: GAUSSIAN

$$\pi_{\mathbf{w}}(\mathbf{a} \mid \mathbf{s}_t) := \mathcal{N}(\mathbf{a}\,|\,\boldsymbol{\mu}(\mathbf{s}_t;\mathbf{w}), \boldsymbol{\sigma}^2(\mathbf{s}_t;\mathbf{w})\mathbf{I})$$

Neural Network to approximate

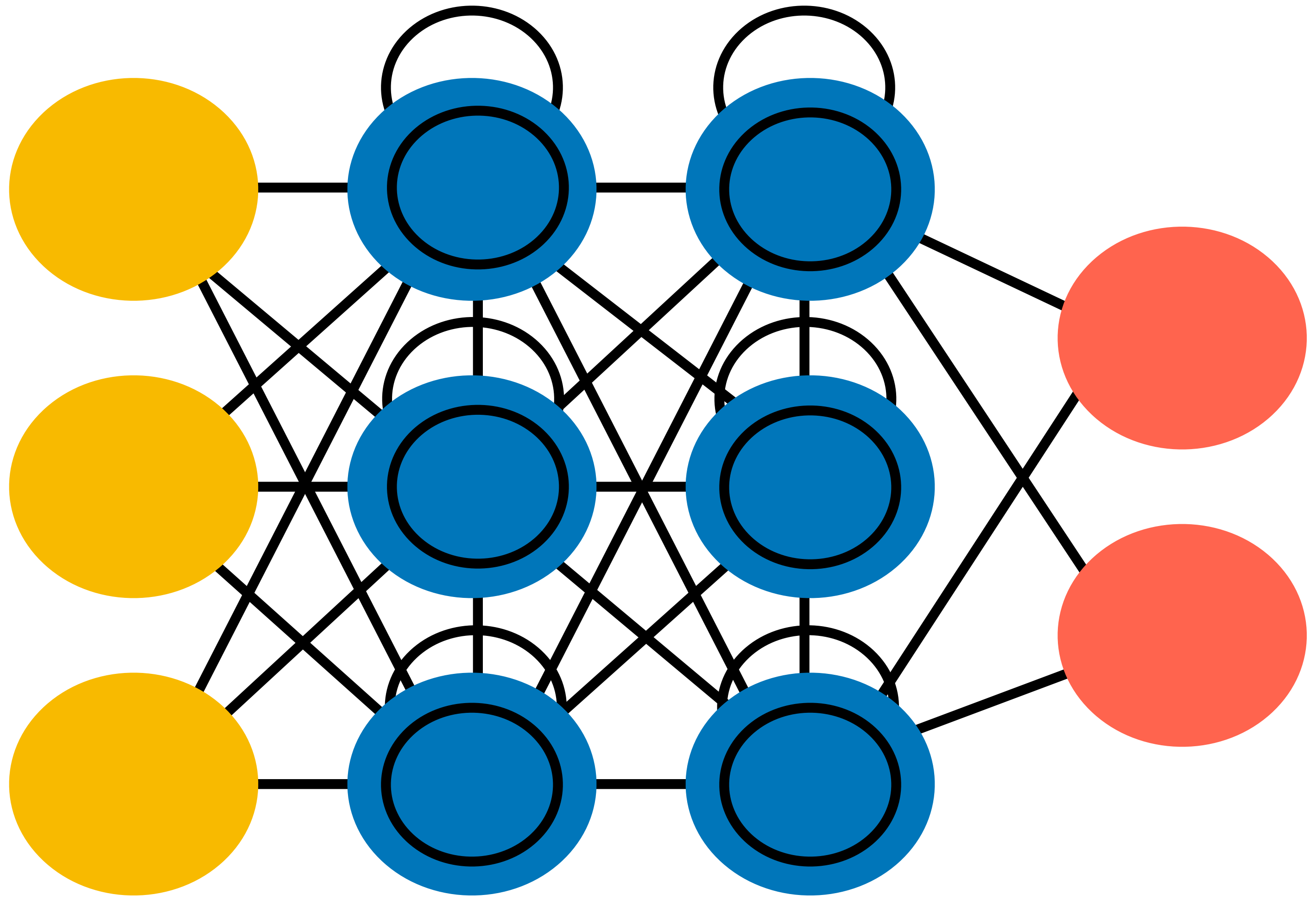
$$\mathbf{s}_t \to (\boldsymbol{\mu}, \boldsymbol{\sigma})$$

Deep Feed Forward Network:



for fully observable system

LSTM Recurrent Neural Network:



use temporal evolution for partially observable systems

Policy Update -> update the weights

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \lambda\,\mathbf{g}(\mathbf{w}_n)$$

How to chose $\mathbf{g}(\mathbf{w})$?

Stochastic Estimator

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \lambda\,\overline{\nabla J(\mathbf{w}_n)}$$

Policy Gradient Methods

$$J(w) := v_{\pi_w}(s)$$

(Sutton, '00)

Policy Gradient Update (Sutton, '00)

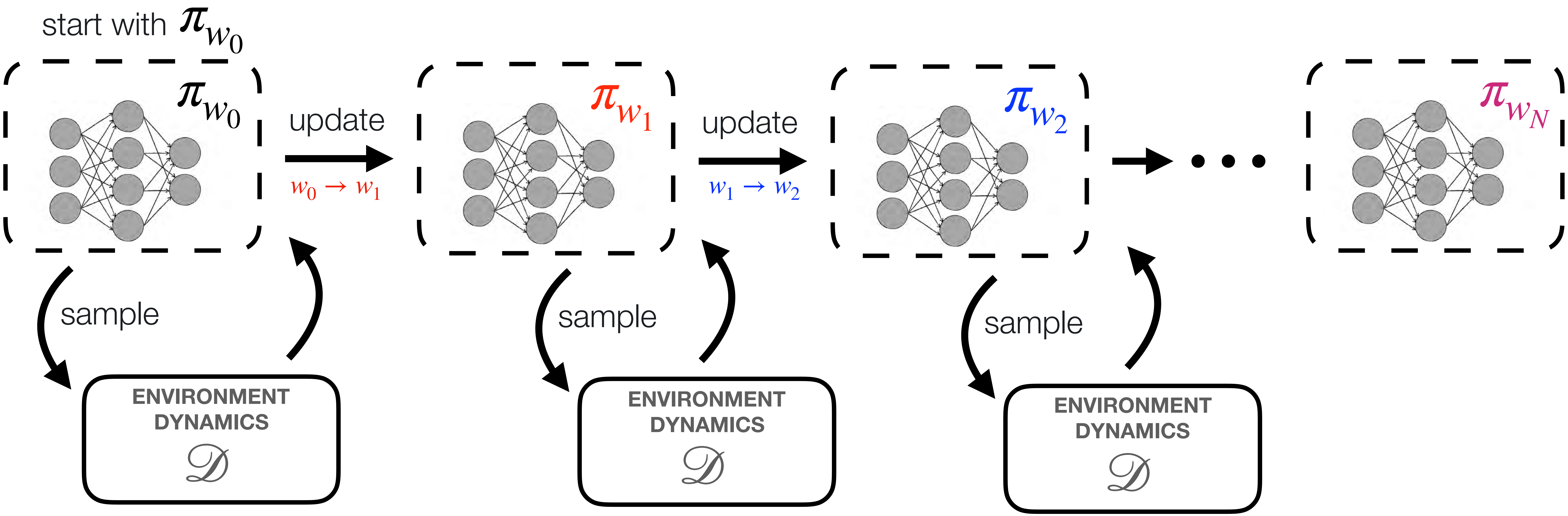
Let : $\boxed{J(w) := v_{\pi_w}(s)}$

Estimate gradient by sampling and taking an expectation over policy (Degris et. al., 2012)

$$\nabla_w J(w) = \mathbb{E}_{\pi_w}\Big[\sum_a \pi_w(a \mid s_t)\, q_{\pi_w}(s, a) \frac{\nabla_w \pi_w(a \mid s, w)}{\pi_w(a \mid s_t)}\Big]$$

$$q_{\pi_w}(s, a) = \mathbb{E}_{\pi_w}\Big[\sum_{t=0}^{\infty} \gamma^t\, r_{t+1} \mid (s_0 = s, a_0 = a, a_t = \pi_w(s_t))\Big]$$

EXPERIENCES IN REINFORCEMENT LEARNING - OFF POLICY LEARNING



Experiences: $\{\{s_t, r_t, \pi_{w_0}, a_t\}_{1:T}\}_{1:M}$

Experiences: $\{\{s_t, r_t, \pi_{w_1}, a_t\}_{1:T}\}_{1:M}$

Experiences: $\{\{s_t, r_t, \pi_{w_2}, a_t\}_{1:T}\}_{1:M}$

After updating $\mathbf{w}_n$, the data collected with policies $\pi_{\mathbf{w}_k}$ $k \in [n-1, n-2, \ldots.]$ are past experiences

How to utilize off policy/past experiences?

Reinforcement Learning and Flow Control

$$a_k \sim \pi_w(a \mid s_k)$$

- Policy: Best action for current state to maximize long term reward

IN FLOW CONTROL

- **Transitions can be sampled but may not be known analytically**
- **Policy/Transitions are not stationary**
- **Samples are expensive to evaluate**

- **Policy: How to Balance Exploration and Exploitation**
- **How to use Memories and Experiences ?**
- **...**

SAMPLING ONLINE POLICY ->EXPLORE —> EXPENSIVE to EXPLOIT

How to reduce the computational cost of RL and its online sampling ?

Economize by using existing samples -> use **memory** of the system

EXPERIENCE REPLAY: Store subset of experiences, and"replays" them offline, learning anew from past successes/failures , (Long-Ji Lin, 1992)

Experience Replay is critical to maximizing data efficiency, avoids the destabilizing effects of learning from consecutive correlated experiences, and allows the network to learn a viable value function even in **complex, highly structured sequential environments such as video games.** (Hassabis et. al. , 2017, Neuron Review)

SAMPLING ONLINE POLICY—> EXPENSIVE

(each sample a simulation)

EXPERIENCE REPLAY (Long-Ji Lin, 1992): Store subset of experiences, and "replays" them offline, learning anew from past successes/failures

EXPERIENCE REPLAY -> IMPORTANCE SAMPLING
Change of probability distribution for computing Expectations

BUT
how good are the experiences ?

IMPORTANCE SAMPLING WITH EXPERIENCES

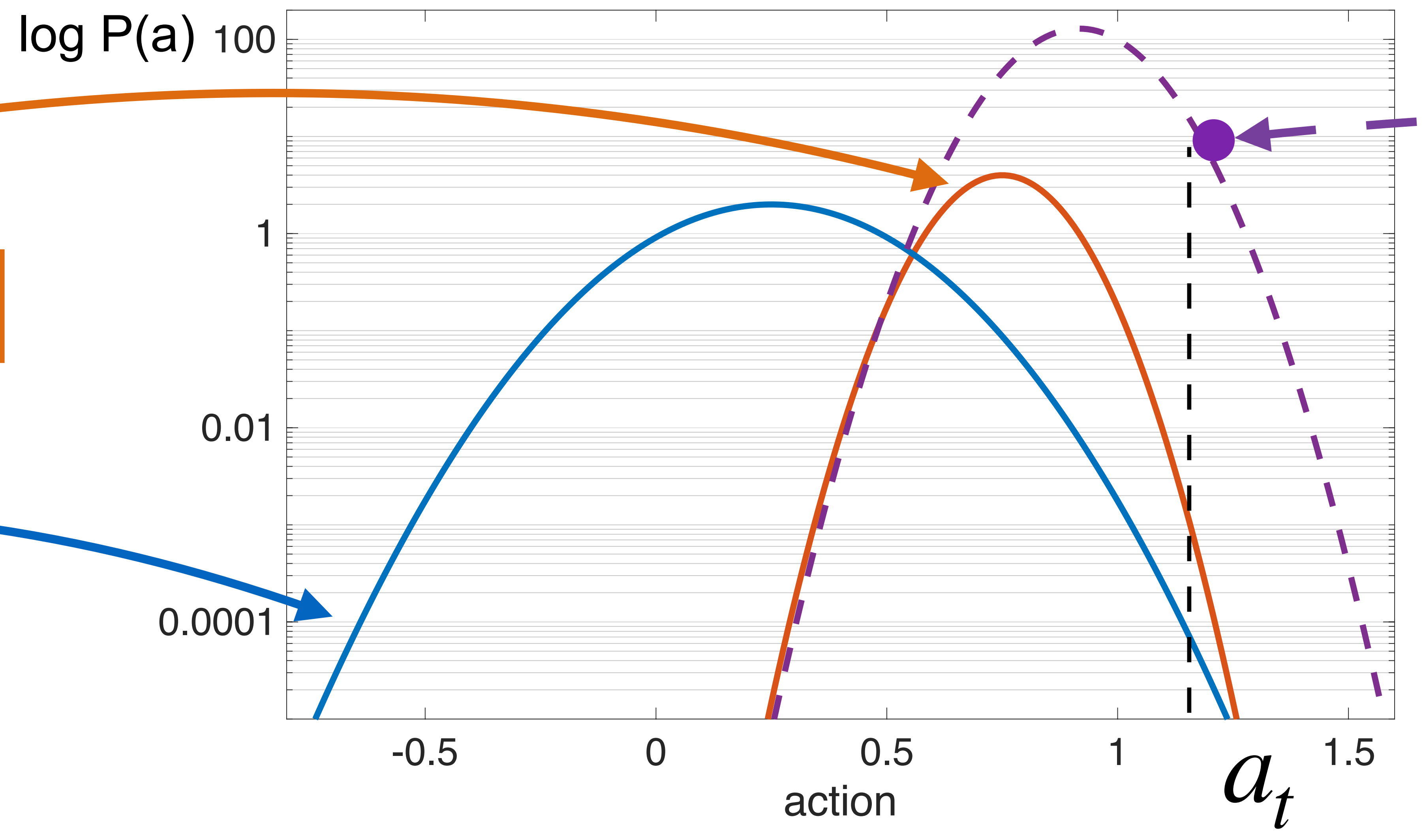
ISSUE #1

$\pi_w(a \mid s_t)$

Current policy that we aim to update

$\mu_t(a \mid s_t)$

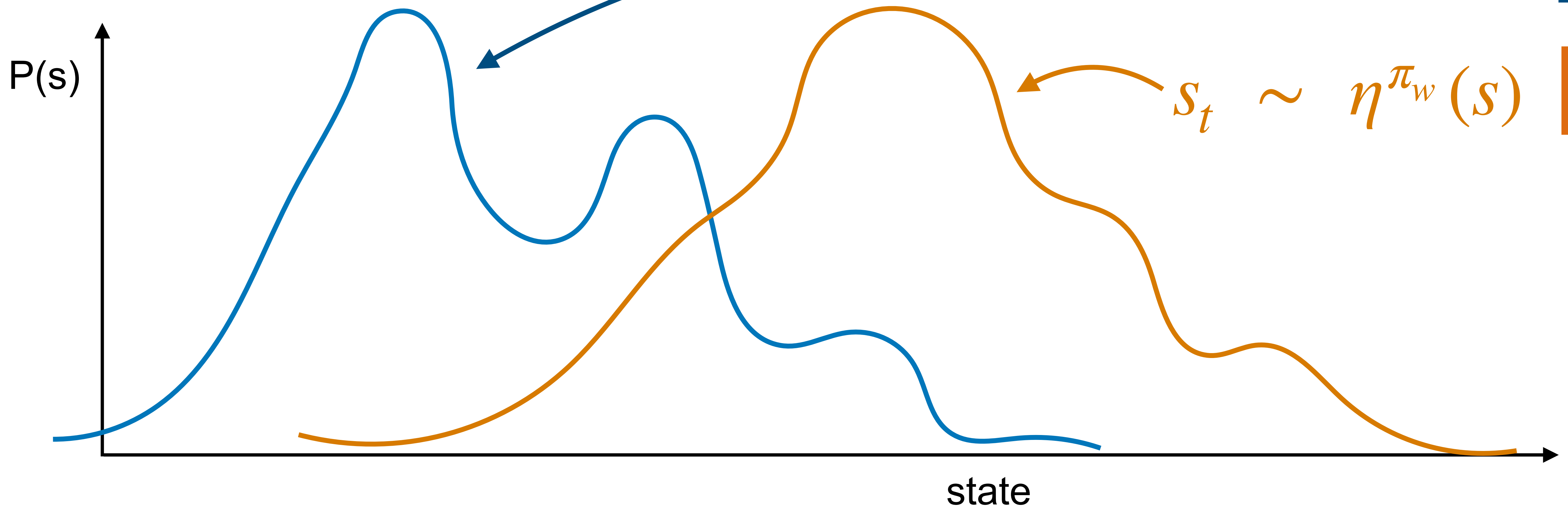
old policy experienced by agent (training data)



$$\rho_{w,t}(s_t, a_t) = \frac{\pi_w(a_t \mid s_t)}{\mu_t(a_t \mid s_t)}$$

Unbounded (0 to ∞) Importance weight, increases the variance and **lowers the accuracy of the cost function** $\tilde{J}(w)$.

ISSUE #2

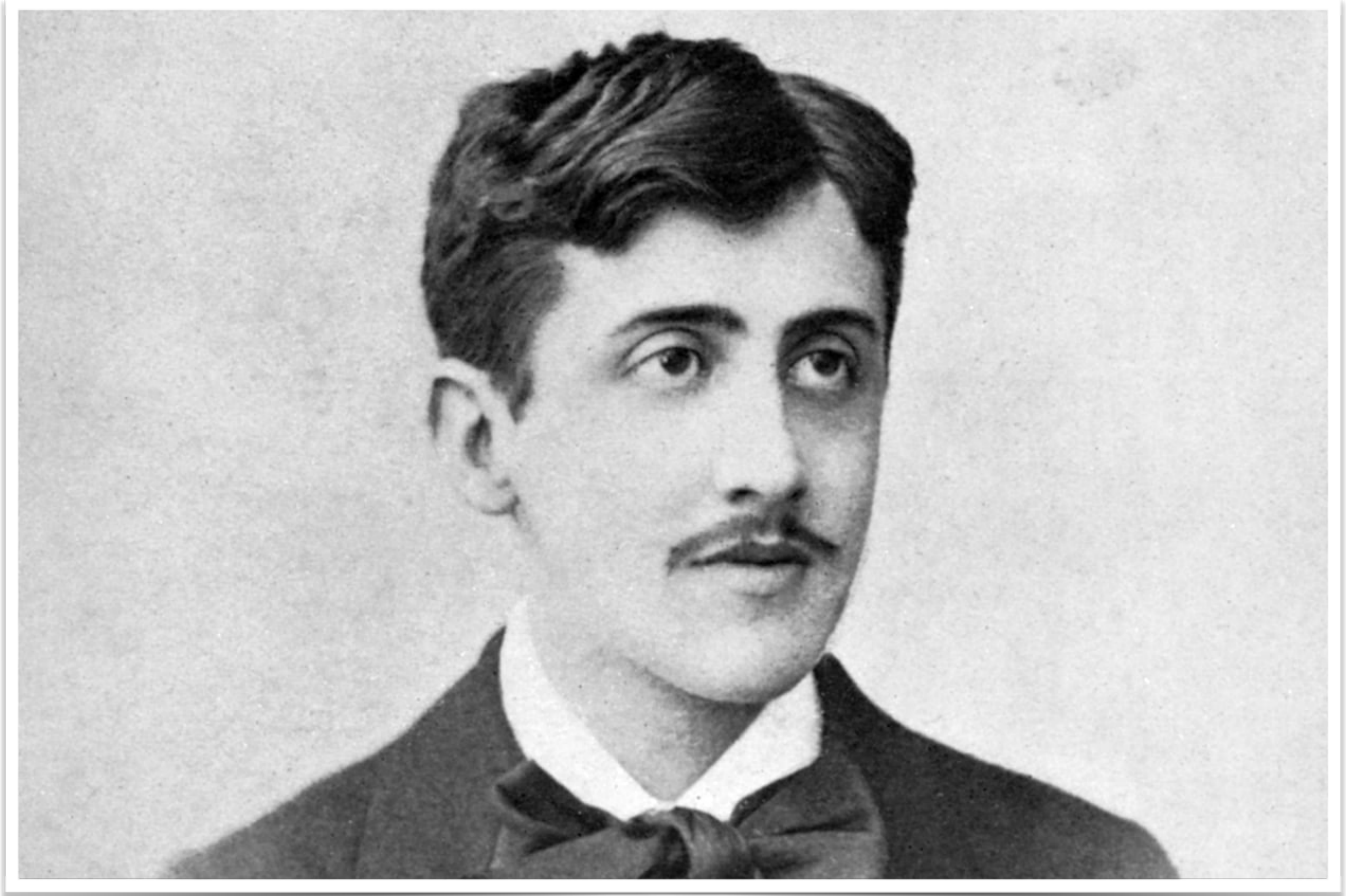


$s_t \sim \eta^{\mu_t}(s)$ state with old policies (training)

$s_t \sim \eta^{\pi_w}(s)$ state with current policy

If distribution of training data is too **dissimilar from on-policy** outcomes, data may be **irrelevant to updating the policy**

Proposed solution: constrain policy changes to past policies




MARCEL PROUST
À LA RECHERCHE DU TEMPS PERDU
Tome X
Sodome et Gomorrhe
BIBEBOOK

Remember and Forget Experience Replay

I. Reject samples if importance weight π_w / μ_t lies outside of a ***trust region***.

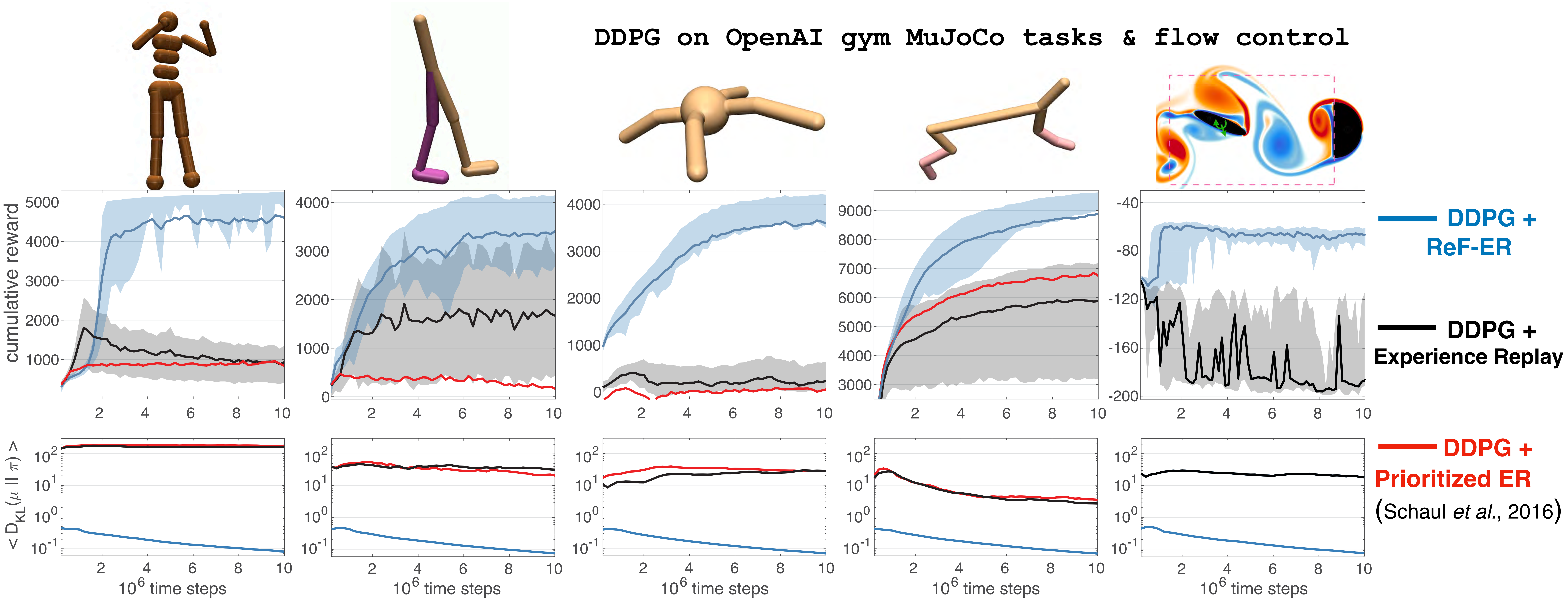
II. *Penalization, based on* **KL divergence**, attracts policy back towards training behaviors.

(Novati & Koumoutsakos, ICML '19

ReF-ER works with most RL methods that learn a policy by Experience Replay (e.g. DPG, NAF, ACER, SAC, ...)

Results

- ReF-ER with: Off-policy pol.-gradients (ACER, Wang *et al.* 2017), Q-learning (NAF, Gu *et al.* 2016), DPG (DDPG, Lillicrap *et al.* 2016).
- We observe: **effectively constrained D_{KL}, increased stability and performance**.
- At the price of: sometimes slower progress at the beginning of training.



arXiv.org > cs > arXiv:1502.05477

Computer Science > Machine Learning

Trust Region Policy Optimization

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, Pieter Abbeel

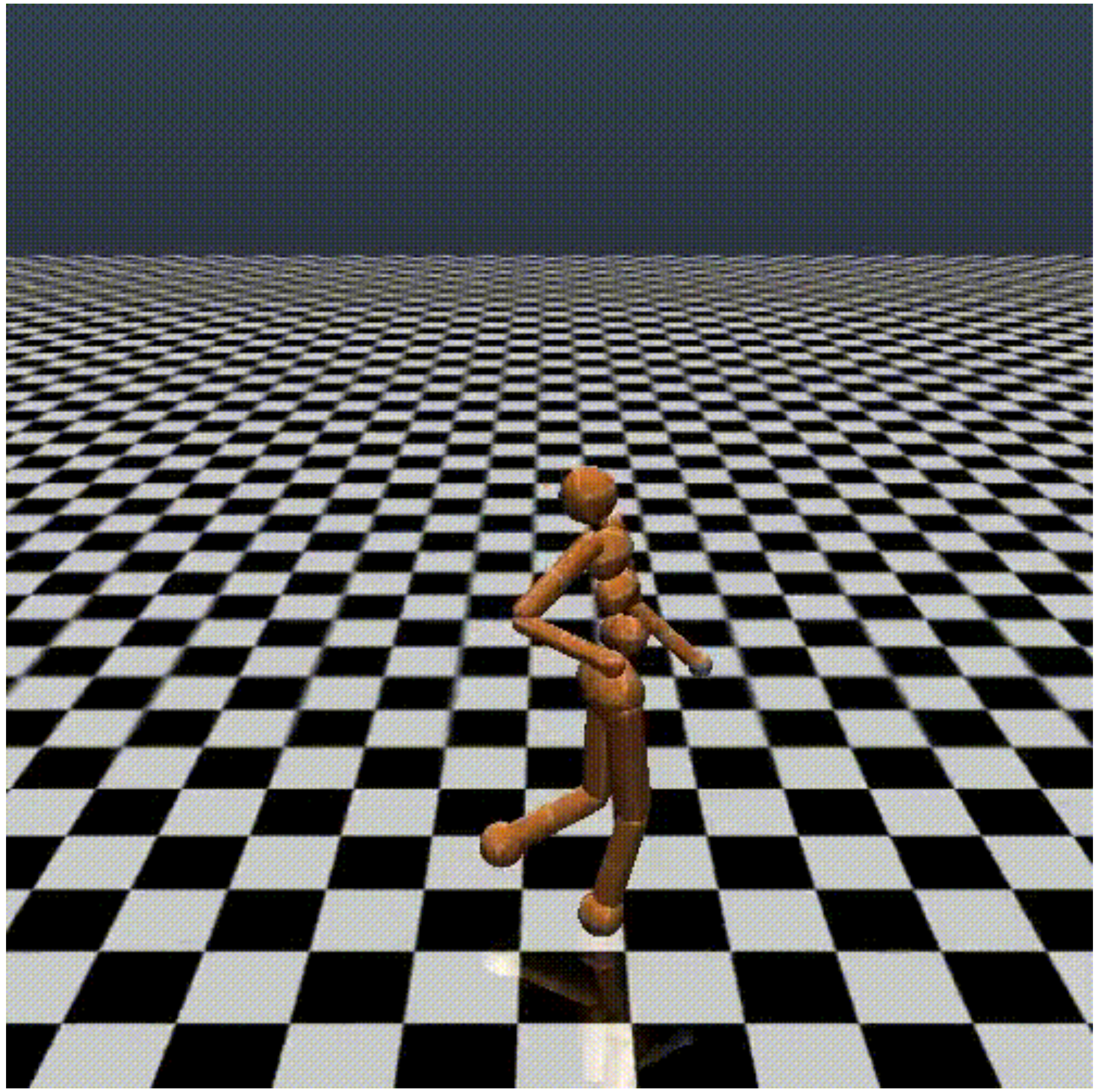
Humanoid Benchmark

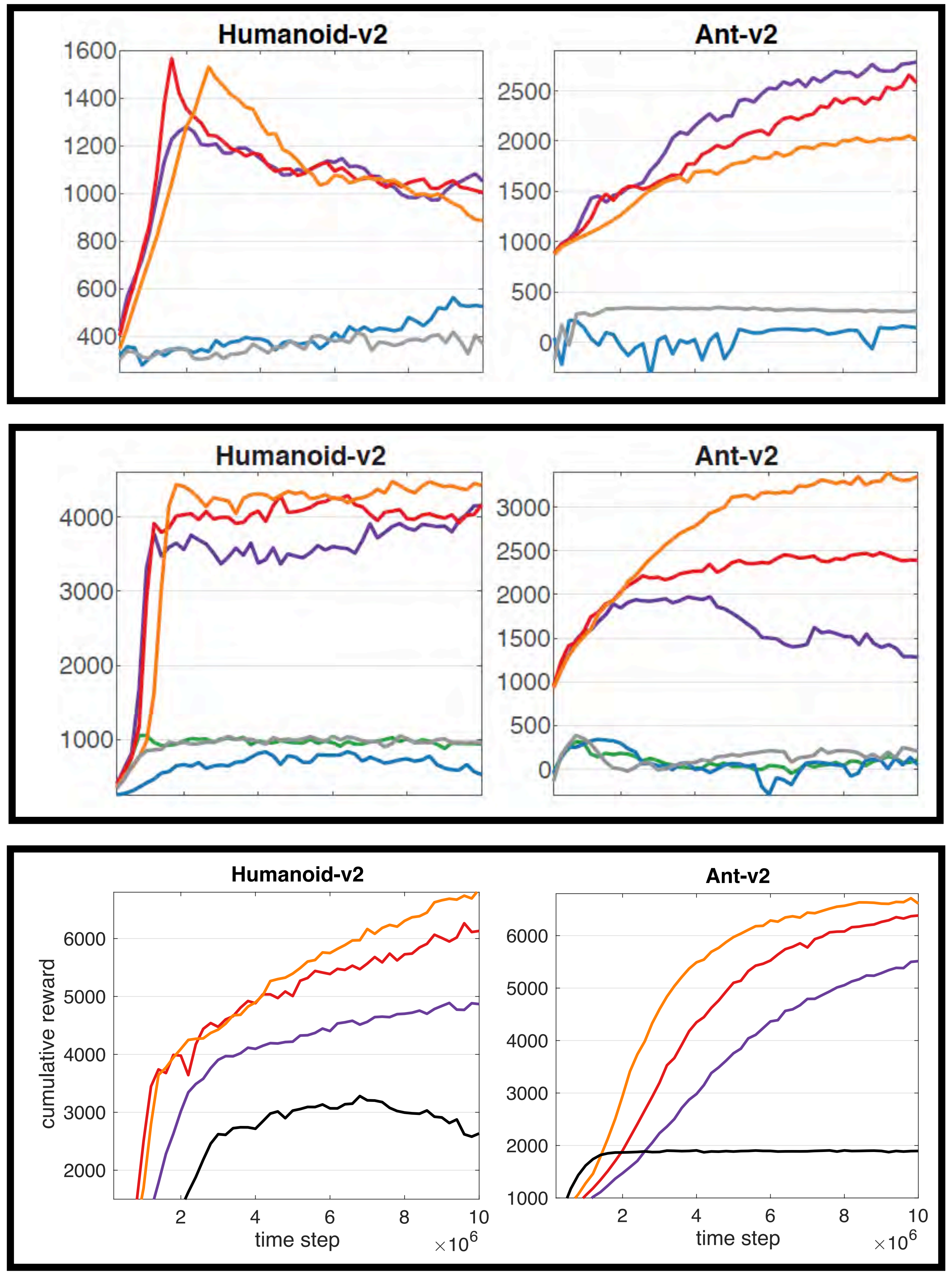
arXiv.org > cs > arXiv:1807.05827

Computer Science > Machine Learning

Remember and Forget for Experience Replay

Guido Novati, Petros Koumoutsakos





Legend:
ReF-ER with C=2, C=4, C=8
Vanilla ER, Prioritized ER, PPO

Reinforcement Learning
vs
Optimal Control

with L. Mahadevan (Harvard U.)

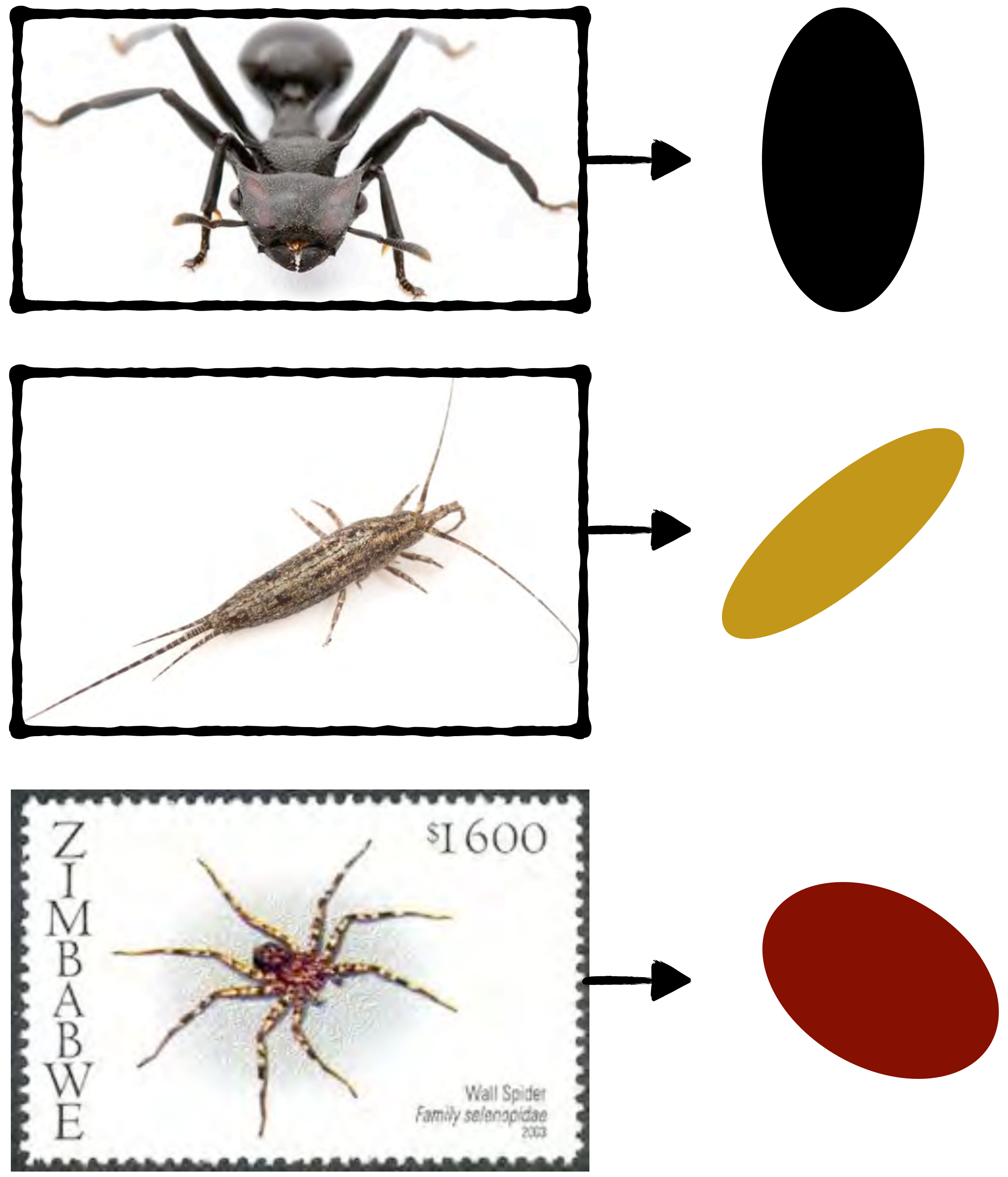

ZIMBABWE
$1600
Wall Spider
Family selenopidae
2003

Model of gliding-Arthropod

Gravity-driven motion of **ellipse** (Lamb, 1932)

$$
\begin{aligned}
(I+\beta^2)\dot{u} &= (I+1)vw - \Gamma v - \sin\theta - F\\
(I+1)\dot{v} &= -(I+\beta^2)uw + \Gamma u - \cos\theta - G\\
\tfrac{1}{4}[I(1+\beta^2)+\tfrac{1}{2}(1-\beta^2)^2]\dot{w} &= (\beta^2-1)uv + \tau - M\\
\dot{x} &= u\cos\theta - v\sin\theta\\
\dot{y} &= u\sin\theta + v\cos\theta\\
\dot{\theta} &= w.
\end{aligned}
$$

- Augmented with a **control torque** (Paoletti 2011) rotating limbs /moving centre of mass
- Ability to exert **torque is constrained** ($|\tau|<1$)



- Dynamics characterized by:
  - the **aspect ratio** $\beta = b / a$
  - the **density ratio** $\rho^* = \rho_s / \rho_f$
- Closure with **model of fluid forces** validated through experiments and simulations by Wang et al 2004-06

Shaping the reward

- **Objective: time/energy-optimal trajectories - compare with optimal control**

$$c_{t,\ \mathrm{time}} = \int_t^{t+\Delta t} \mathrm{d}t = \Delta t \qquad c_{t,\ \mathrm{energy}} = \int_t^{t+\Delta t} \tau^2 \mathrm{d}t = \tau^2 \Delta t$$

- RL task designed to nudge system towards desired behavior only through rewards
  - sample initial state $s_0 \sim \{U(-10,10), 0, 0, 0, 0, 0\}$
  - **rewards** must nudge towards x_G = 100 and penalize time/energy:

$$r_t = -c_t + \|x_G - x_{t-1}\| - \|x_G - x_t\|$$

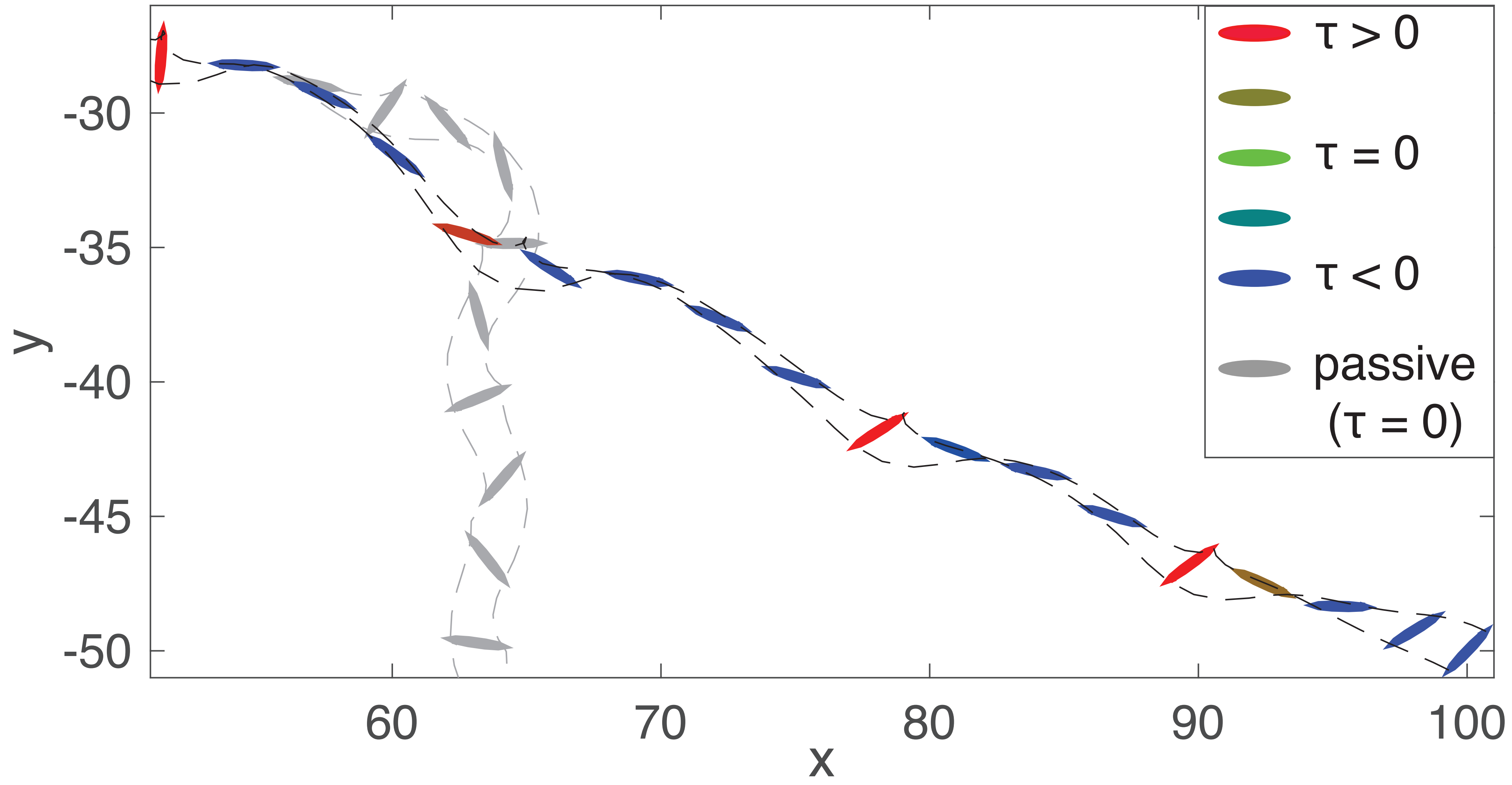
terminal reward: $$r_T = \|x_G - x_T\| + K\left(e^{-(x_G - x_T)^2} + e^{-10(\theta_G - \theta_T)^2}\right)$$

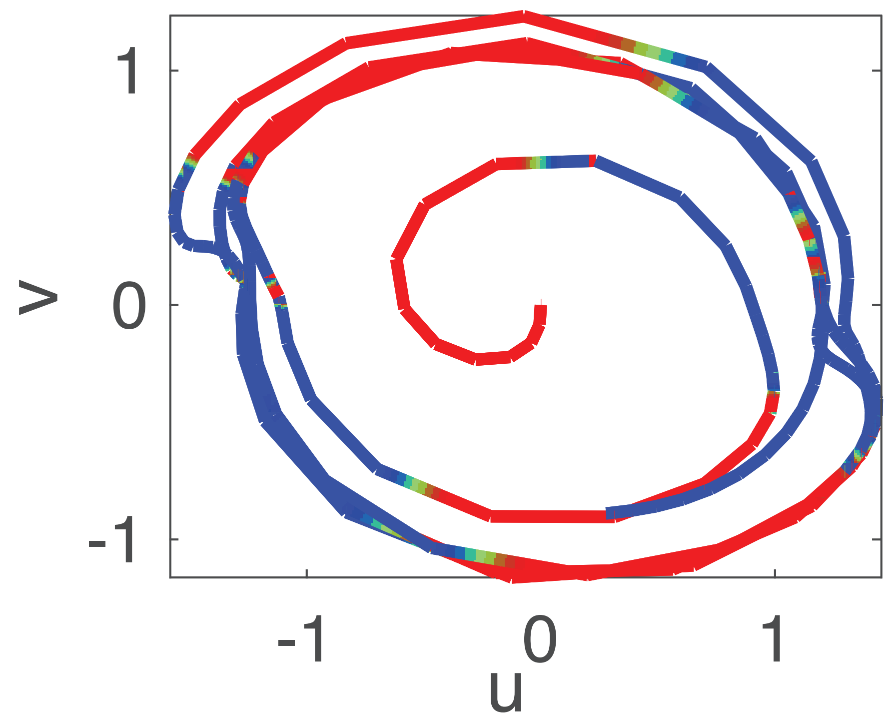
- Note that the sum of rewards along any trajectory that monotonically moves toward x_G:

$$\sum_{t=0}^{T} r_t = \underbrace{-\sum_{t=0}^{T} c_t}_{\text{total cost}} + \underbrace{\|x_G - x_0\|}_{\text{baseline, no effect on policy}} + \underbrace{K\left(e^{-(x_G - x_T)^2} + e^{-10(\theta_G - \theta_T)^2}\right)}_{\text{bonus for reaching } x_G,\ \theta_G}$$

Two emerging gliding strategies

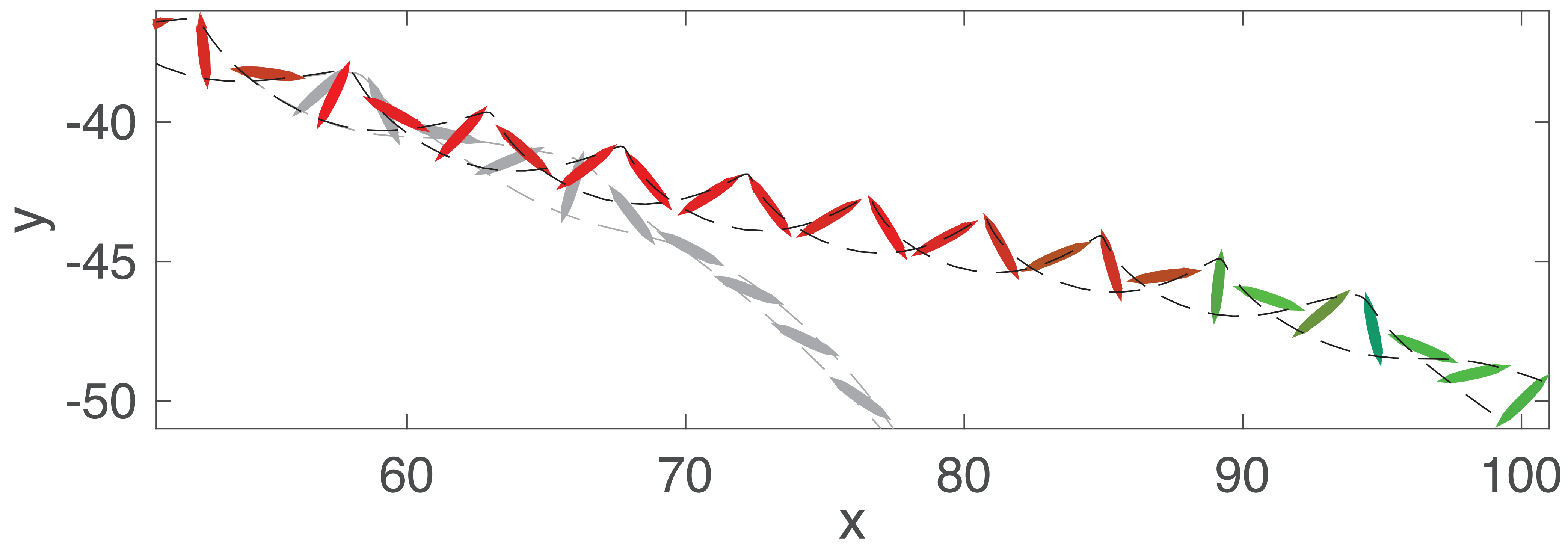
Bounding flight: time optimal β=0.1 ρ*=50

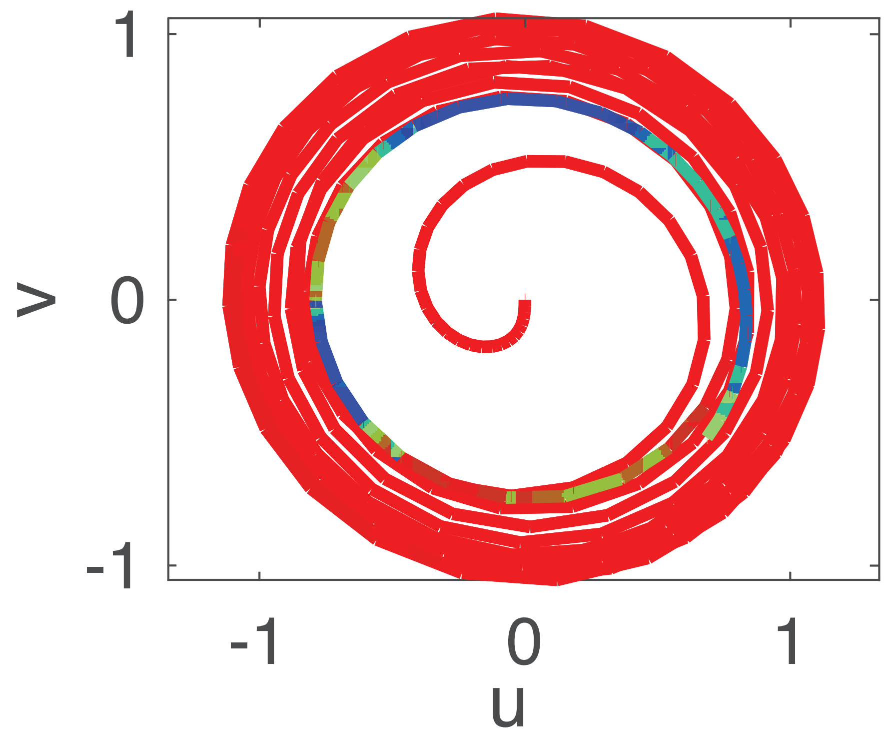




- Bounding flight owes name to energy-saving flight pattern employed by birds
- Composed of 2 phases:
  - positive torque inducing tumbling to generate lift
  - negative torque to glide while maintaining small angle of attack

Tumbling flight: energy optimal β=0.1 ρ*=200

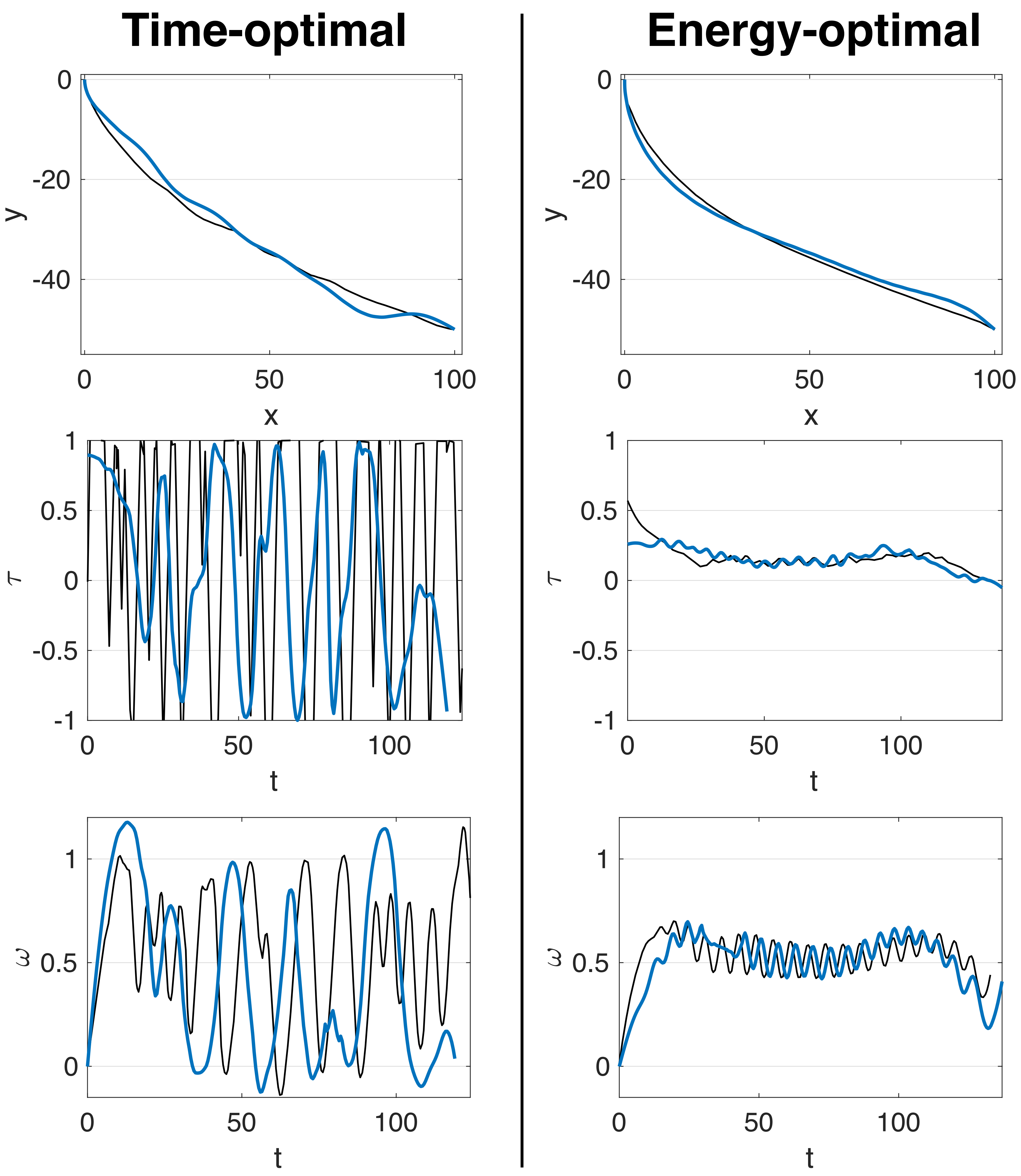




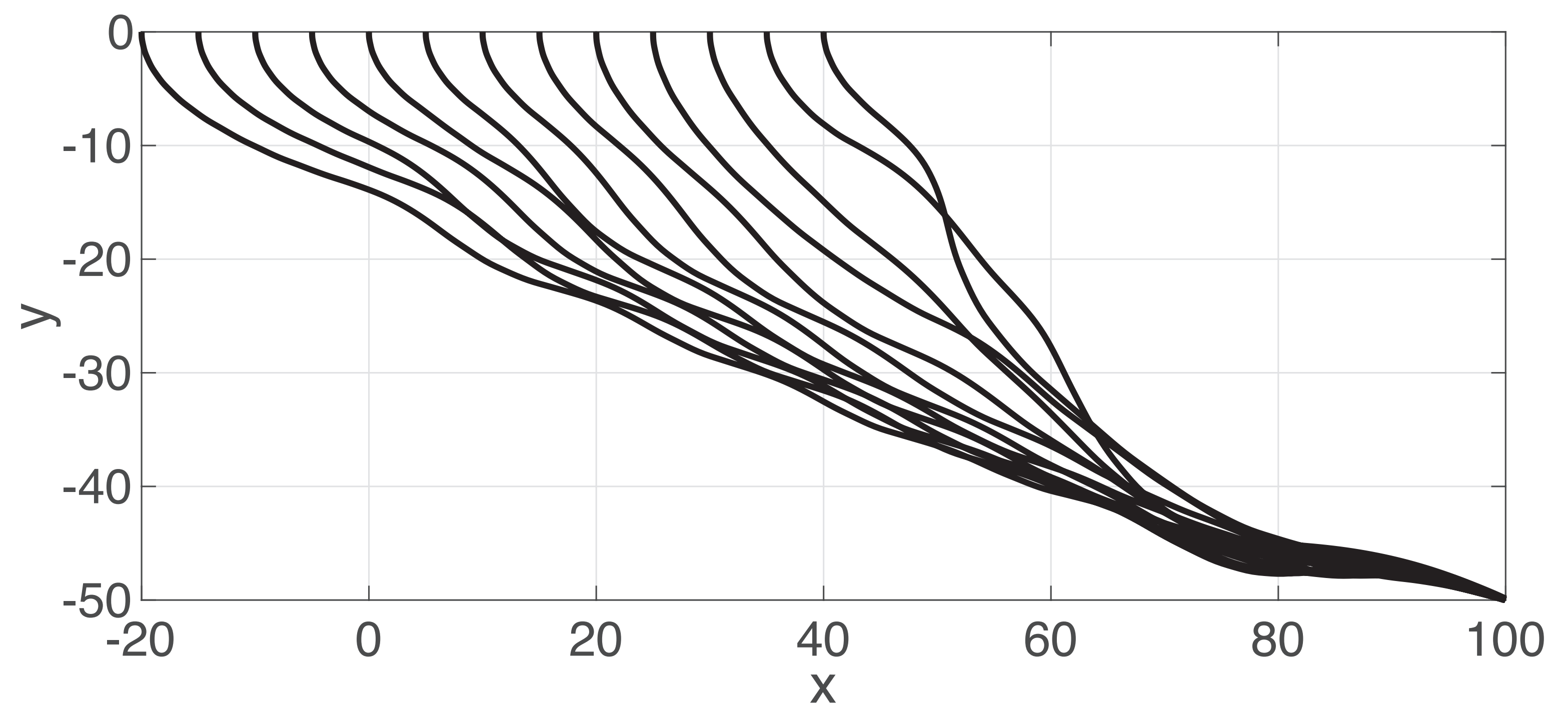
- Tumbling flight consists of:
  - maintaining an almost-constant minimal torque until landing
  - continuous generation of lift

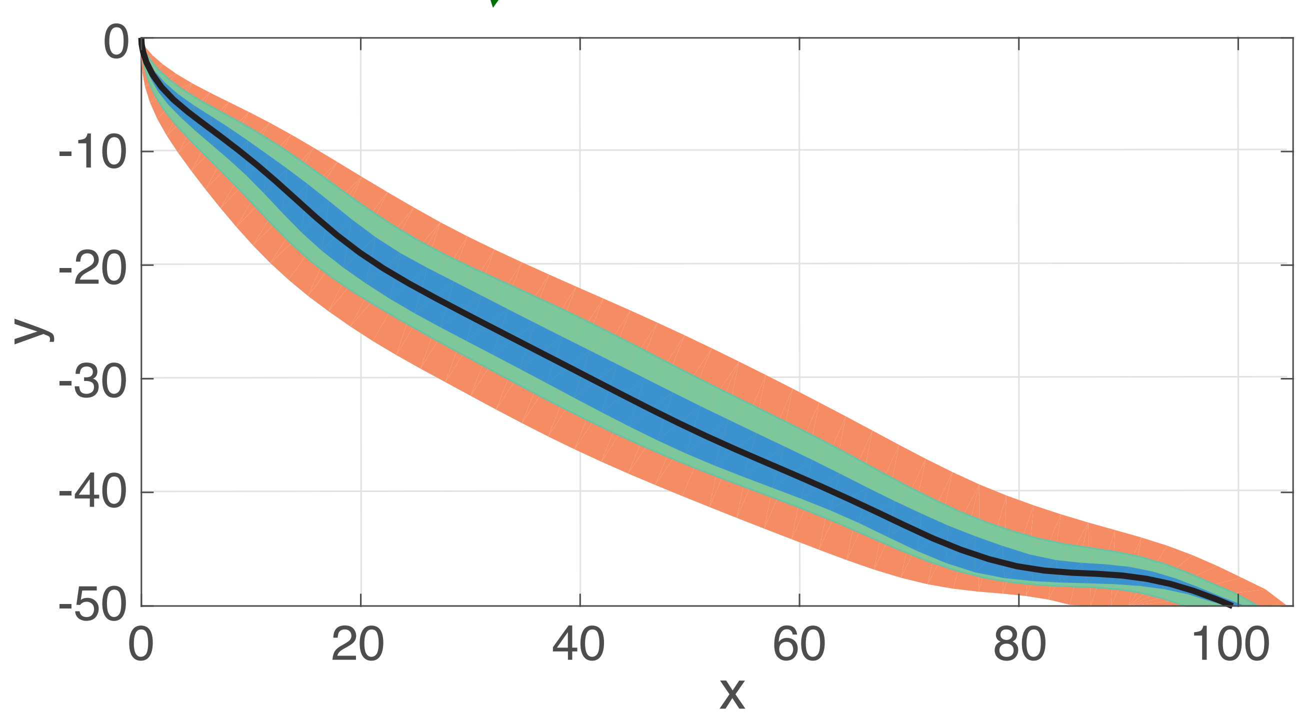
Thanks to Greg from San Diego for the bounding flight sketch.

RL vs. optimal control



- Strategies found by RL slightly surpass the performance of those obtained with optimal control (Paoletti 2011)
- Qualitative behavior is preserved
- RL is shown to be robust to **perturbation of model parameters** and **unseen initial conditions**



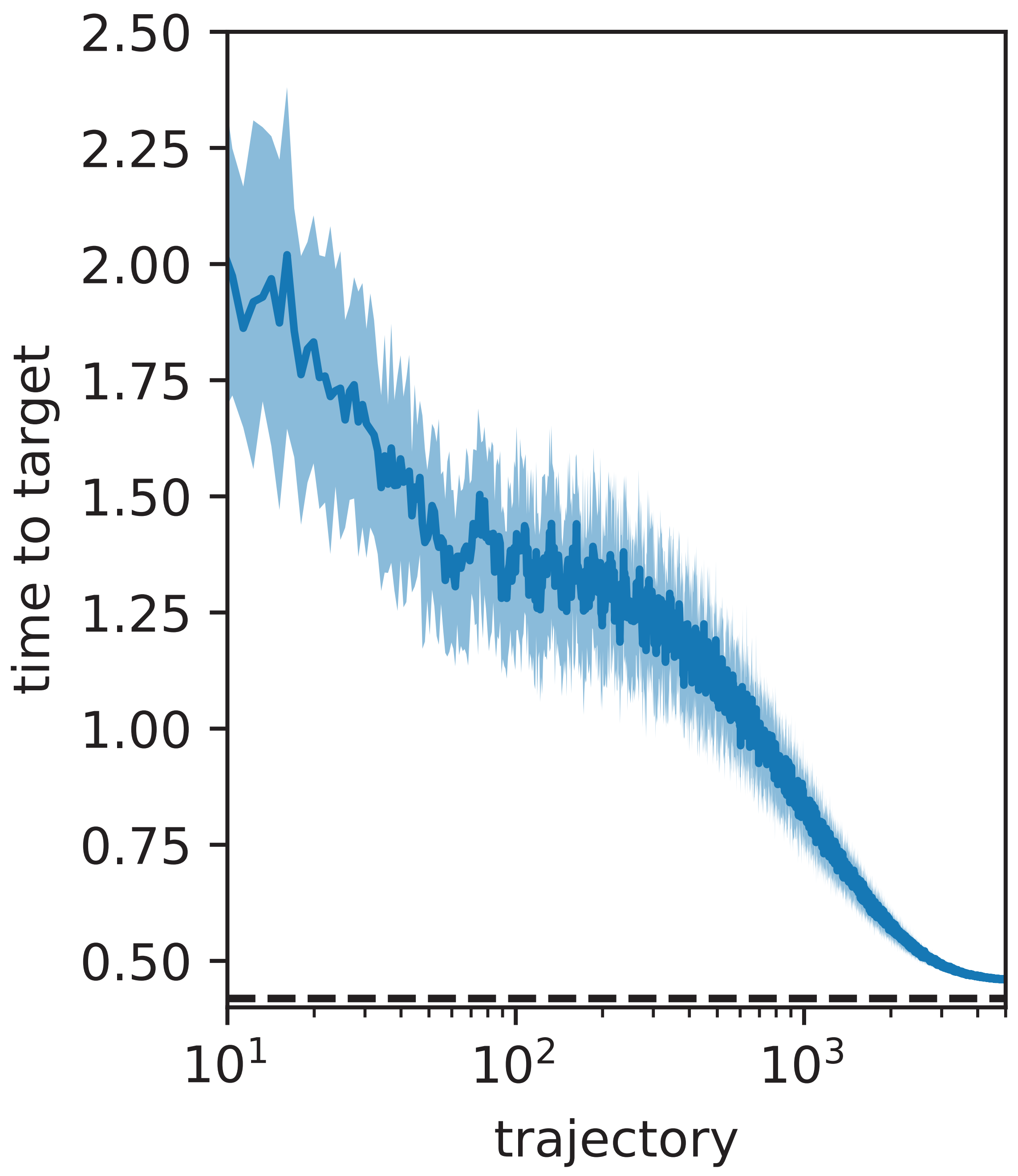


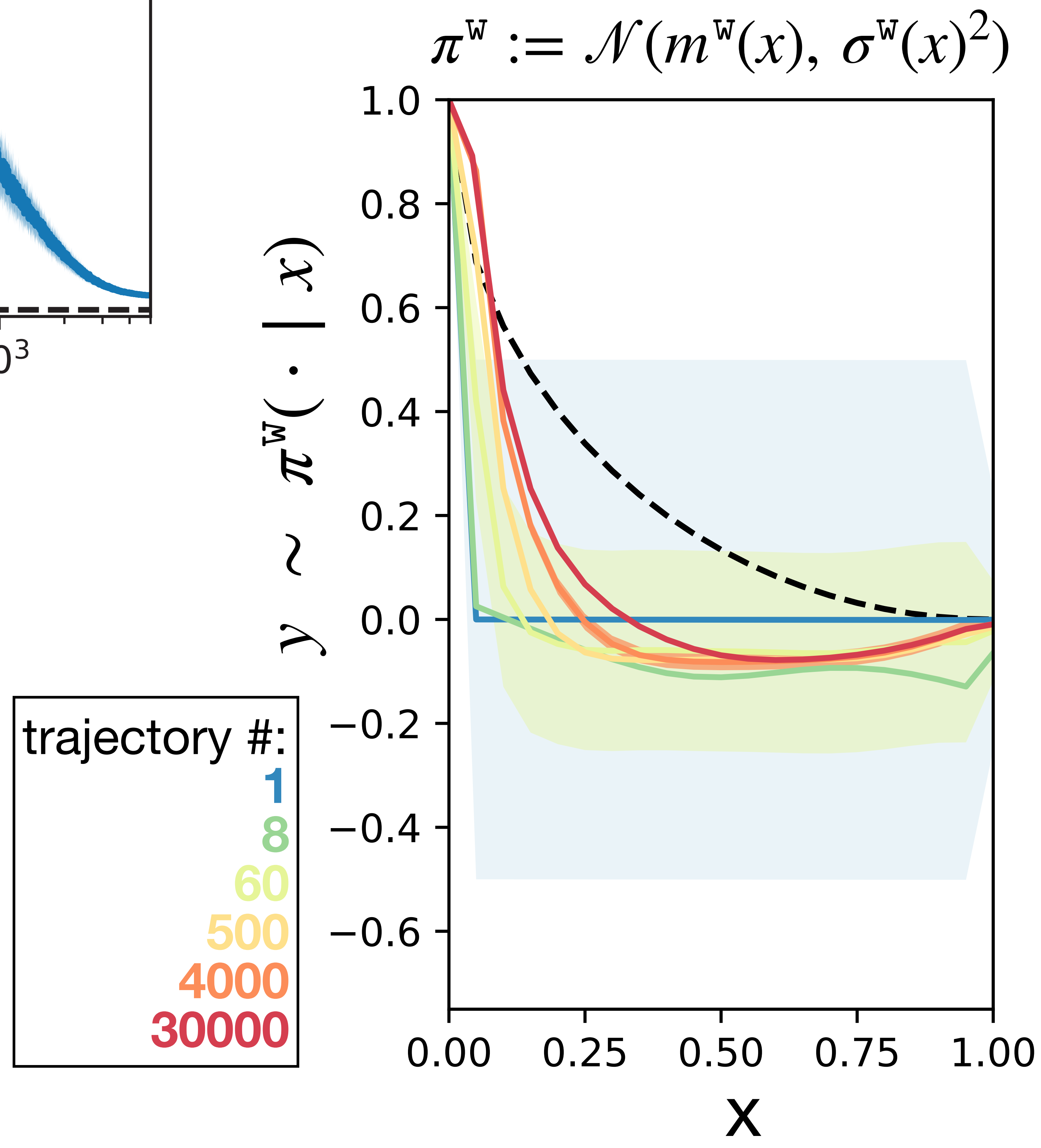
Legend: **RL**, **OC** for $\beta = 0.1$ $\rho^* = 100$

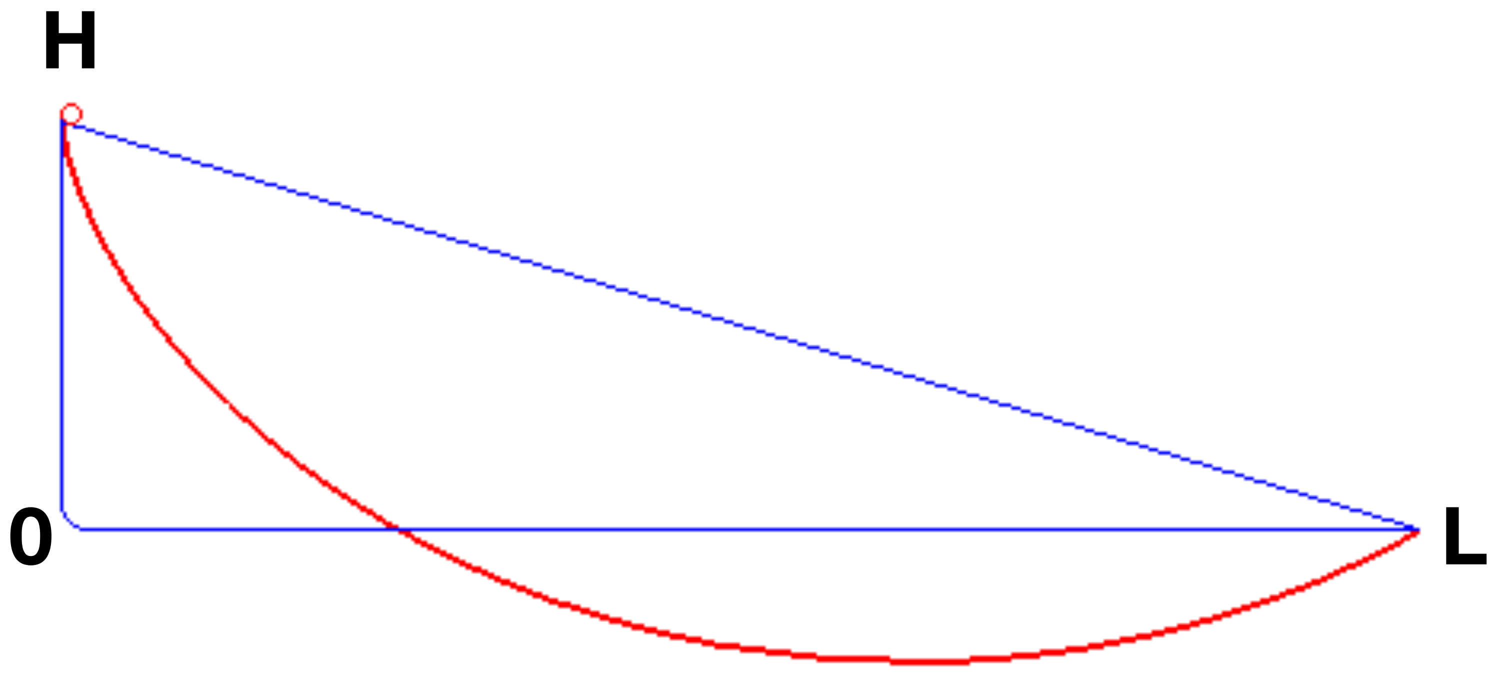
RL FAILURE : Brachistochrone problem (Johann Bernoulli in 1696)

"Given two points A and B in a vertical plane, what is the curve traced out by a point acted on only by gravity, which starts at A and reaches B in the shortest time."

$$
\begin{aligned}
\underset{T}{\text{minimize}} \quad & T = \int_0^L \frac{1}{v_x}\mathrm{dx} \\
\text{subject to} \quad & v_x^2 + v_y^2 = 2\sqrt{g\Delta y} \\
& y(x=0) = H \\
& y(x=L) = 0 \\
& y(x) \;\sim\; \pi^{\mathtt{W}}(y \mid x)
\end{aligned}
$$







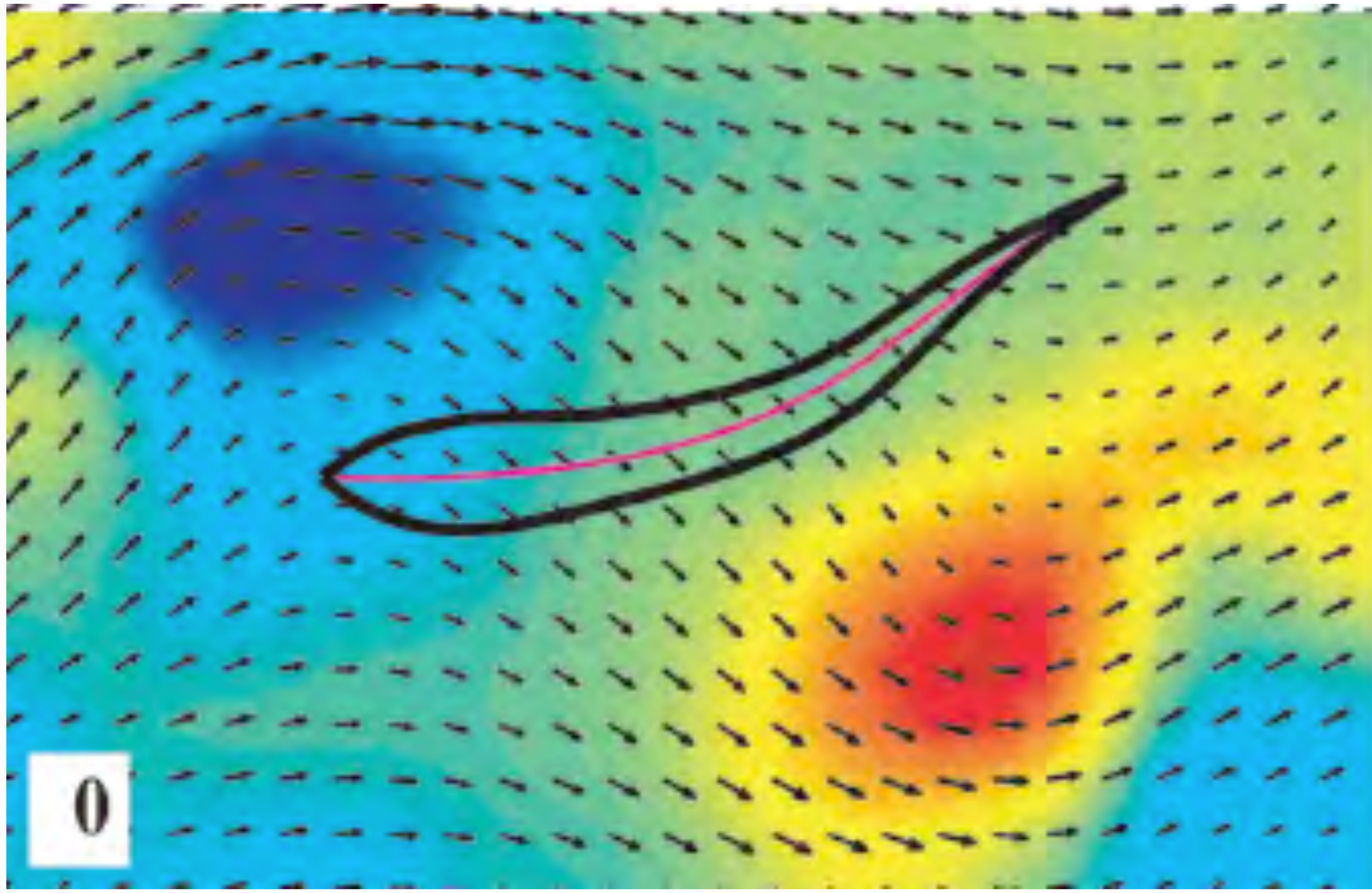
Solution is a cycloid which always starts at a cusp.

Fish Exploiting Vortices Decrease Muscle Activity

James C. Liao[1,*], David N. Beal[2], George V. Lauder[1], Michael S. Triantafyllou[2]

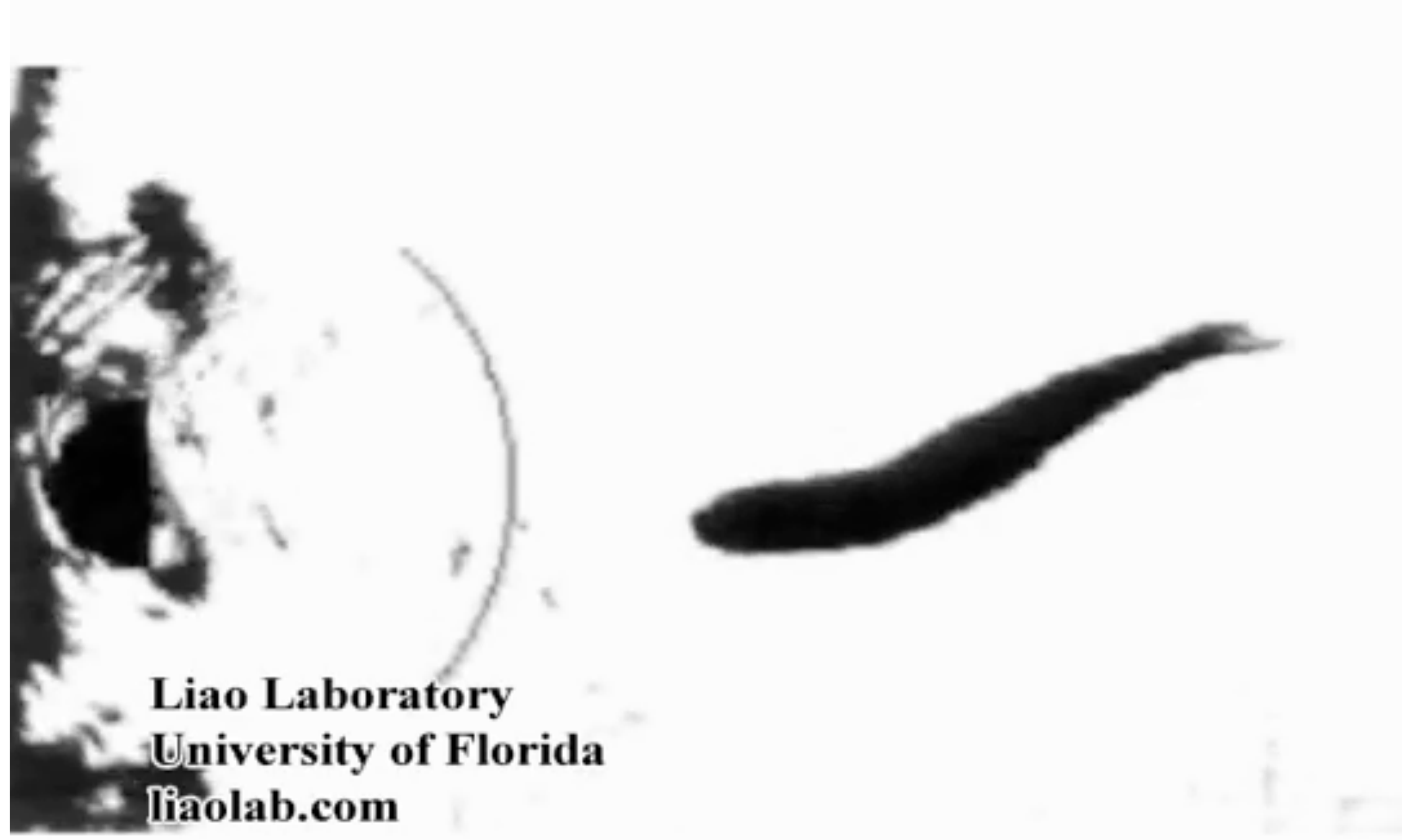
+ See all authors and affiliations

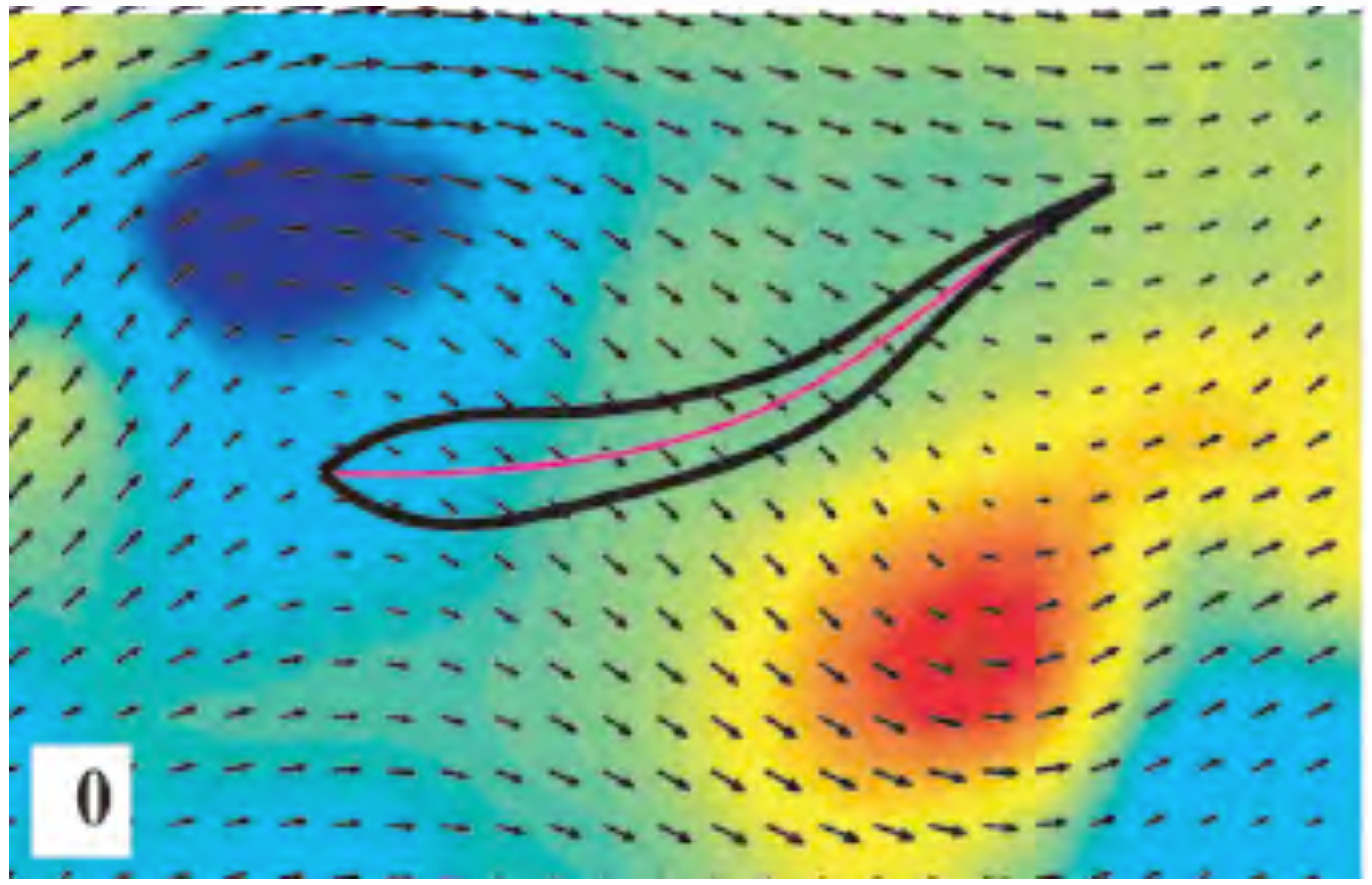
Science 28 Nov 2003:
Vol. 302, Issue 5650, pp. 1566-1569
DOI: 10.1126/science.1088295



Learning to Capture Vortices

Kármán gait employed by fish to decrease effort when swimming behind obstacles





Liao, Beal, Lauder, Triantafyllou, Science 2003

Agent : a **self-propelled swimmer**

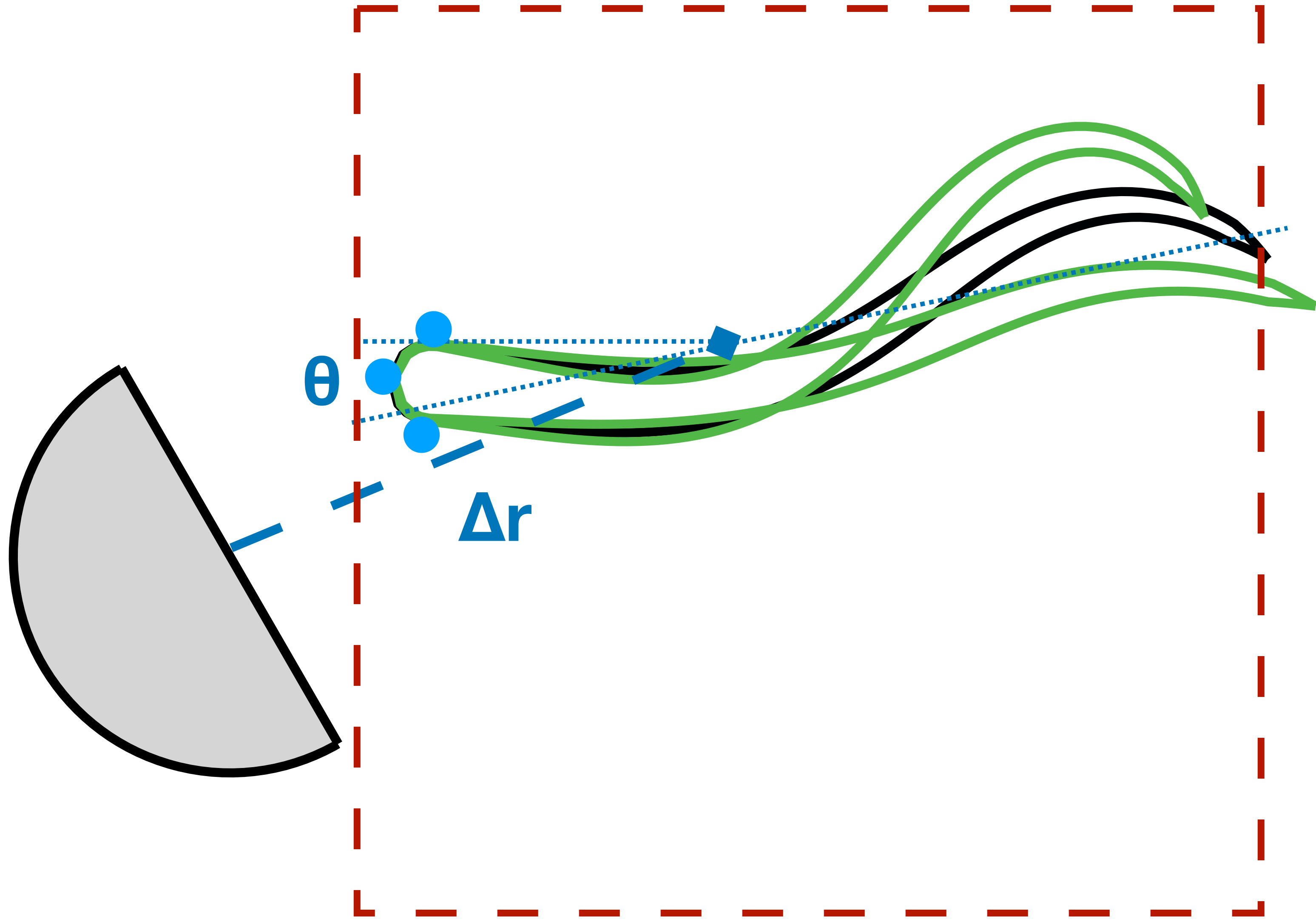
State:
- Relative position **Δr**, angle **θ**, velocities
- Shear stress sensors (lateral-line):

Action:
- Increase/decrease undulation amplitude
- Tail-beating frequency

Reward:
- Minimize the swimmer's energy output:

$$= -P_{\mathrm{def}} = \int_S \mathbf{u}_{\mathrm{deformation}} \cdot \mathrm{d}\mathbf{F}_{\mathrm{fluid}}$$

- Terminate if reaches border: $r_T = -100$



- **swimmer's power output decreases by 45% relative to swimming in quiescent flow**

Sensitivity/Failure of Deep Reinforcement Learning

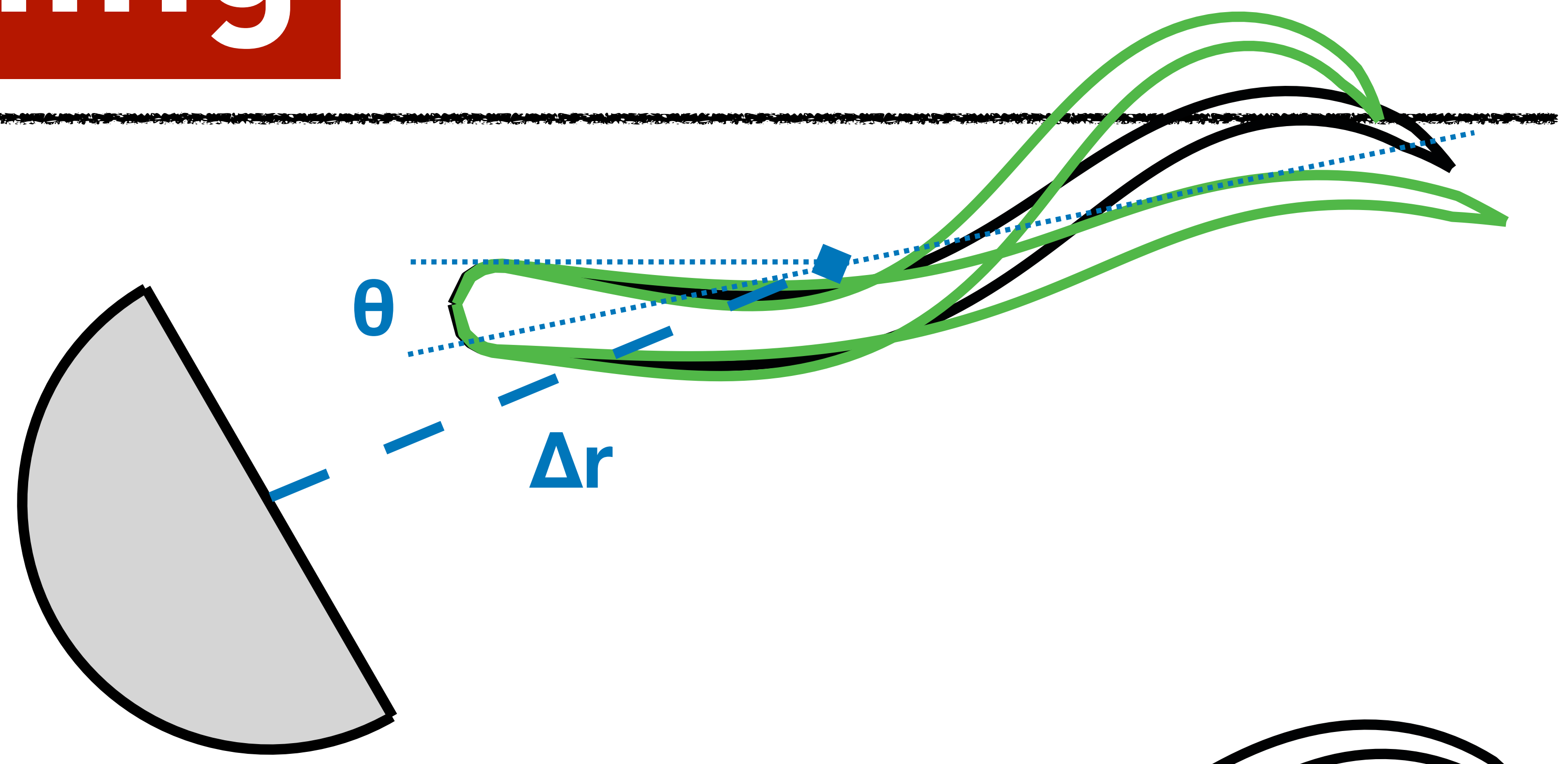
Policy trained at $Re = 1000$

at $Re = 1200$

at $Re = 2000$

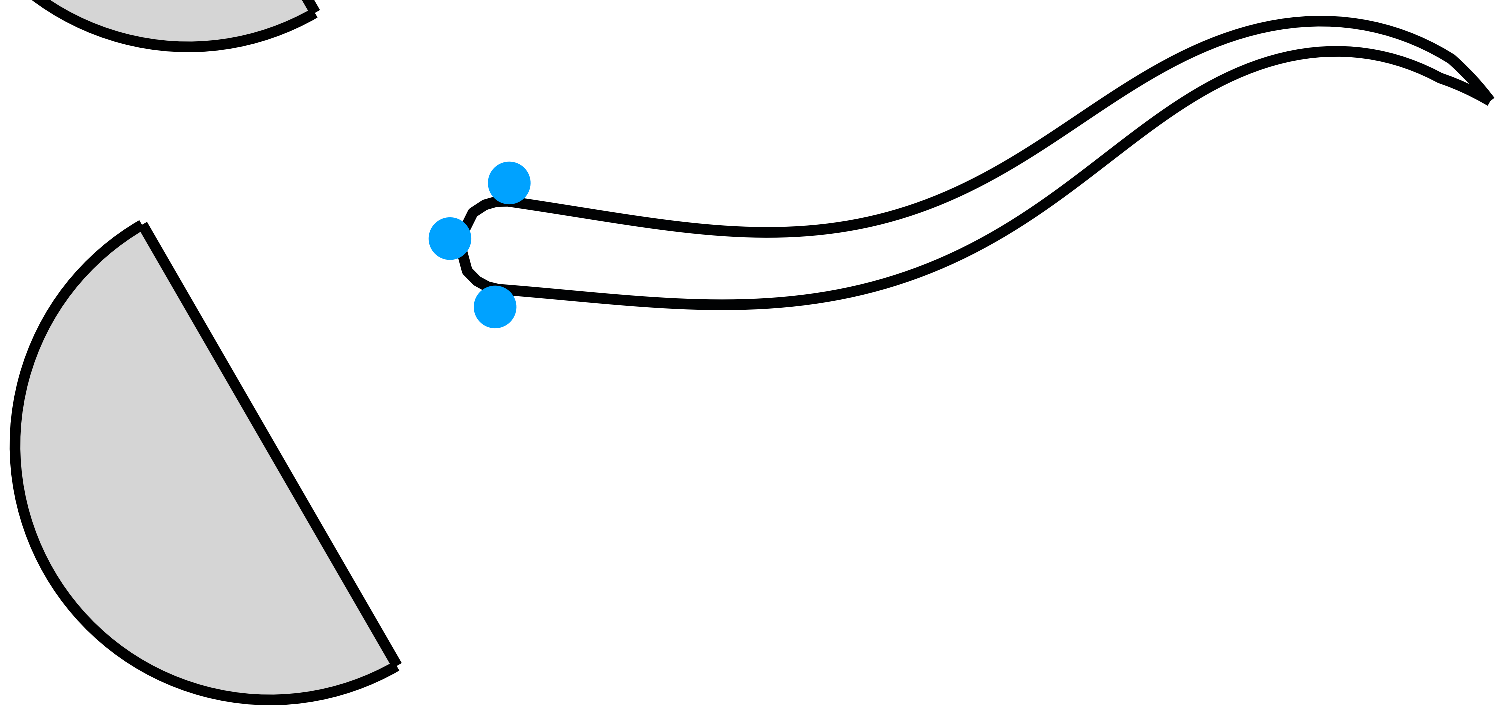
FAILURE of Deep Reinforcement Learning

State:
- Relative position **Δr**, angle **θ**, velocities
- Shape in previous two timesteps
- ~~Shear stress sensors (lateral-line):~~



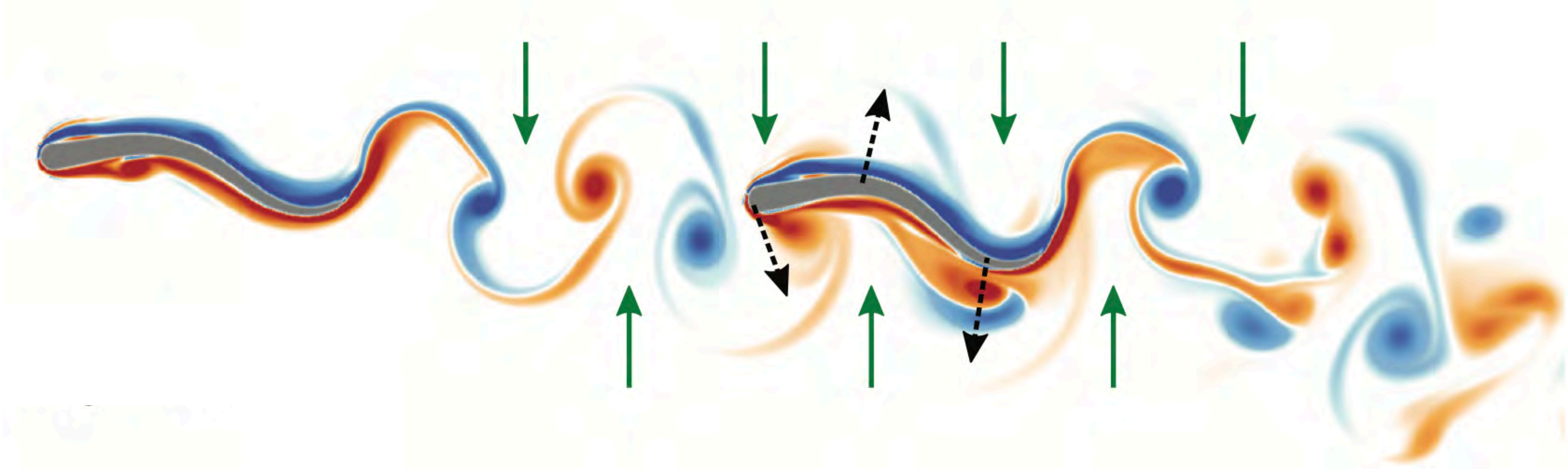
or

State:
- ~~Relative position **Δr**, angle **θ**, velocities~~
- ~~Shape in previous two timesteps~~
- Shear stress sensors (lateral-line):



Deep Reinforcement Learning is very sensitive to the choice of states

Two FISH

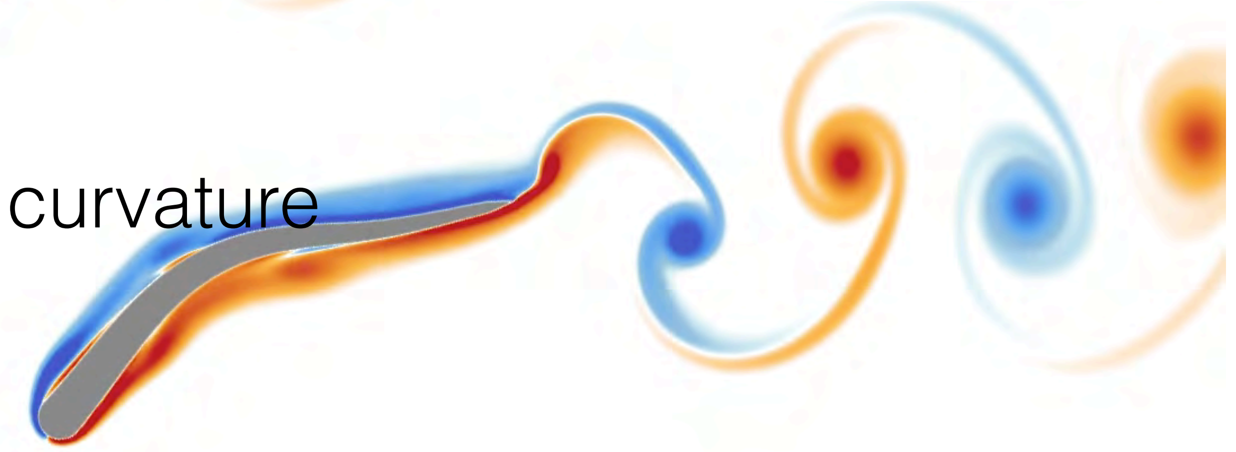


Synchronisation through Learning for Self-propelled Swimmers

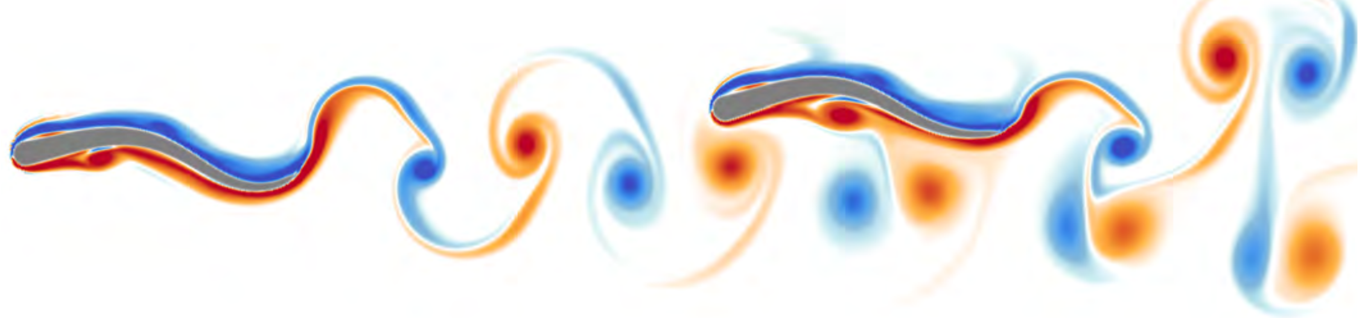
ACTIONS: **Modulate** body deformation

- Increase curvature

- Decrease curvature

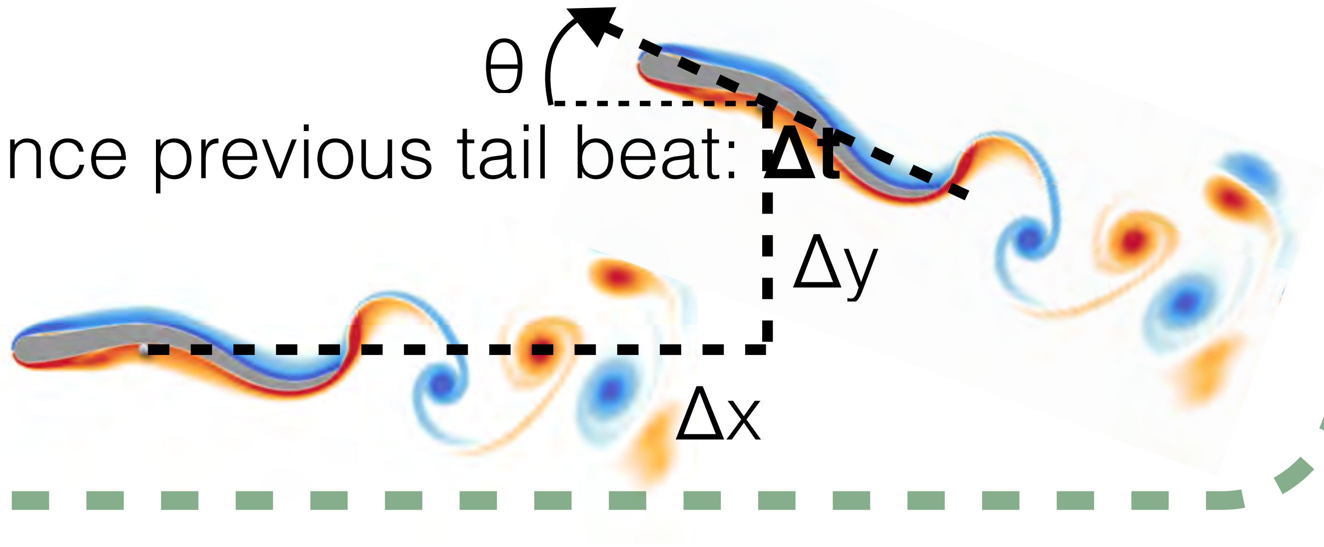


REWARD :distance to leader OR swimming-efficiency

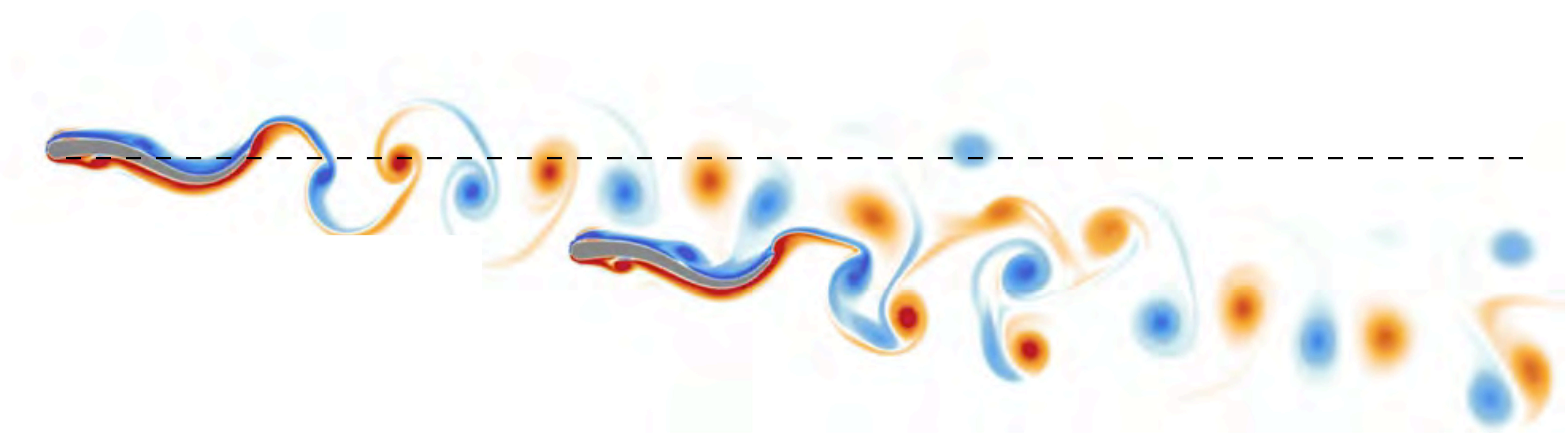


STATES

- Orientation relative to leader: **Δx, Δy, θ**
- Time since previous tail beat: **Δt**
- Current manoeuvre

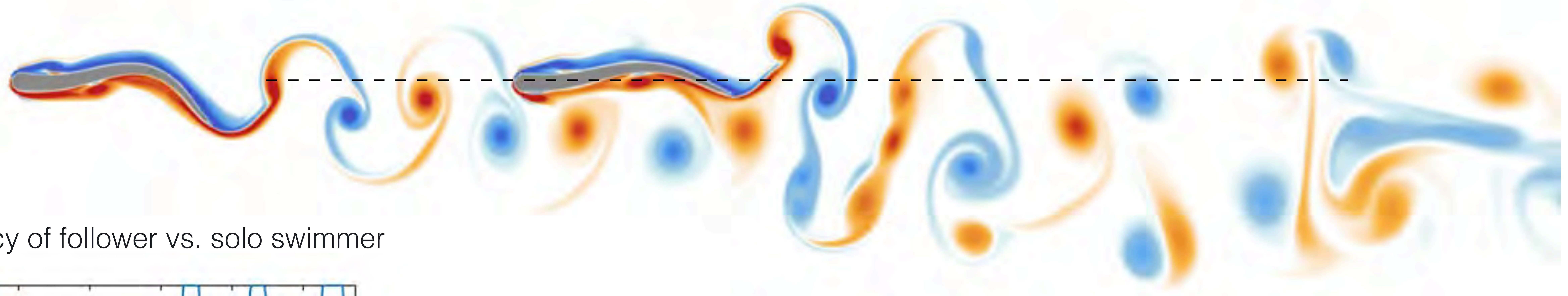


GOAL I : minimize Δy

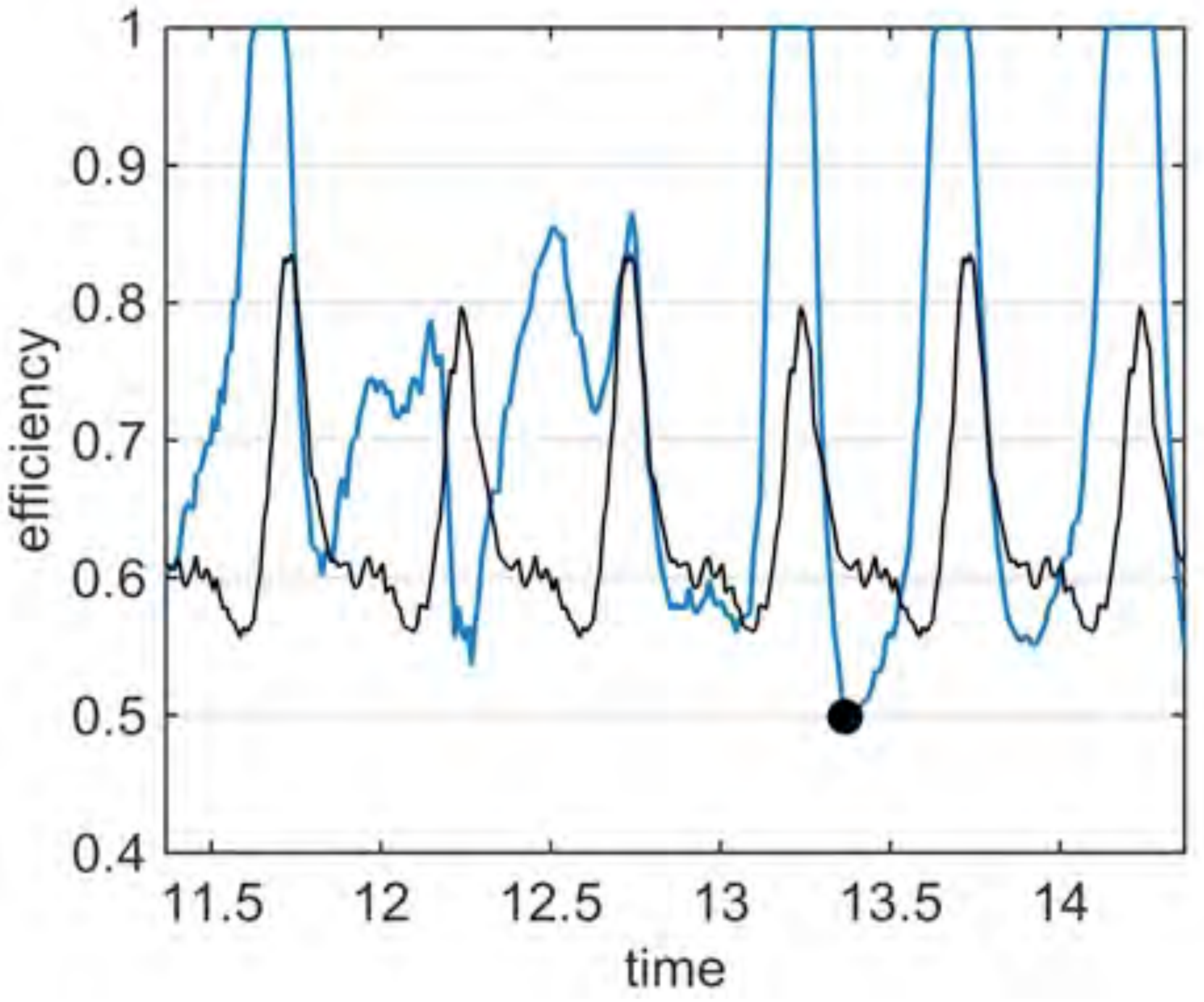


- Smart Follower:
  - Stay in the leader's wake
  - Steer and course correct
  - P_{def} **drops** : decreased fluid resistance due to assistance from vortices

IS IT EFFICIENT TO SWIM IN A WAKE ?



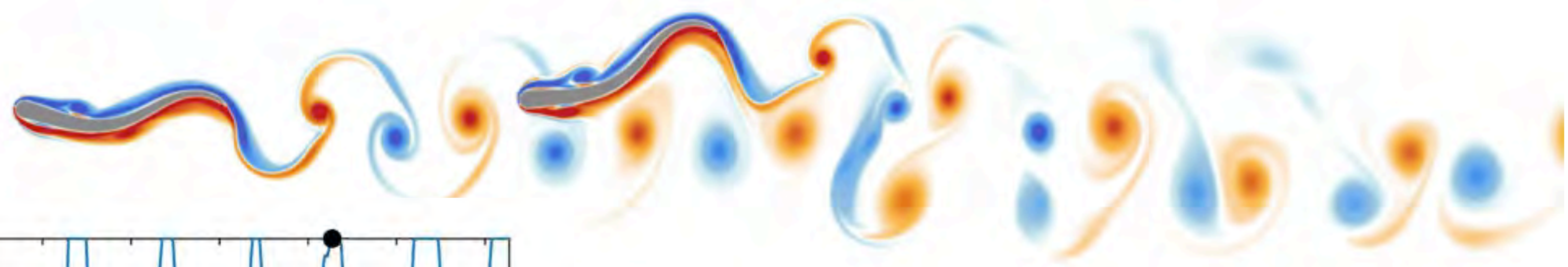
Efficiency of follower vs. solo swimmer

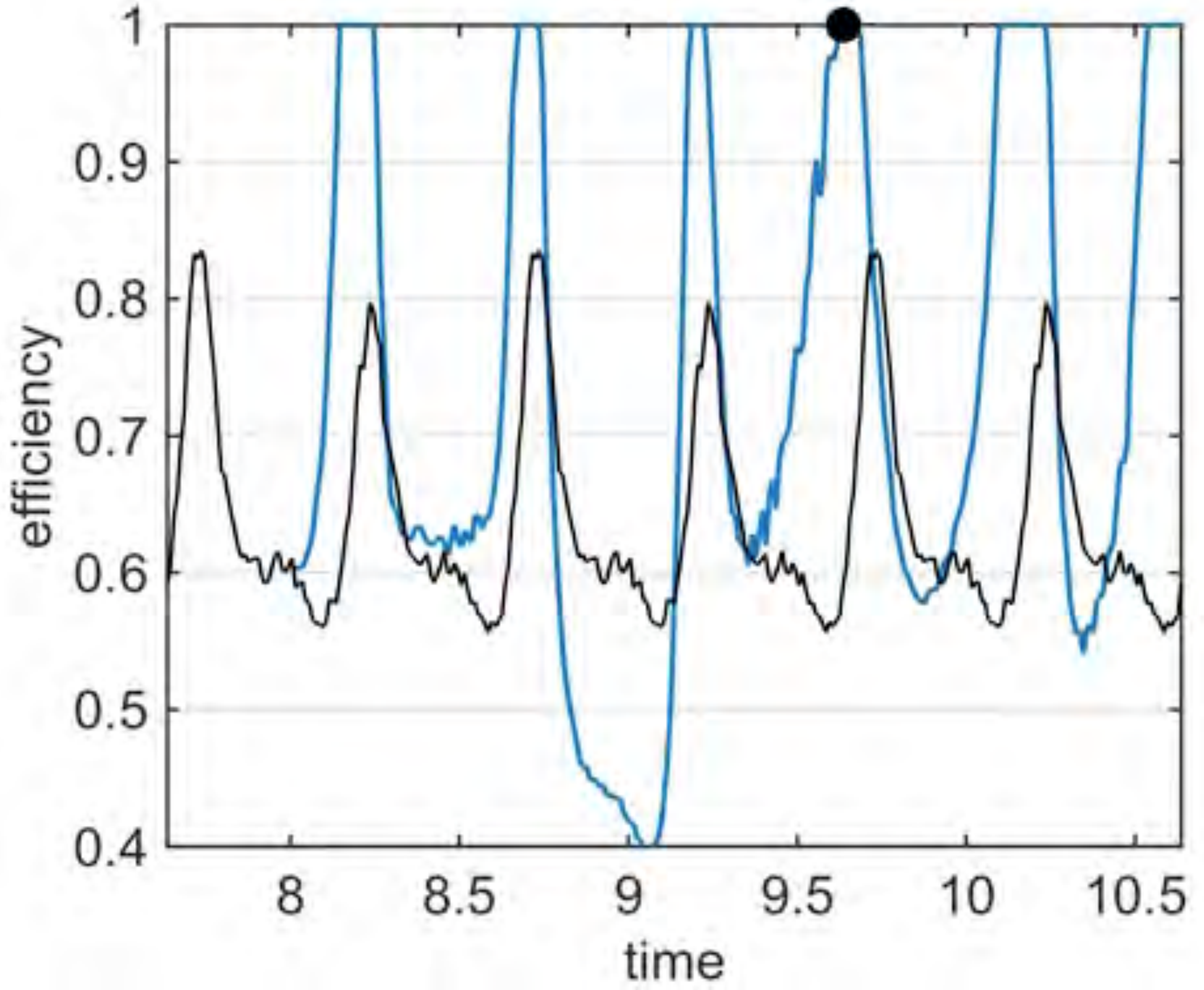


$$\text{efficiency} = \frac{Tu}{Tu + max(\int \mathbf{F} \cdot \mathbf{u}_{def} dS, 0)}$$

- Average **efficiency increased by 11.8**%
- Increased efficiency was not a learning objective:
  - <u>Emerges</u> from learning to stay in leader's wake

GOAL II : MAX EFFICIENCY



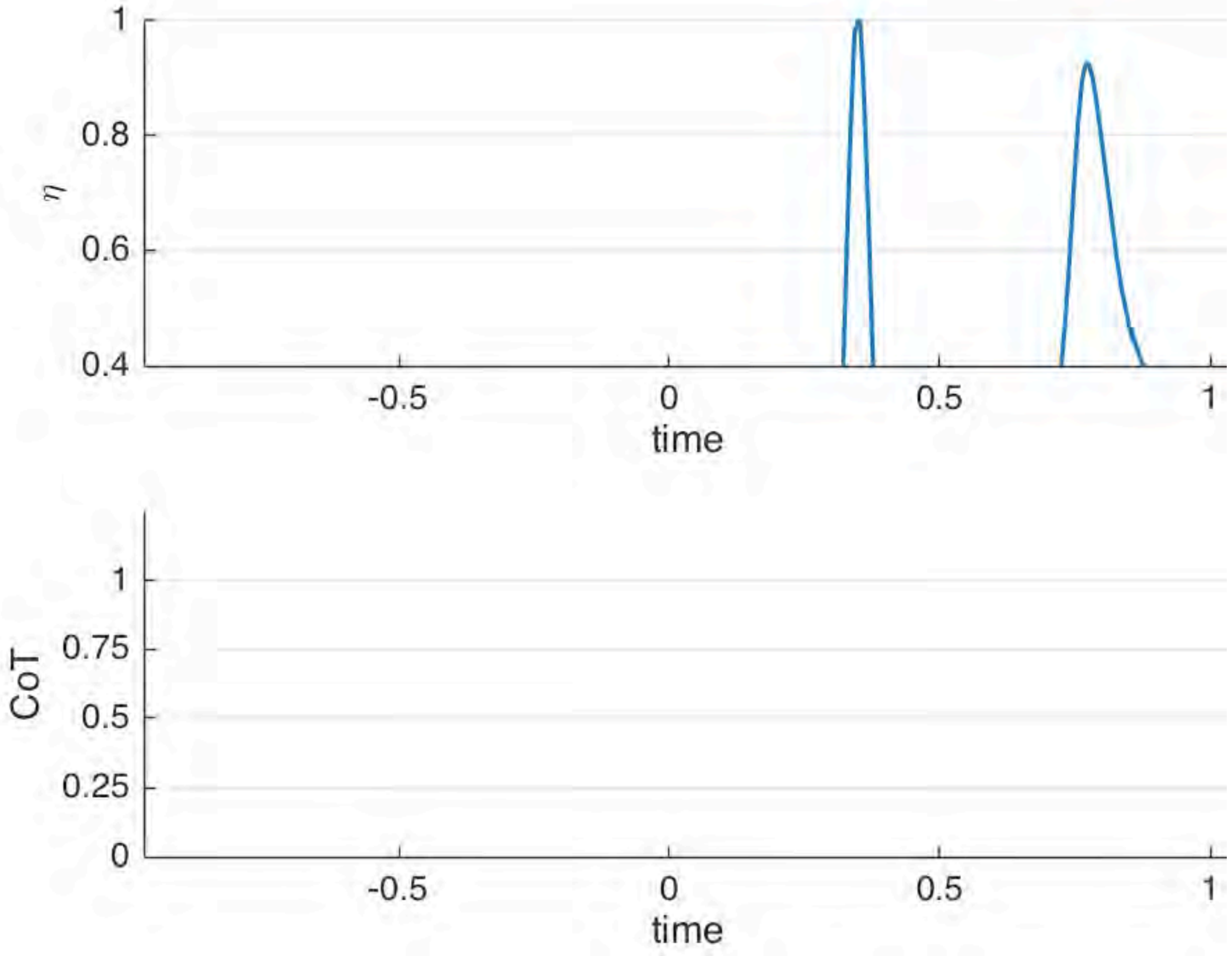


No distance constraints specified

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

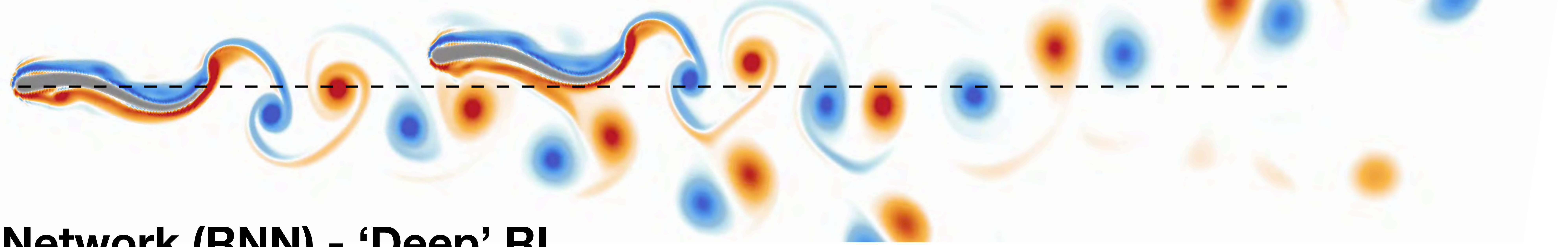
$$R_\eta = \frac{Tu}{Tu + max(P_{def}, 0)}$$

Autonomous “smart” follower **reacts effectively** to **unfamiliar actions** by leader

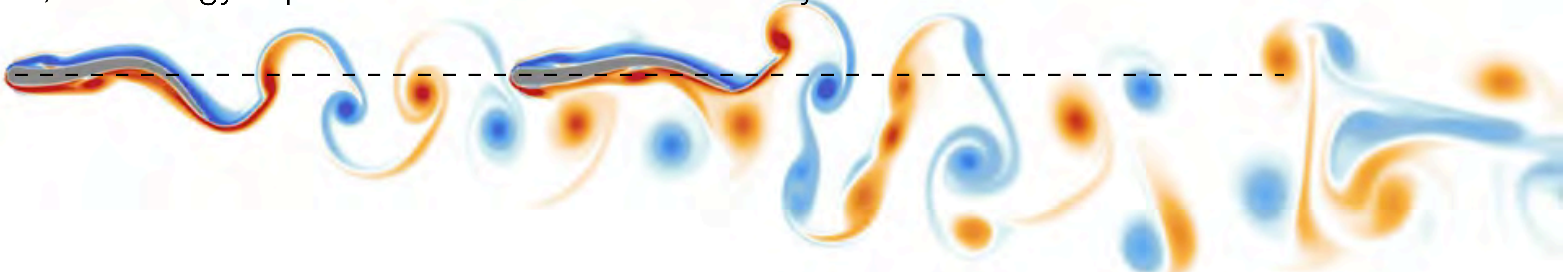


Influence of Time-history - MEMORY

- **Feed Forward Network (FFN) - ‘Vanilla’ RL**
  - Action depends <u>*explicitly*</u> on the current state only
  - Long-term dependencies ignored
  - ***Overshoots*** - Frequent course correction necessary

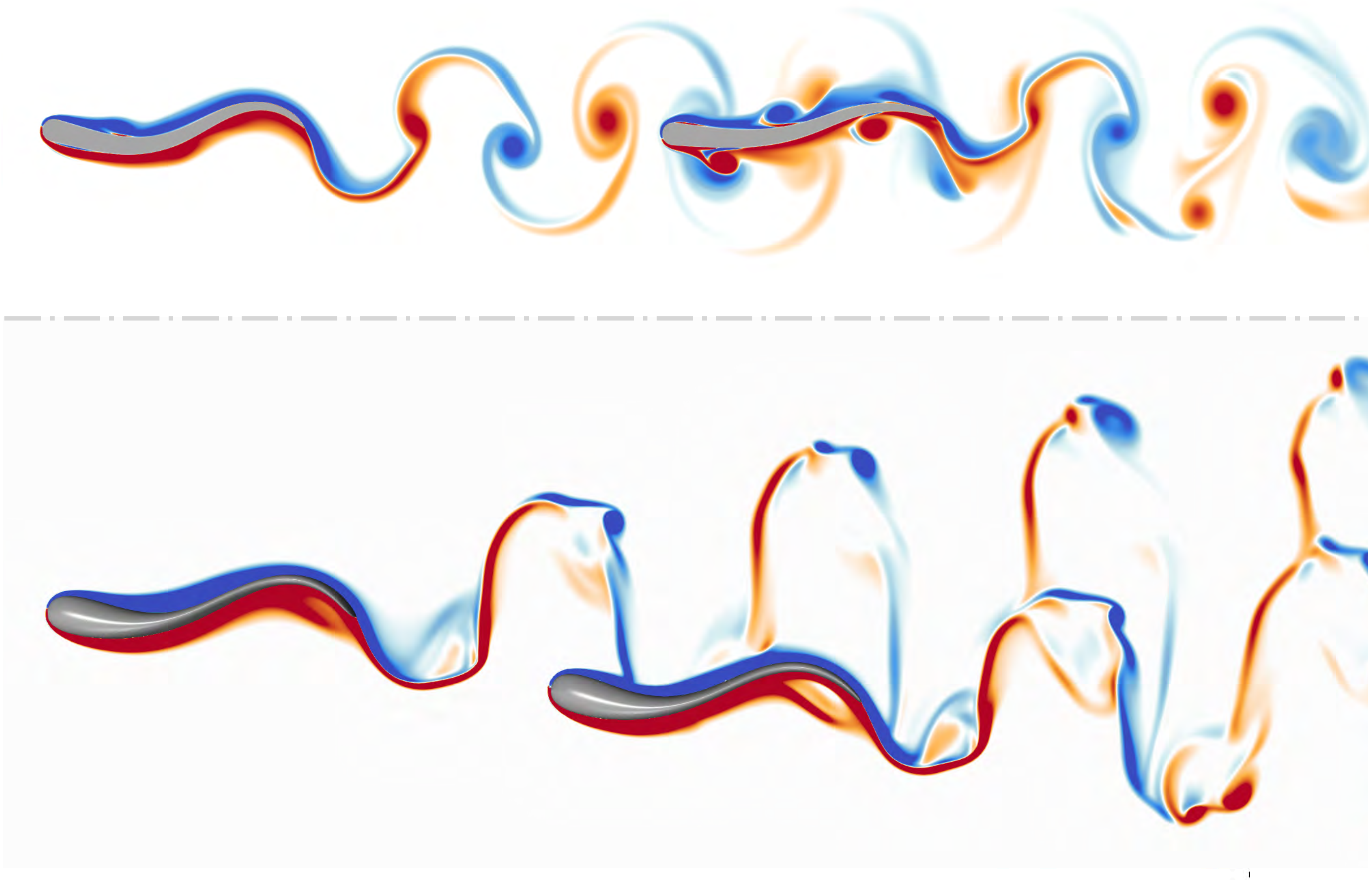


- **Recurrent Neural Network (RNN) - ‘Deep’ RL**
  - Action depends on information from agent’s time-history
  - Agent ***anticipates*** interaction with the flow - more robust control
  - Better attitude-control, but energy expenditure increases considerably

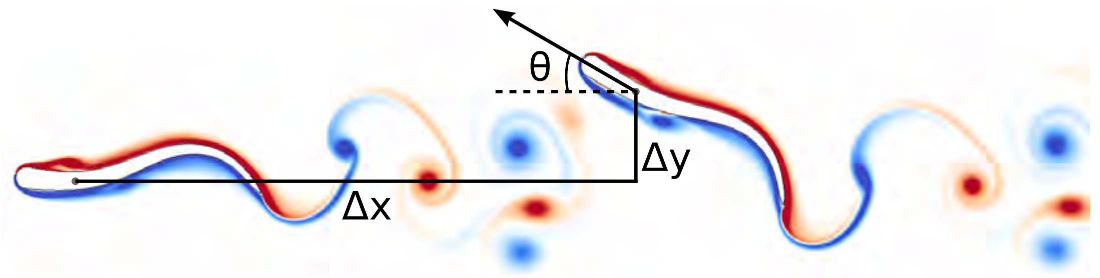


FF NN - distance
+17%
RNN - distance
+12%
RNN - efficiency
+28.0%

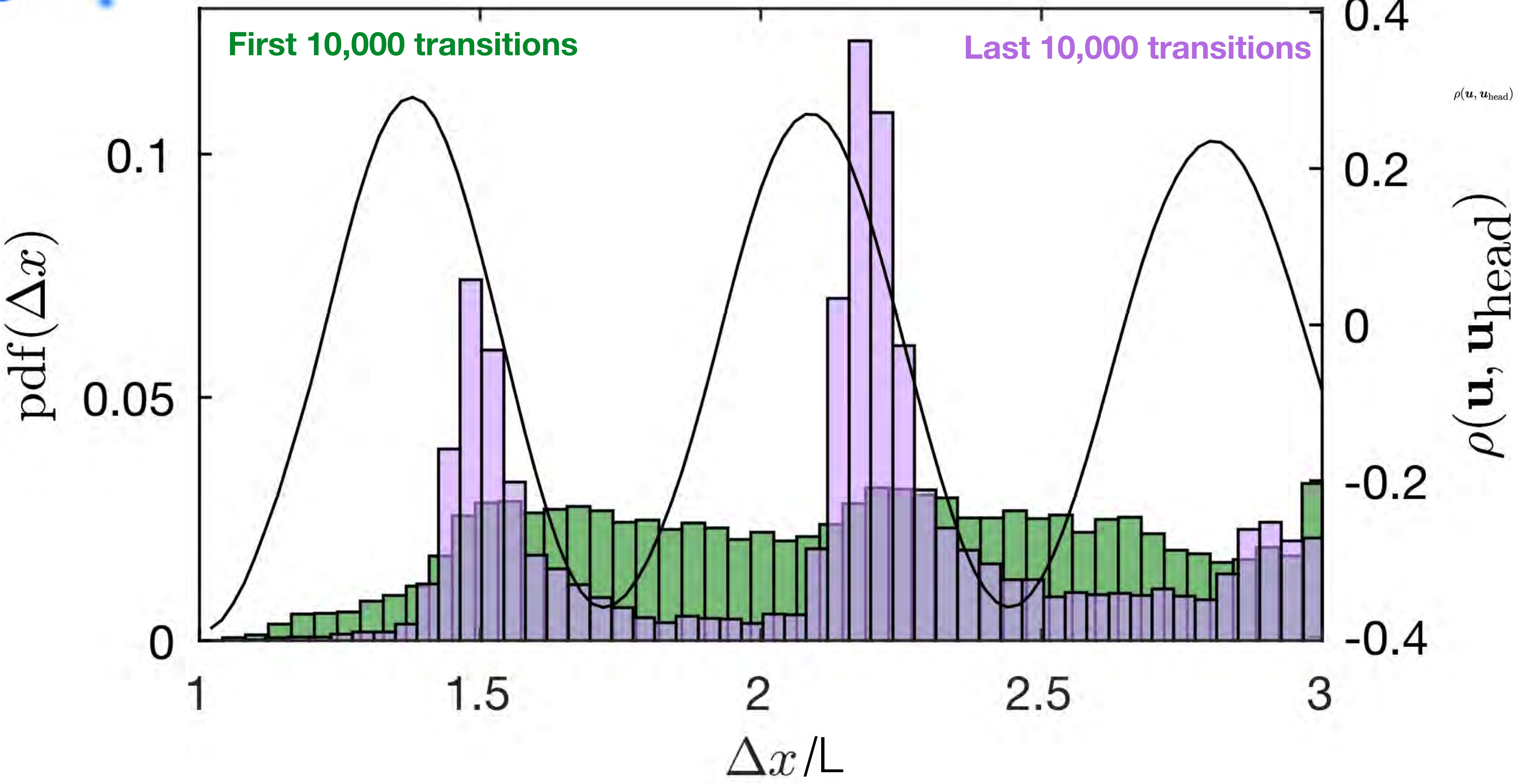
3D Schools



What can WE learn from 'smart' swimmers?



How does behaviour evolve ***during*** training?

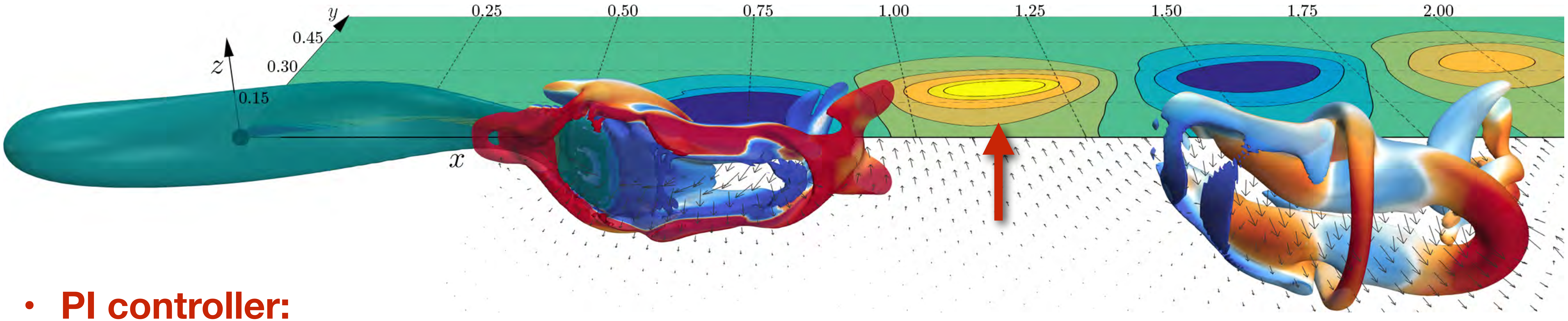


$$\rho(\boldsymbol{u}, \boldsymbol{u}_{\mathrm{head}}) = \frac{\operatorname{cov}\left(\boldsymbol{u}(x,y), \boldsymbol{u}_{\mathrm{head}}\right)}{\sigma_{\boldsymbol{u}(x,y)}\ \sigma_{\boldsymbol{u}_{\mathrm{head}}}} = \frac{\sum_t \boldsymbol{u}(x,y,t) \cdot \boldsymbol{u}_{\mathrm{head}}(t)}{\sqrt{\sum_t \|\boldsymbol{u}(x,y,t)\|^2\ \sum_t \|\boldsymbol{u}_{\mathrm{head}}(t)\|^2}}$$

$$\rho(\boldsymbol{u}, \boldsymbol{u}_{\mathrm{head}}) = \frac{\operatorname{cov}\left(\boldsymbol{u}(x,y), \boldsymbol{u}_{\mathrm{head}}\right)}{\sigma_{\boldsymbol{u}(x,y)}\ \sigma_{\boldsymbol{u}_{\mathrm{head}}}} = \frac{\sum_t \boldsymbol{u}(x,y,t) \cdot \boldsymbol{u}_{\mathrm{head}}(t)}{\sqrt{\sum_t \|\boldsymbol{u}(x,y,t)\|^2\ \sum_t \|\boldsymbol{u}_{\mathrm{head}}(t)\|^2}}$$

averaged over one tail beat

$$\rho(\boldsymbol{u}, \boldsymbol{u}_{\text{head}}) = \frac{\operatorname{cov}\left(\boldsymbol{u}(x,y), \boldsymbol{u}_{\text{head}}\right)}{\sigma_{\boldsymbol{u}(x,y)}\ \sigma_{\boldsymbol{u}_{\text{head}}}} = \frac{\sum_t \boldsymbol{u}(x,y,t)\cdot \boldsymbol{u}_{\text{head}}(t)}{\sqrt{\sum_t \|\boldsymbol{u}(x,y,t)\|^2\ \sum_t \|\boldsymbol{u}_{\text{head}}(t)\|^2}}$$



- **PI controller:**
  - Modulate follower’s undulations (curvature + amplitude)
  - Maintain the target position specified

Verma et al., PNAS, 2018

Verma et al., PNAS, 2018

SUMMARY:

LEARNING

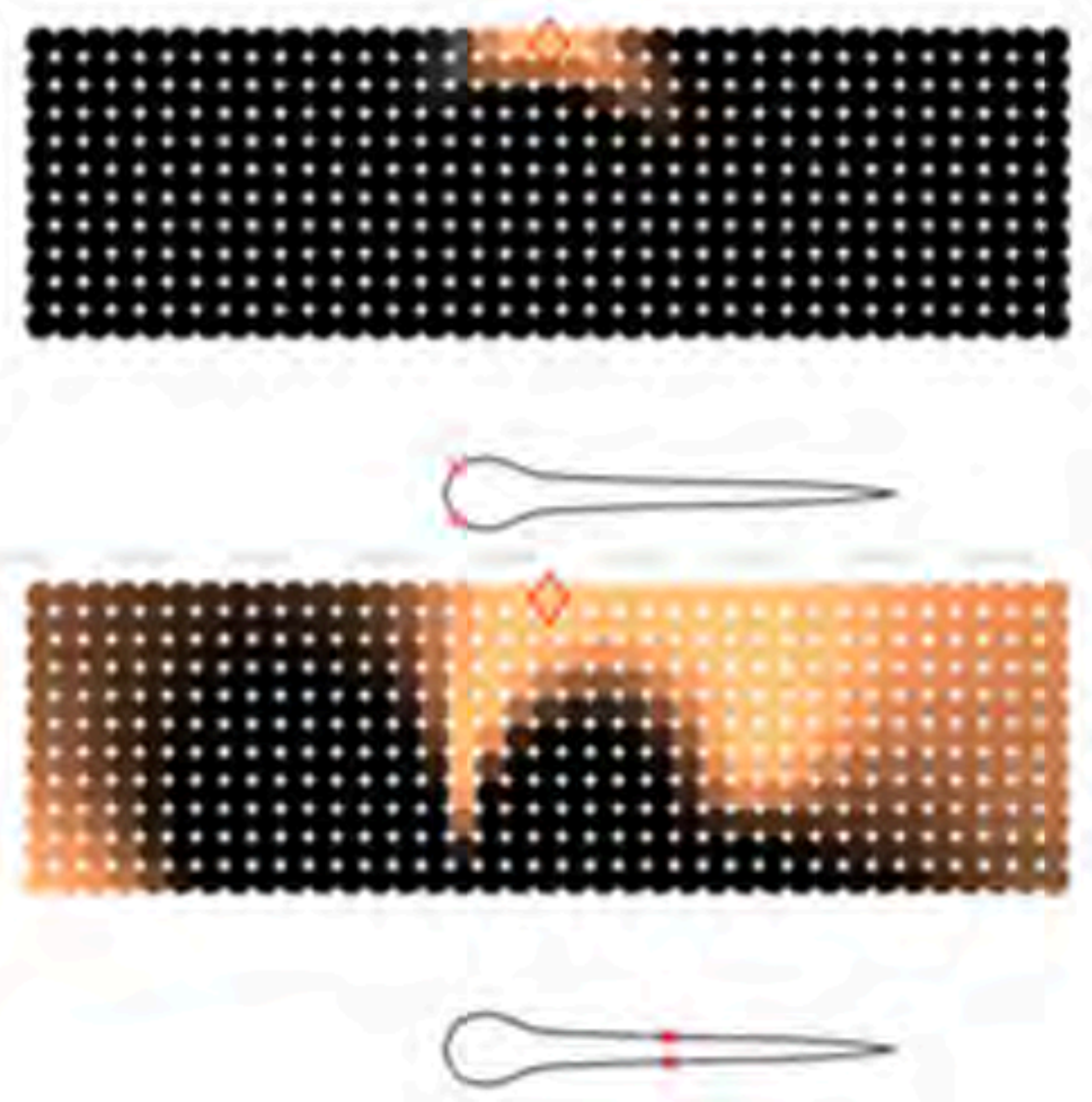
find an effective algorithm **(not ONLY machine learning)** for the **flow** problem.

Optimal sensor placement for artificial swimmers

Siddhartha Verma (a1) (a2) (a3), Costas Papadimitriou (a4), Nora Lüthen (a1), Georgios Arampatzis (a1) ...

DOI: https://doi.org/10.1017/jfm.2019.940 Published online by Cambridge University Press: 10 December 2019

Abstract

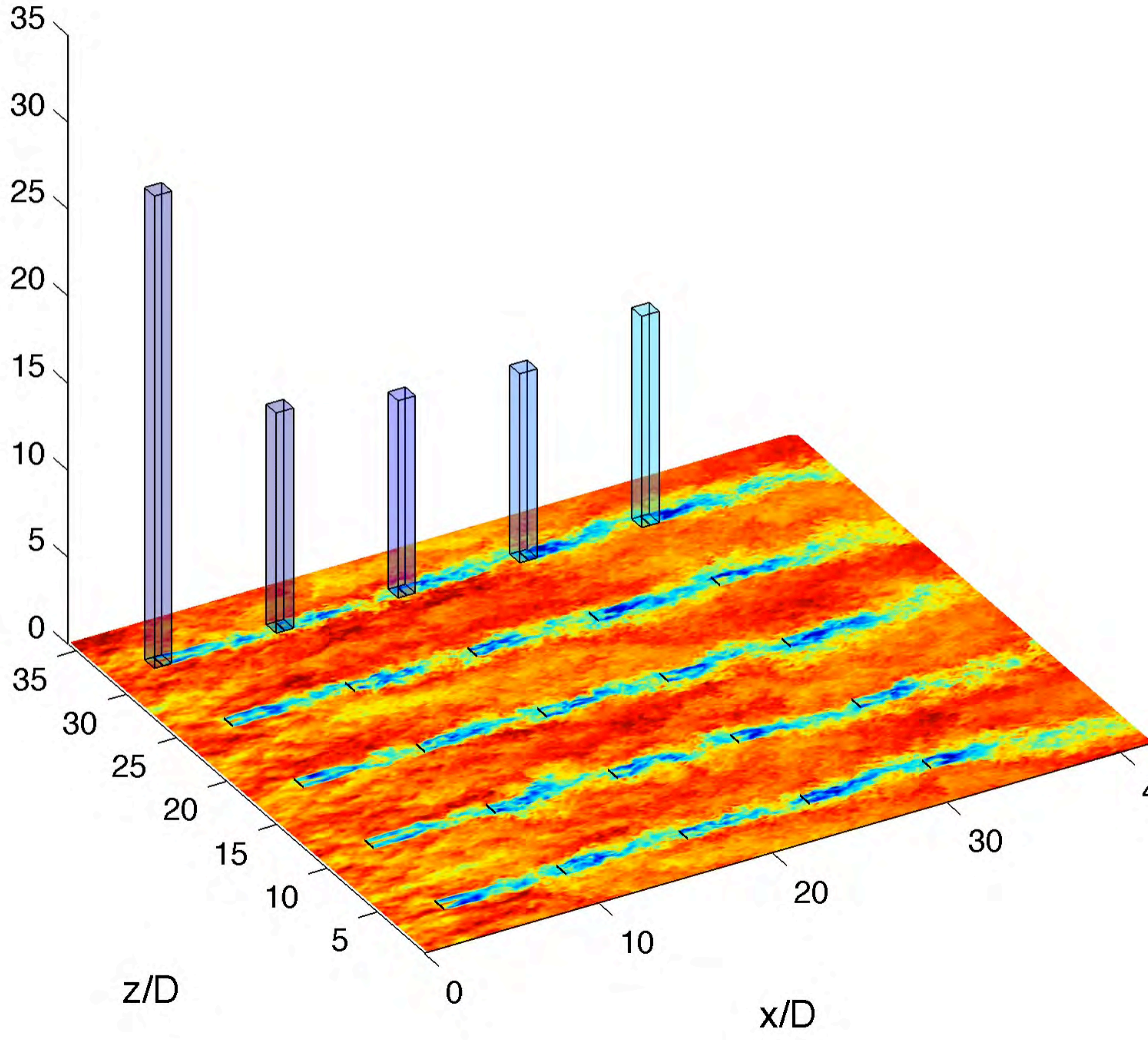


Natural swimmers rely for their survival on sensors that gather information from the environment and guide their actions. The spatial organization of these sensors, such as the visual fish system and lateral line, suggests evolutionary selection, but their optimality remains an open question. Here, we identify sensor configurations that enable swimmers to maximize the information gathered from their surrounding flow field. We examine two-dimensional, self-propelled and stationary swimmers that are exposed to disturbances generated by oscillating, rotating and D-shaped cylinders. We combine simulations of the Navier–Stokes equations with Bayesian experimental design to determine the optimal arrangements of shear and pressure sensors that best identify the locations of the disturbance-generating sources. We find a marked tendency for shear stress sensors to be located in the head and the tail of the swimmer, while they are absent from the midsection. In turn, we find a high density of pressure sensors in the head along with a uniform distribution along the entire body. The resulting optimal sensor arrangements resemble neuromast distributions observed in fish and provide evidence for optimality in sensor distribution for natural swimmers.

Wind Farms and wakes of wind turbine

http://www.noaanews.noaa.gov/stories2011/20110426_windwakes.html

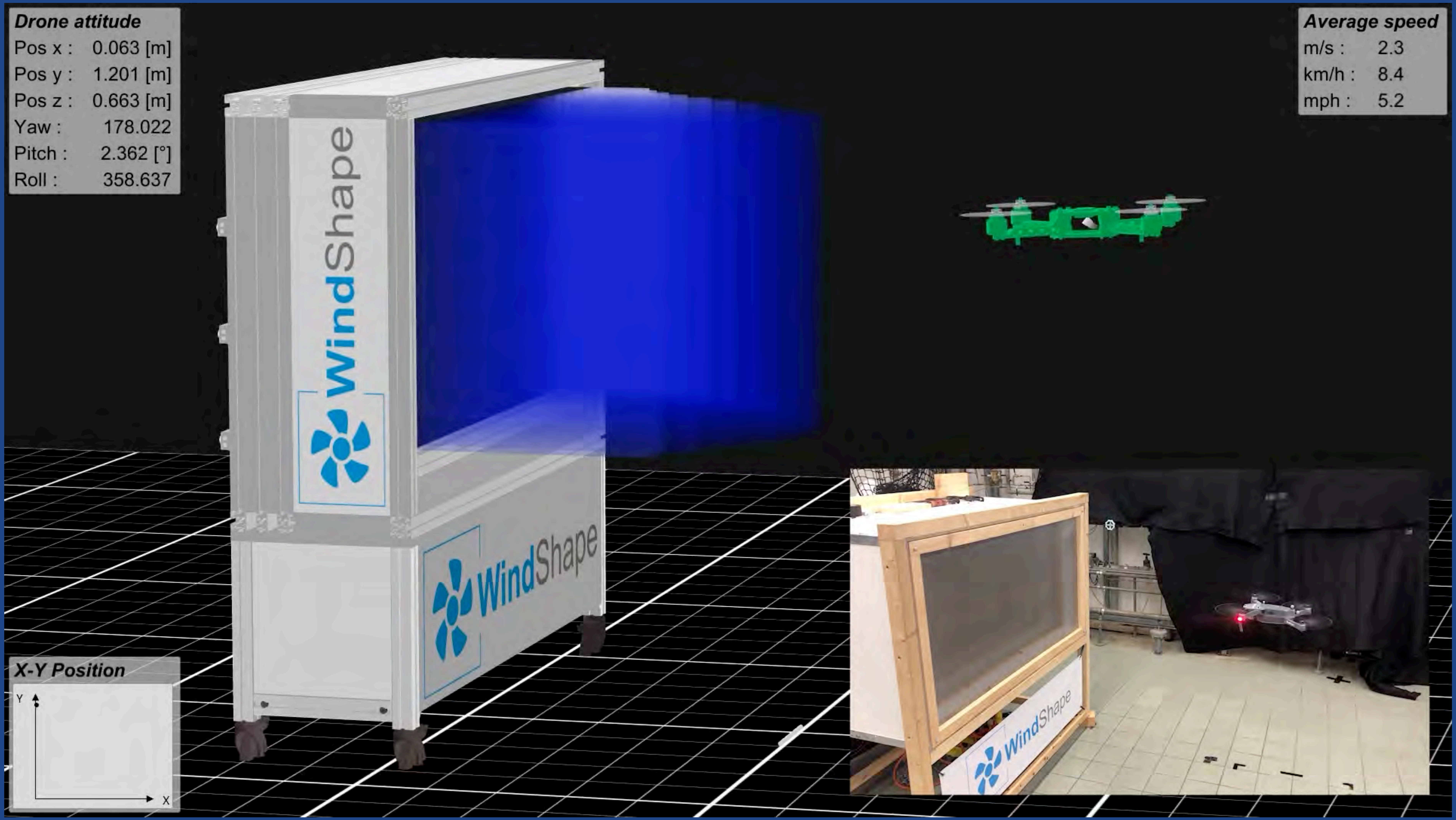
the velocity field at hub height and also the instantaneous powers for one "column" of wind turbines



CREDIT: P. CHATELAIN, UC LOUVAIN

Digital Wind Machine

Gusts & Shear



Thank you !

Machine Learning

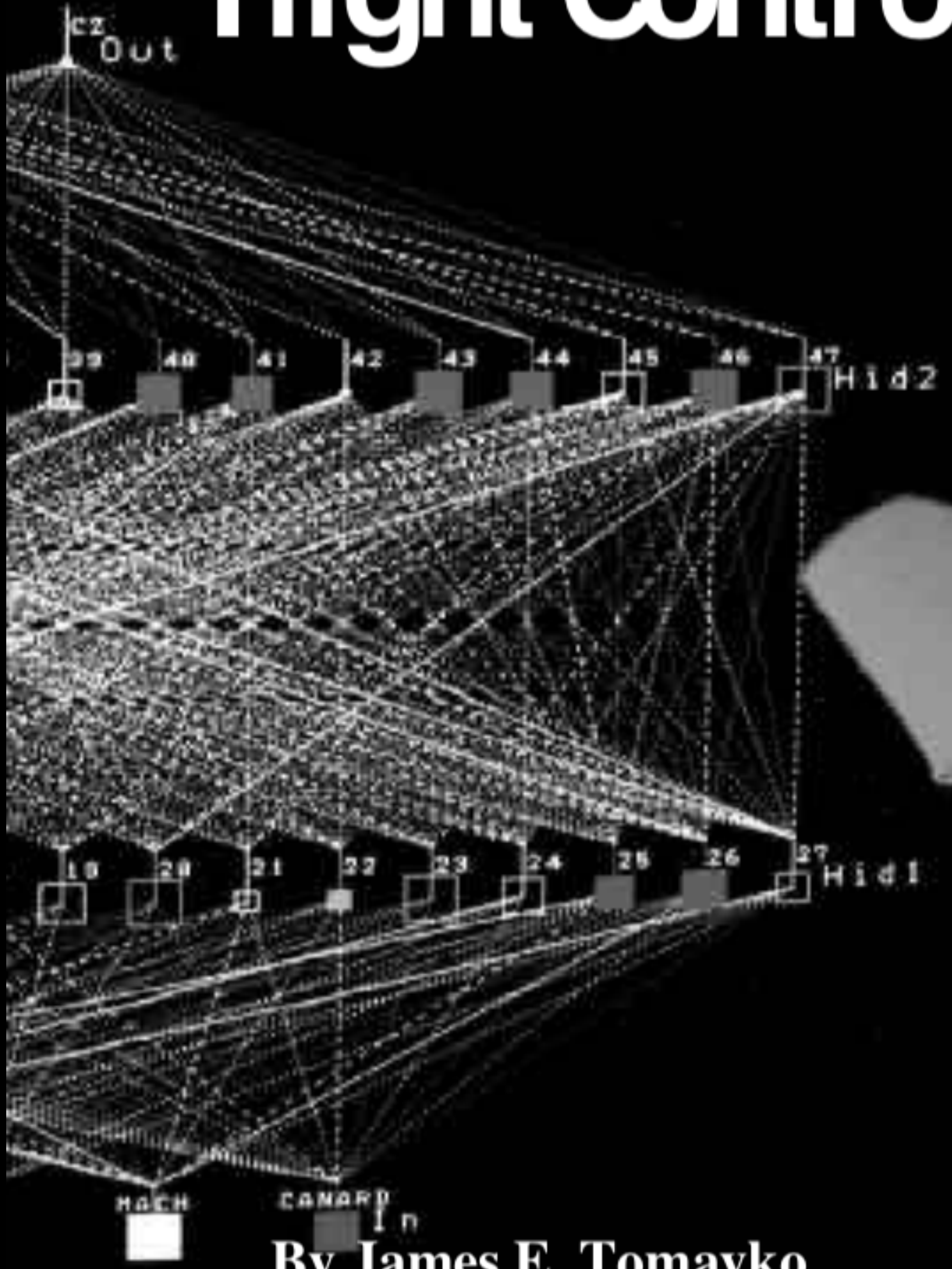
a personal history

NASA Photo *EC89 232-1*

An Israeli Air Force F-15 involved in a midair collision during a training mission in 1983. In spite of losing virtually the entire starboard wing, the pilot successfully landed the jet.

How to Fly with a Broken Wing

The Story of Self-Repairing Flight Control Systems



By James E. Tomayko
Edited by Christian Gelzer

Dryden Historical Study No. 1

NASA Technical Memorandum 112198

Direct Adaptive Aircraft Control Using Dynamic Cell Structure Neural Networks

Charles C. Jorgensen, Ames Research Center, Moffett Field, California

May 1997

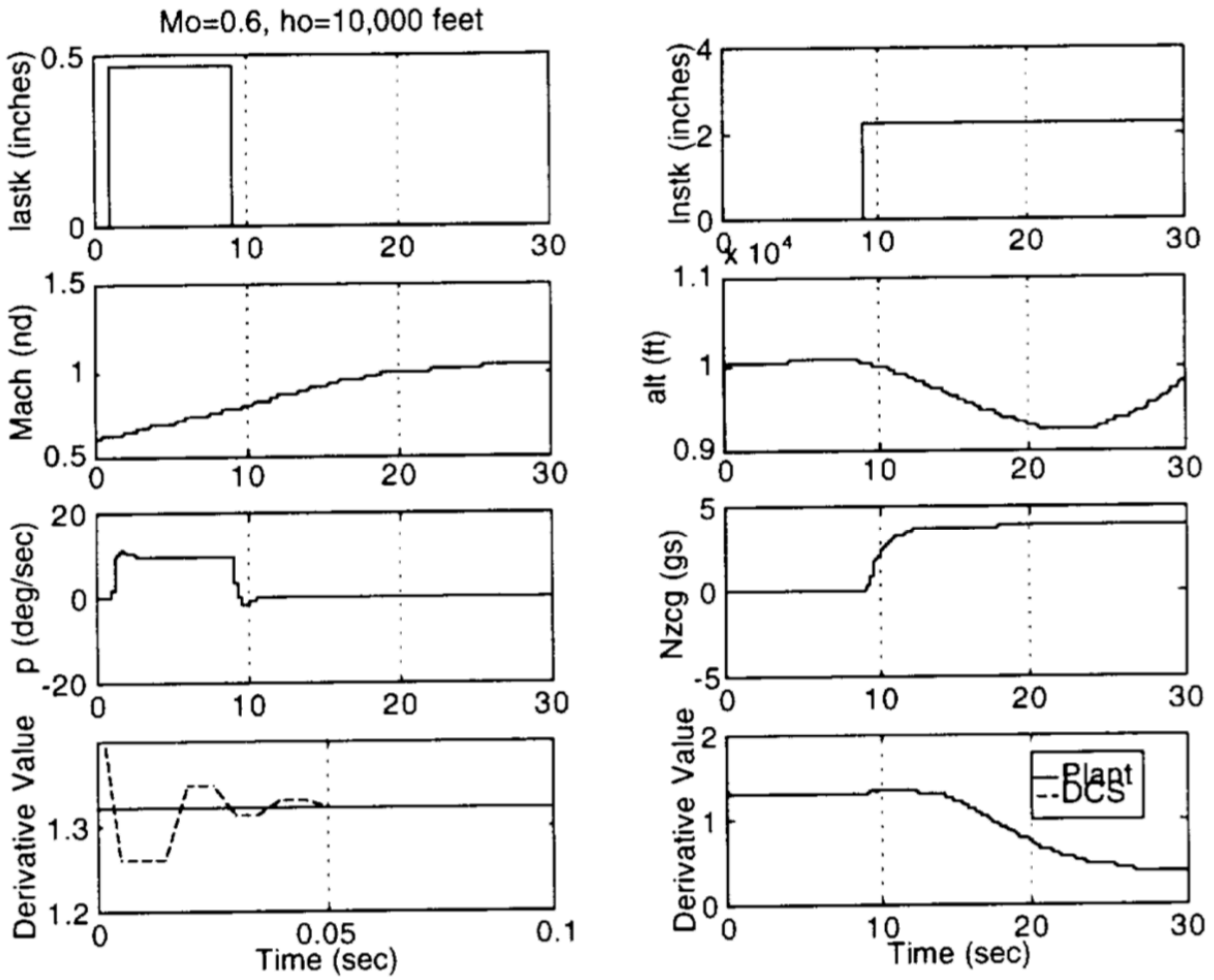


Figure 5(b). Learning parameter changes.

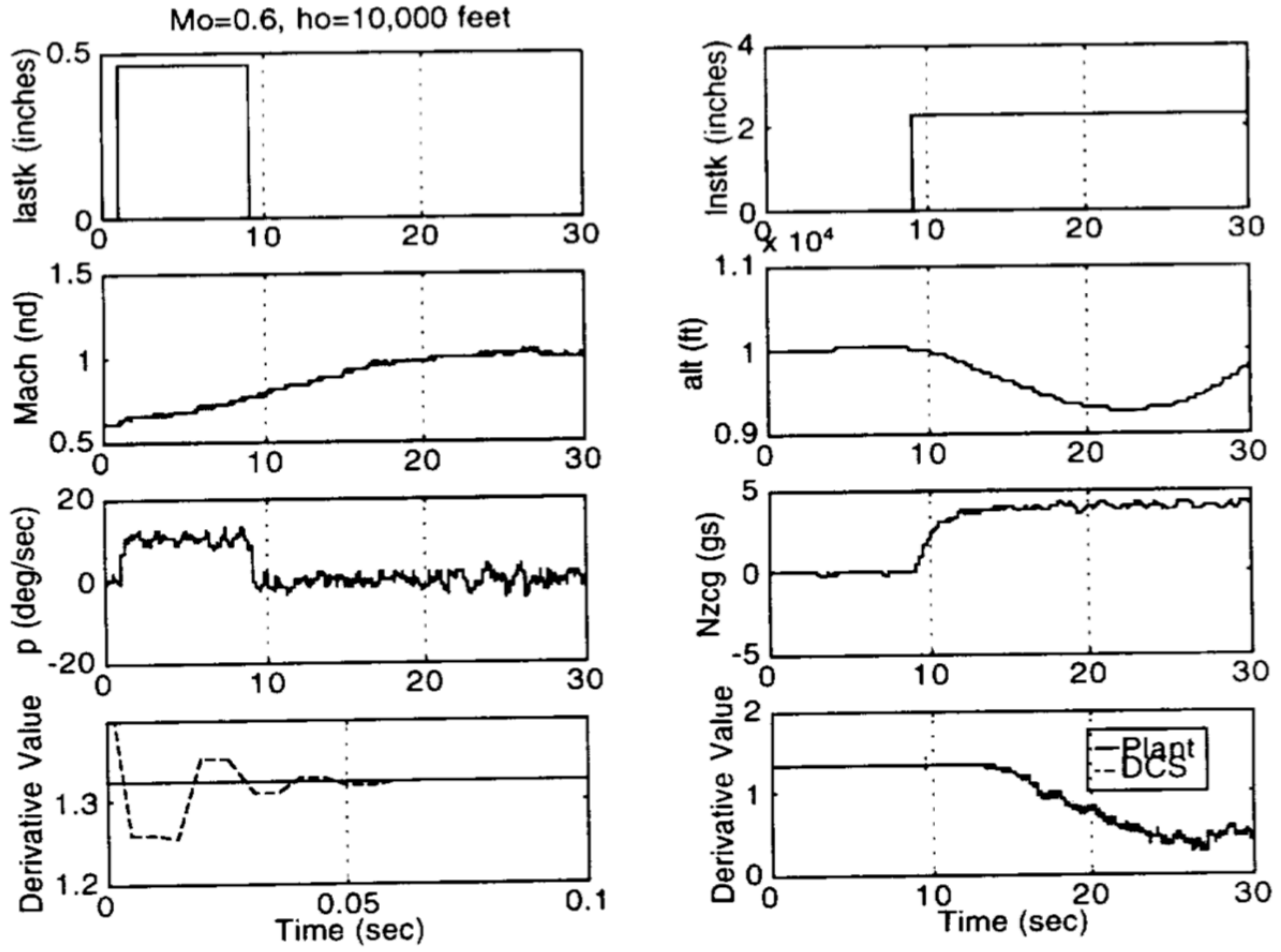


Figure 5(c). Learning with turbulence.

NASA Technical Memorandum 112197

Neural Network Prediction of New Aircraft Design Coefficients

Magnus Nørgaard, Institute of Automation, Technical University of Denmark
Charles C. Jorgensen, Ames Research Center, Moffett Field, California
James C. Ross, Ames Research Center, Moffett Field, California

May 1997

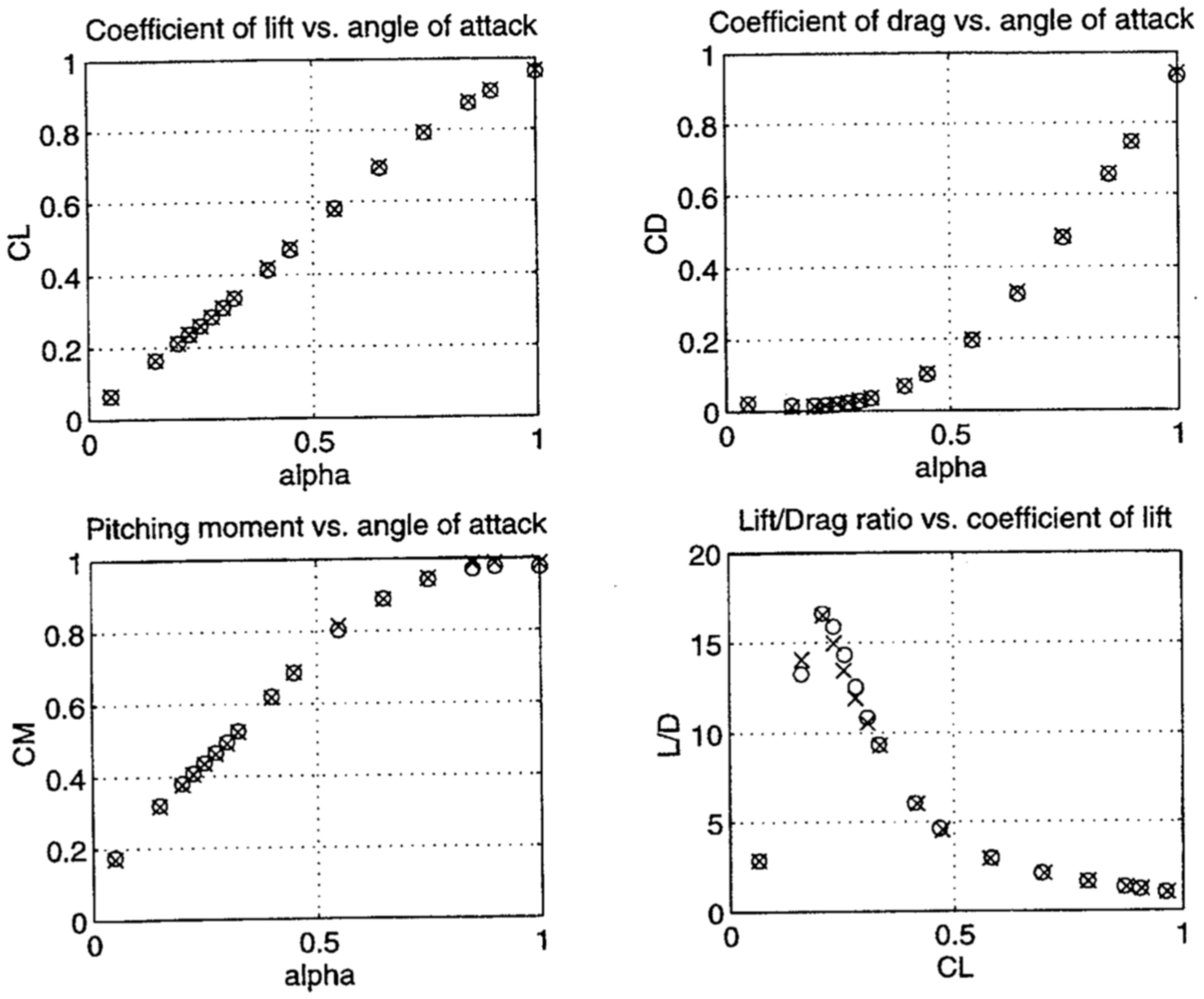
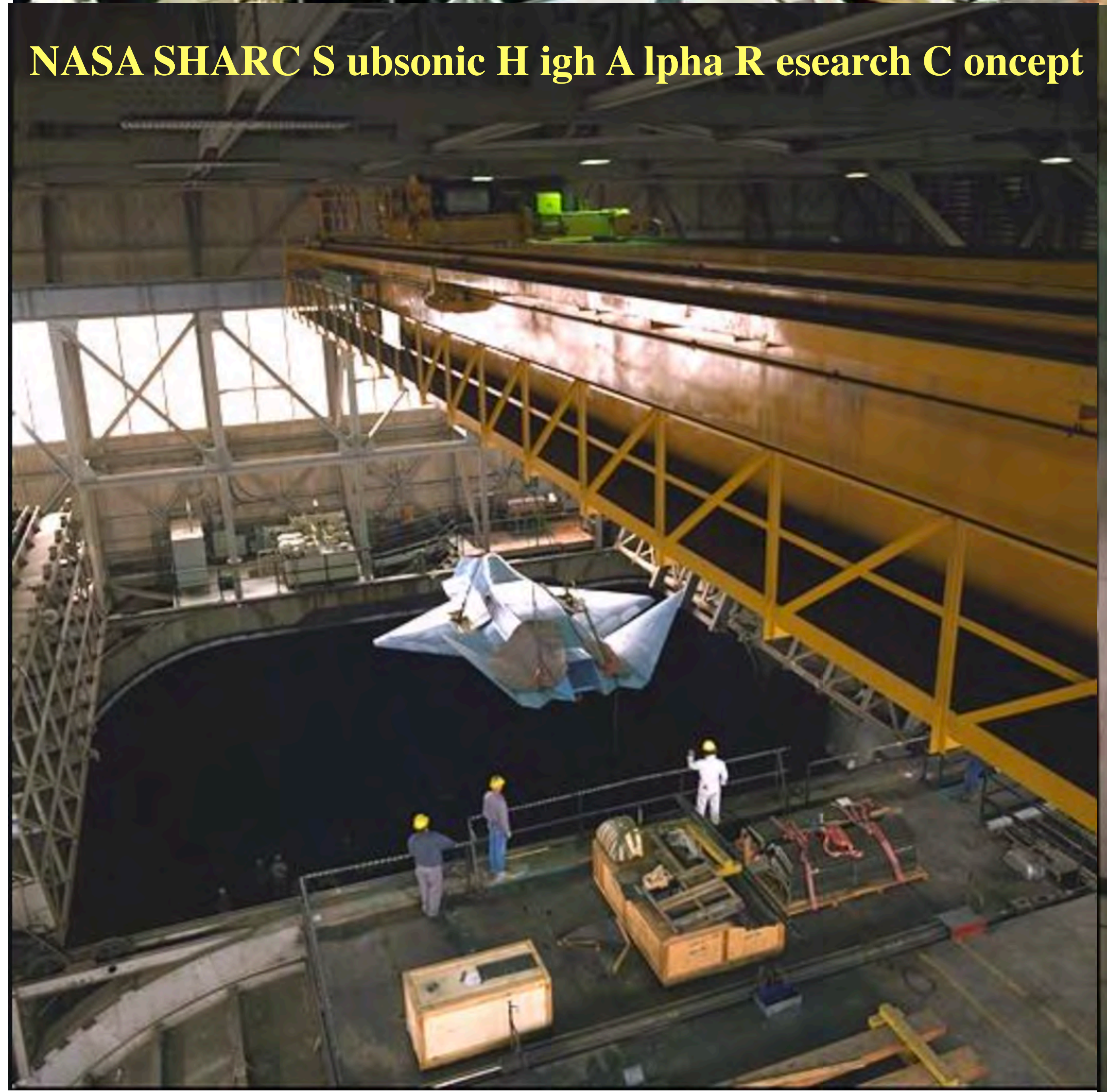


Figure 4. Comparison of test data and network predictions for LE=0.50 and TE=0.0.



nature

Letter | Published: 09 October 1986

Learning representations by back-propagating errors

David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams

Learning representations by back-propagating errors | Nature
https://www.nature.com › letters
by DE Rumelhart - 1986 - Cited by 18783 - Related articles
Oct 9, 1986 - We describe a new **learning** procedure, **back-propagation**, for networks of neurone-like units. ... Rumelhart, D. E., Hinton, G. E. & Williams, R. J. in Parallel Distributed Processing: Explorations in the Microstructure of Cognition.

Annu. Rev. Fluid Mech. 1993. 25 : 539–75
Copyright © 1993 by Annual Reviews Inc. All rights reserved

THE PROPER ORTHOGONAL DECOMPOSITION IN THE ANALYSIS OF TURBULENT FLOWS

Gal Berkooz, Philip Holmes, and John L. Lumley

The Proper Orthogonal Decomposition in the Analysis of ...
https://www.annualreviews.org › doi › abs › annurev.fl.25.010193.002543
by G Berkooz - 1993 - Cited by 2969 - Related articles
Annual Review of **Fluid Mechanics**. Vol. 25:539-575 (Volume publication date January 1993)
https://doi.org/10.1146/annurev.fl.25.010193.002543. G Berkooz, P ...

POD and Linear PCA

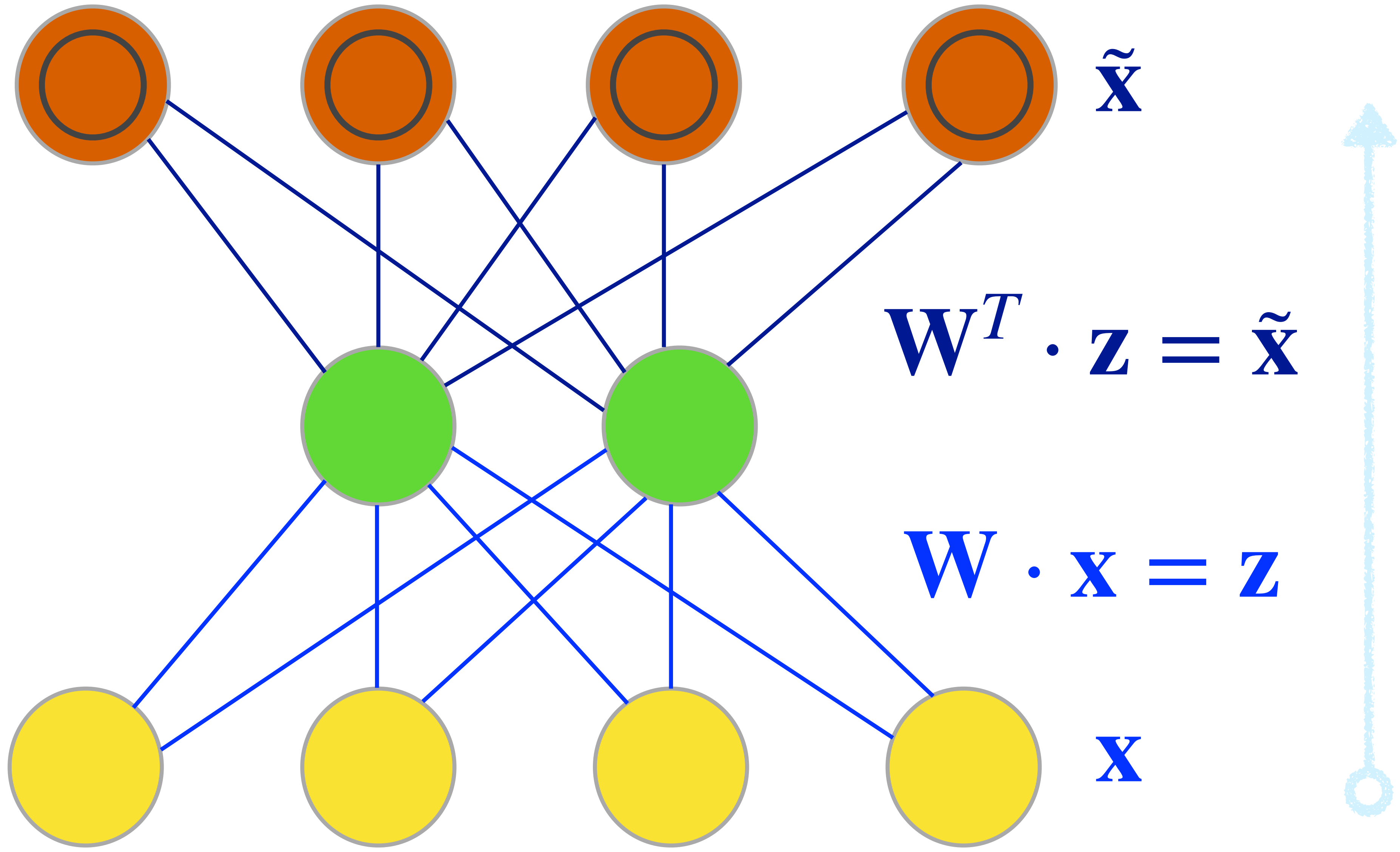
$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n$$

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

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

`retain M < D eigenvectors`

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



Neural Networks, Vol. 2, pp. 53–58, 1989
Printed in the USA.
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)

(adapted from C. Bishop)


Understanding Neural Networks
Volume 1: Basic Networks
Maureen Caudill and Charles Butler
Macintosh version 1.0
Minimum: MacPlus with 1.0 MB RAM
© 1992 Maureen Caudill
The MIT Press
Massachusetts Institute of Technology
Cambridge, Massachusetts 02142

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

(5) *To summarize, the advantage of the approach, over classical POD, seems to be on a better data compression; advantage which is lost due to a more complex process necessary in the least-square minimization problem solved in the neural net problem. Finally, I think the approach needs more confrontation with more general configurations to be effective.*

Since the proposed approach is an extension of POD, it appeared natural to compare it to the POD directly for a standard benchmark problem. Once again, the good compression and modeling results shown in the paper should be taken as an encouragement to work further on this model, in order to make it more efficient.

NEAR WALL TURBULENT FLOW

ORIGINAL POD NN

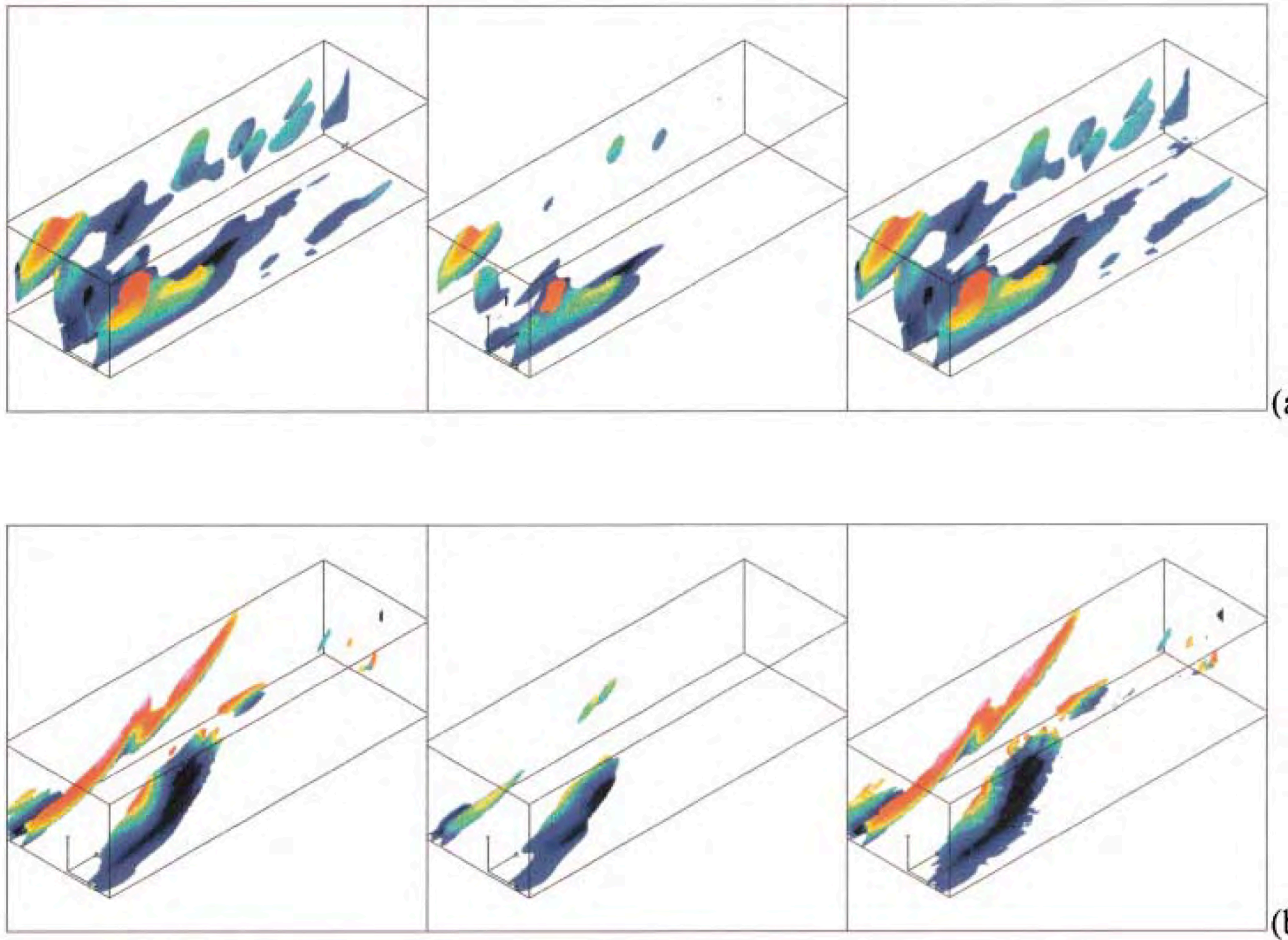
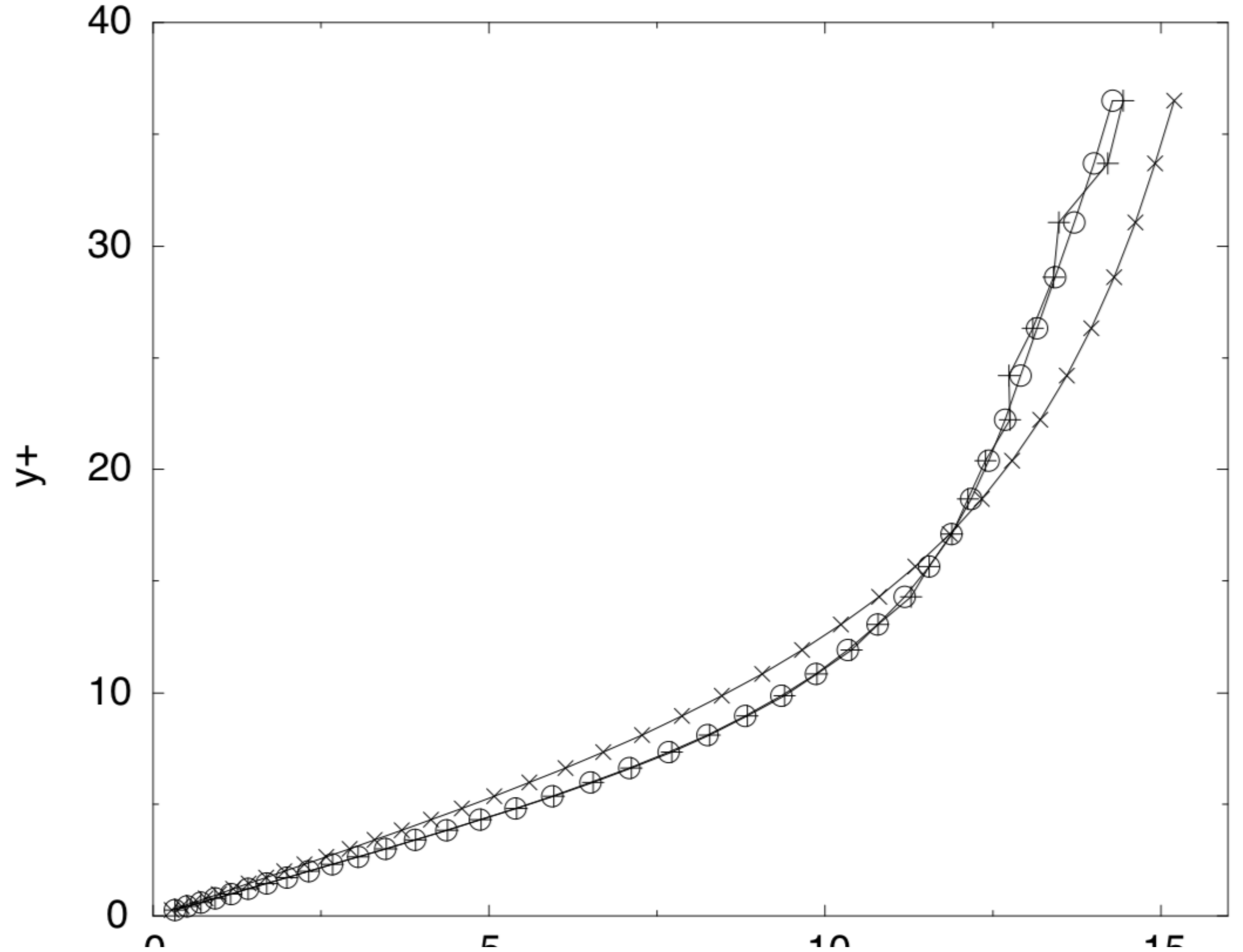


FIG. 1. Layer representation of a nonlinear neural network structure.

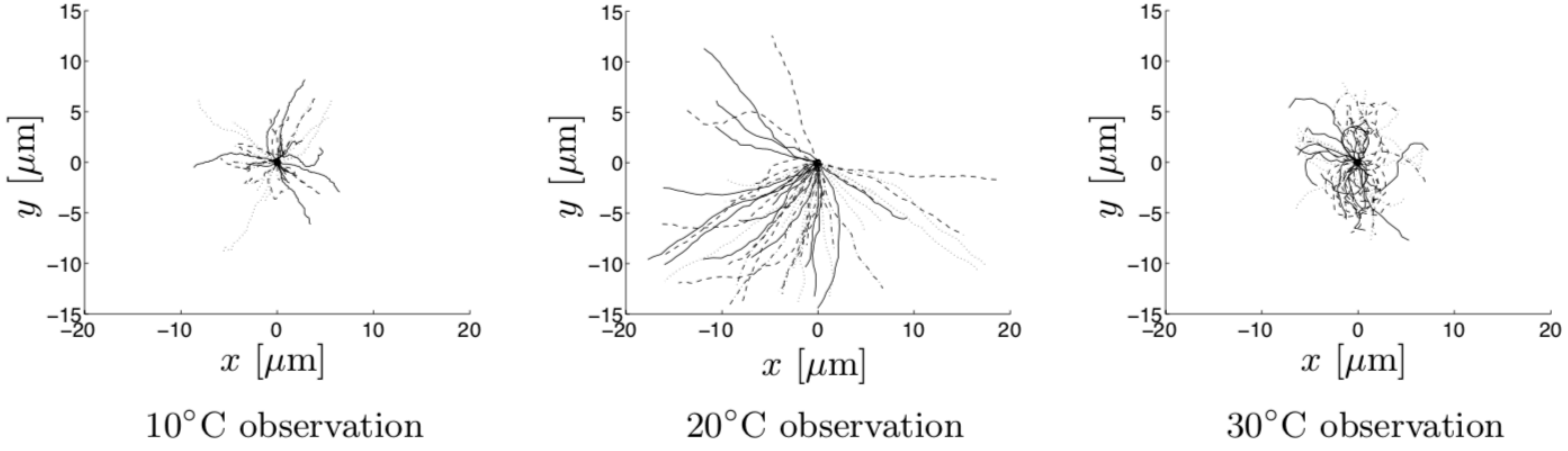


$$u(\mathbf{x},t) = \omega_z^w y + \frac{Re}{2}\frac{\partial P^w}{\partial x} y^2 + \mathbf{M}(P^w, S^w)$$

Center for Turbulence Research
Proceedings of the Summer Program 2002

Machine learning for biological trajectory classification applications

By Ivo F. Sbalzarini†, Julie Theriot ‡ AND Petros Koumoutsakos ¶



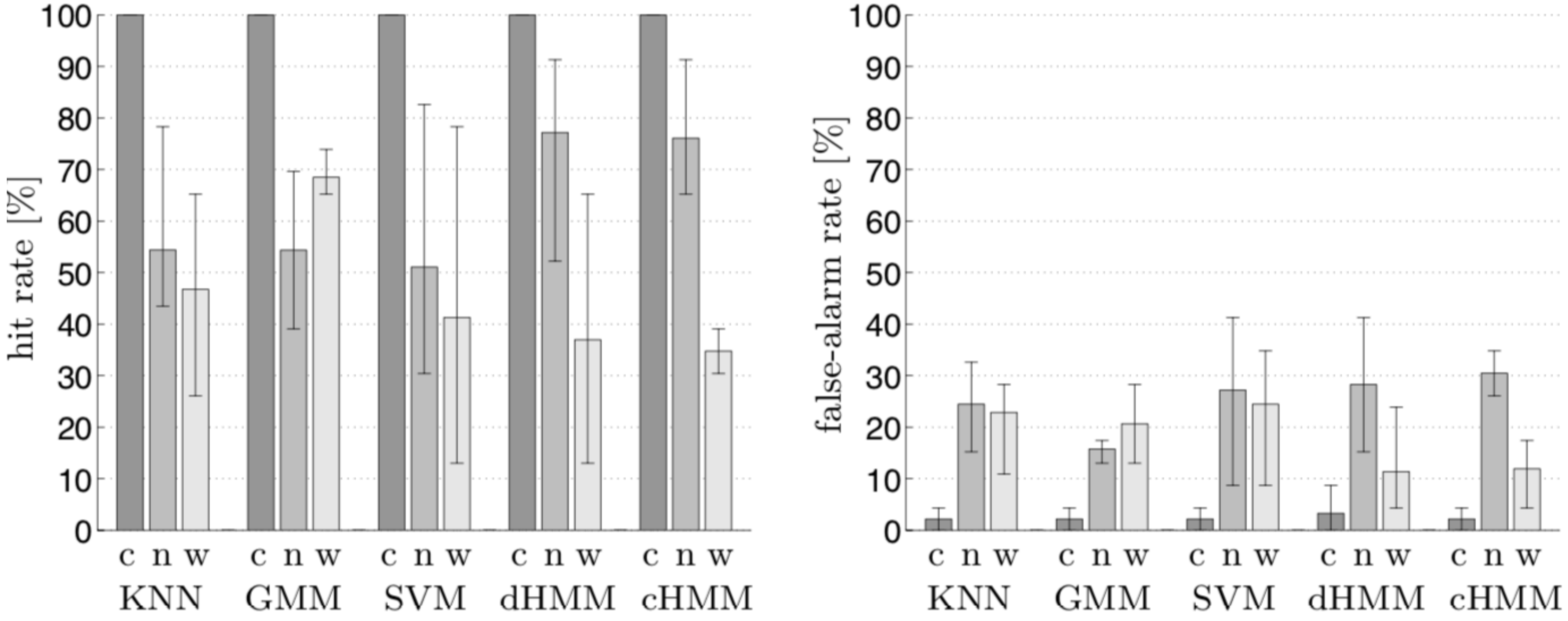


FIGURE 5. Hit and false-alarm rates for all classifiers. The percentage of hits (left) and false alarms (right) on the temperature data set is shown for each classifier in each of the 3 temperature classes: 10°C ("c"), 20°C ("n") and 30°C ("w"). The error bars range from the smallest observed rate to the largest one (min–max bars).

What is Intelligence ?

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

John McCarthy

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

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

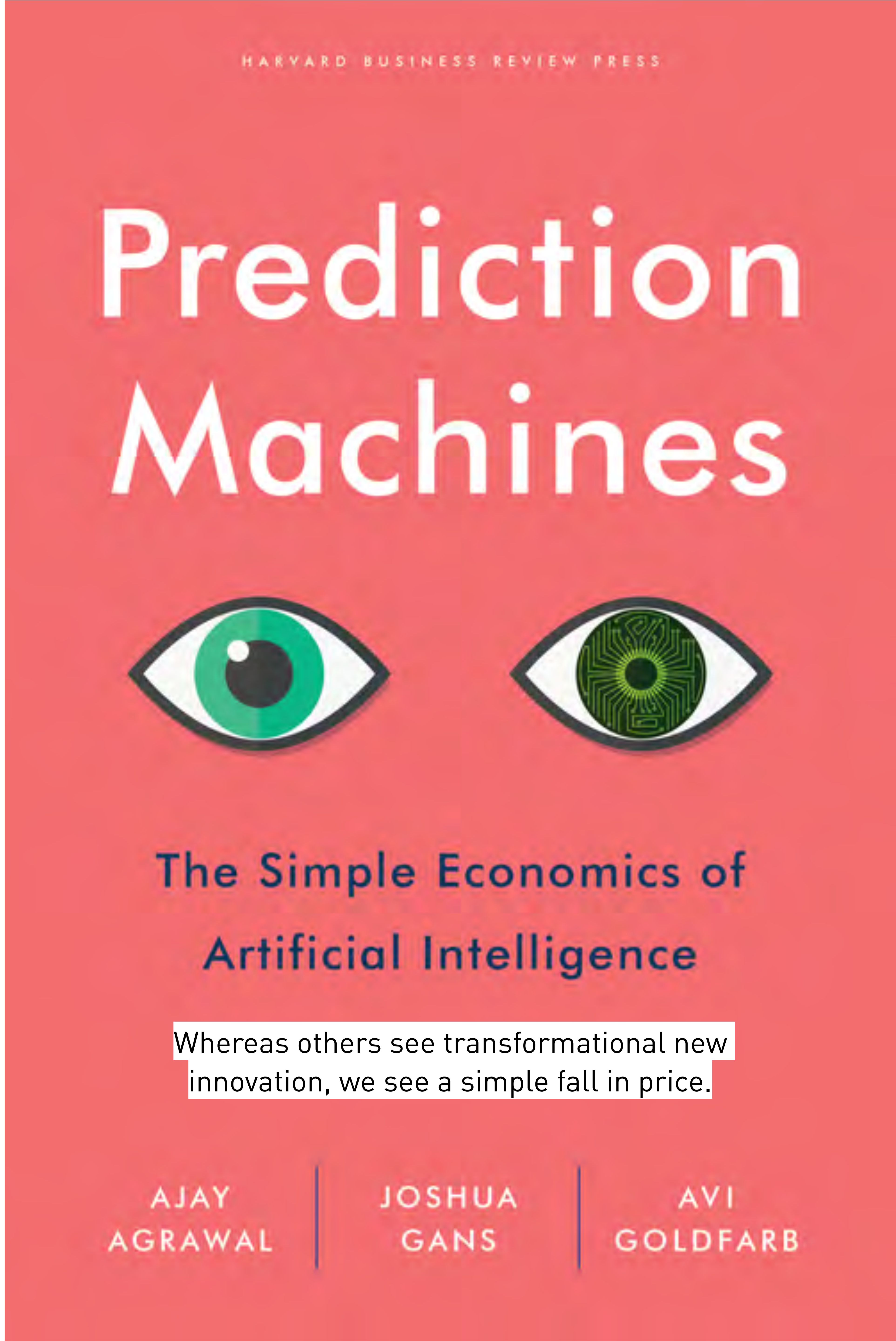
...computers have made arithmetic cheap.

Solving complex equations is done more easily and in less time ..

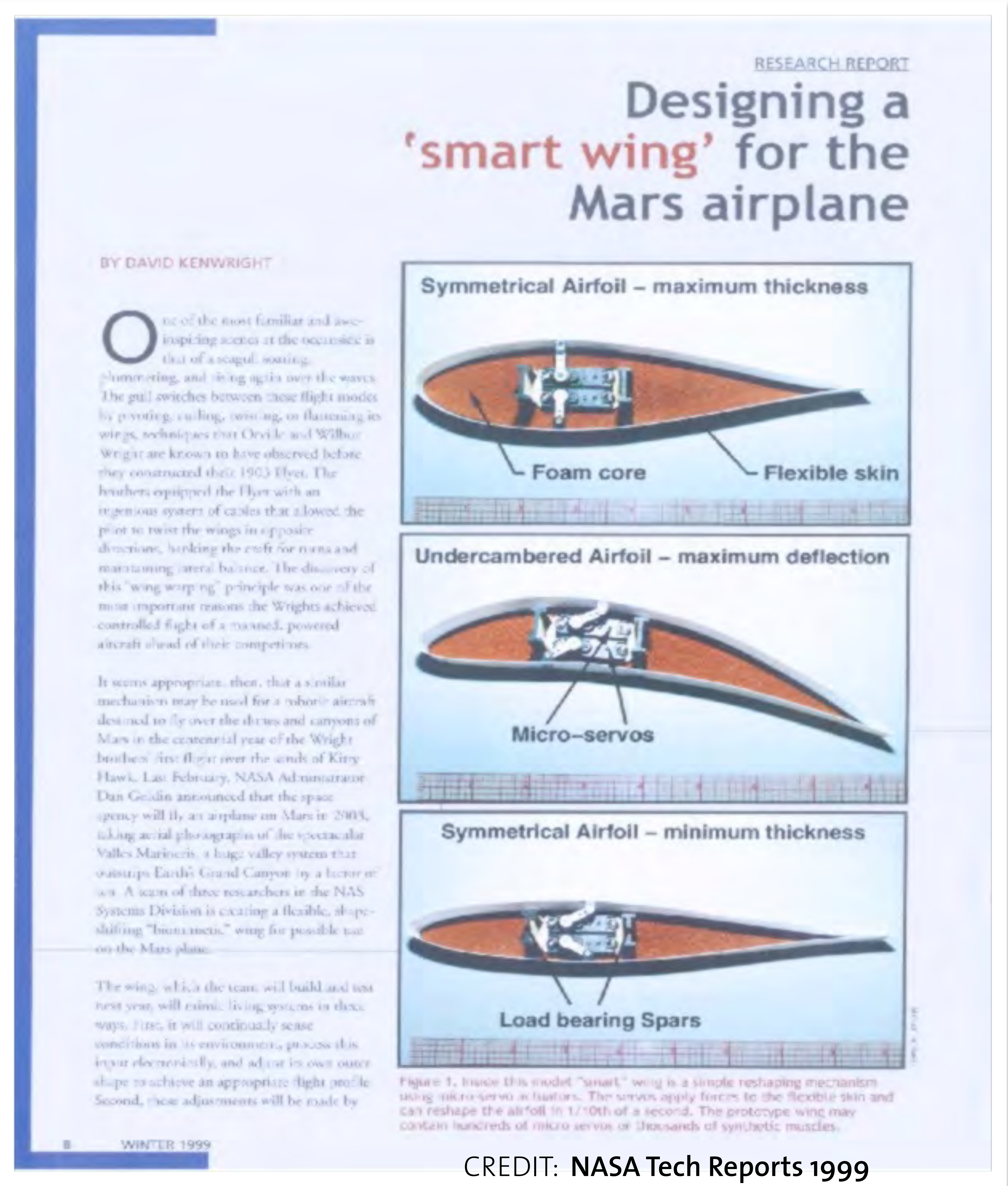
- **What will AI technology make cheap ?**
  Prediction.

- **Prediction is central to decision-making under uncertainty**

- **Better prediction under uncertainty -> new opportunities for all companies**



DESIGN ADAPTIVITY: Self-Optimizing Machines



RESEARCH REPORT

Designing a 'smart wing' for the Mars airplane

BY DAVID KENWRIGHT

One of the most familiar and awe-inspiring scenes at the oceanside is that of a seagull soaring, plummeting, and rising again over the waves. The gull switches between these flight modes by pivoting, curling, twisting, or flattening its wings, techniques that Orville and Wilbur Wright are known to have observed before they constructed their 1903 Flyer. The brothers equipped the Flyer with an ingenious system of cables that allowed the pilot to twist the wings in opposite directions, banking the craft for turns and maintaining lateral balance. The discovery of this "wing warping" principle was one of the most important reasons the Wrights achieved controlled flight of a manned, powered aircraft ahead of their competitors.

It seems appropriate, then, that a similar mechanism may be used for a robotic aircraft destined to fly over the dunes and canyons of Mars in the centennial year of the Wright brothers' first flight over the sands of Kitty Hawk. Last February, NASA Administrator Dan Goldin announced that the space agency will fly an airplane on Mars in 2003, taking aerial photographs of the spectacular Valles Marineris, a huge valley system that outstrips Earth's Grand Canyon by a factor of ten. A team of three researchers in the NAS Systems Division is creating a flexible, shape-shifting "biomimetic" wing for possible use on the Mars plane.

The wing, which the team will build and test next year, will mimic living systems in three ways. First, it will continually sense conditions in its environment, process this input electronically, and adjust its own outer shape to achieve an appropriate flight profile. Second, these adjustments will be made by

Symmetrical Airfoil – maximum thickness

Foam core

Flexible skin

Undercambered Airfoil – maximum deflection

Micro-servos

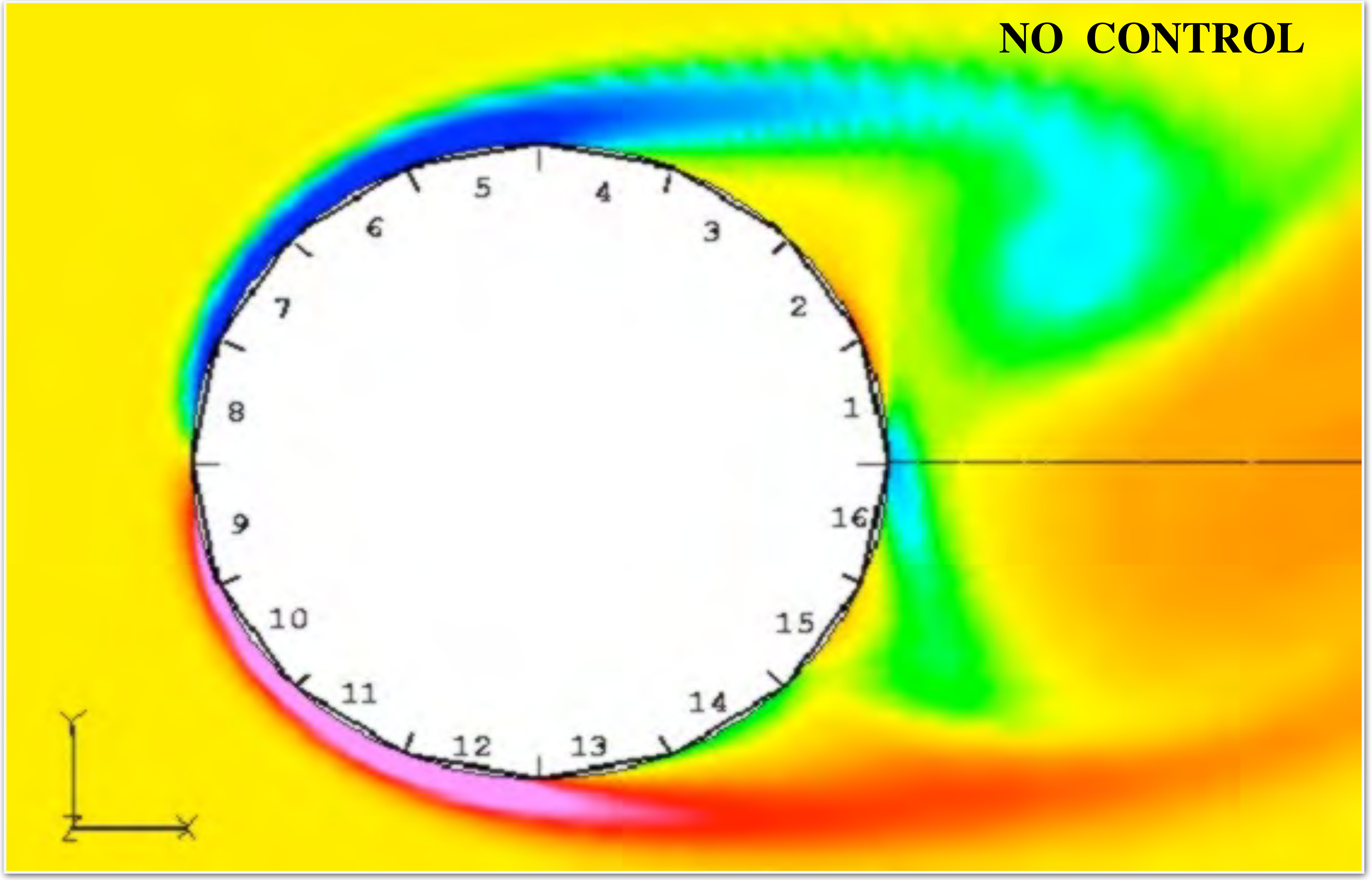
Symmetrical Airfoil – minimum thickness

Load bearing Spars

Figure 1. Inside this model "smart" wing is a simple reshaping mechanism using micro-servo actuators. The servos apply forces to the flexible skin and can reshape the airfoil in 1/10th of a second. The prototype wing may contain hundreds of micro servos or thousands of synthetic muscles.

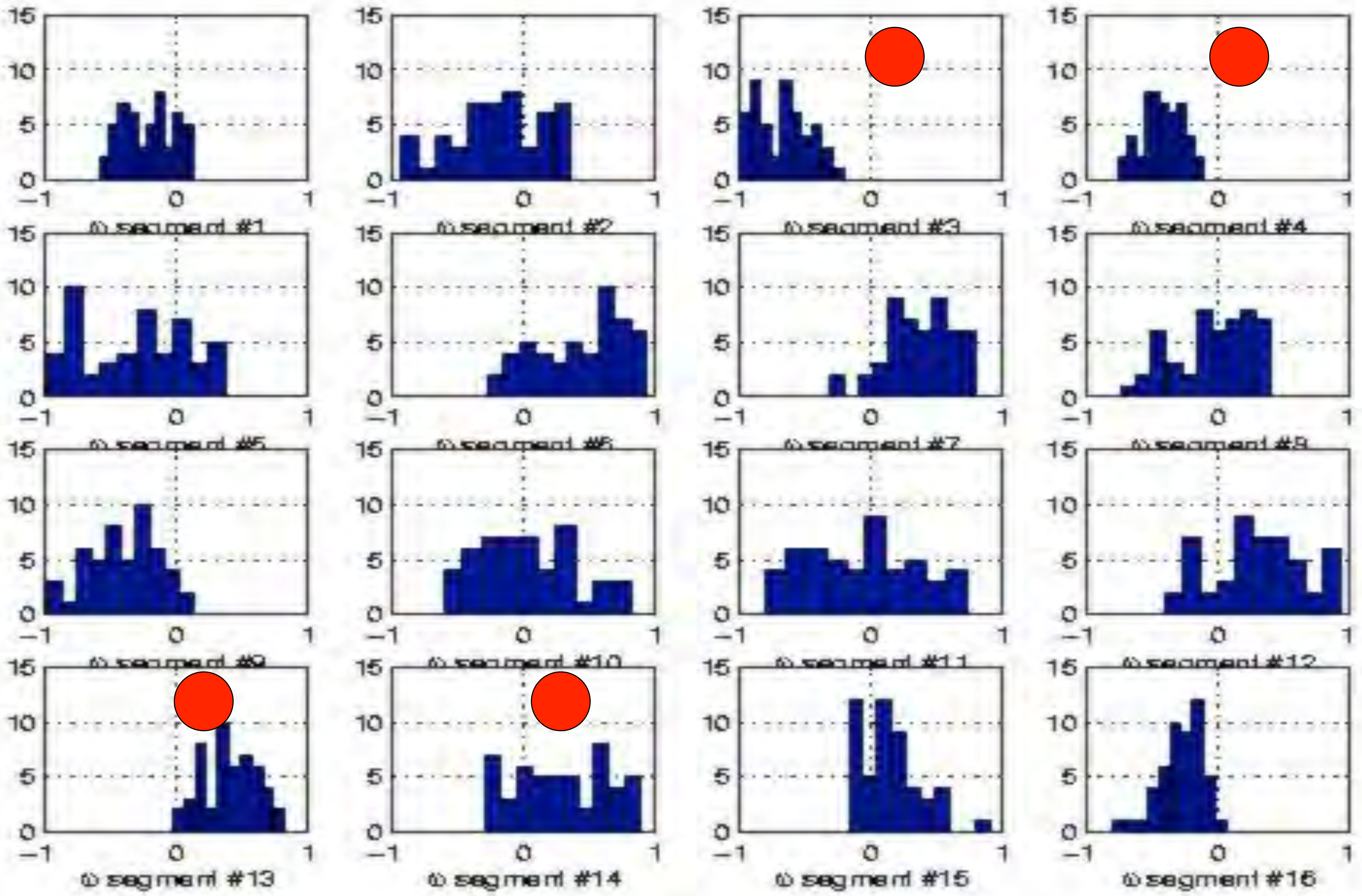
CREDIT: **NASA Tech Reports 1999**

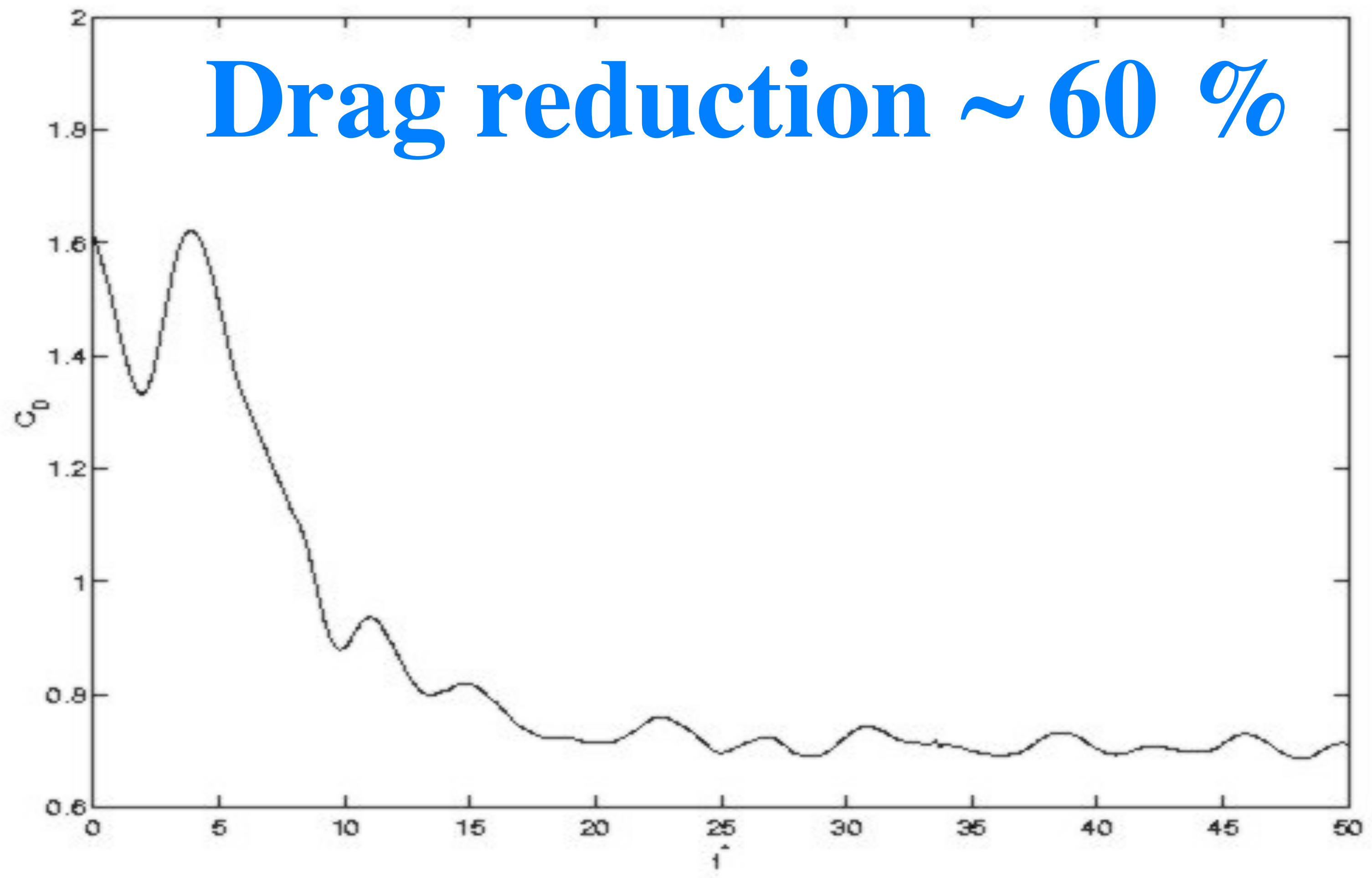
Actuation on Each Panel: Steady Tangential Wall Motion

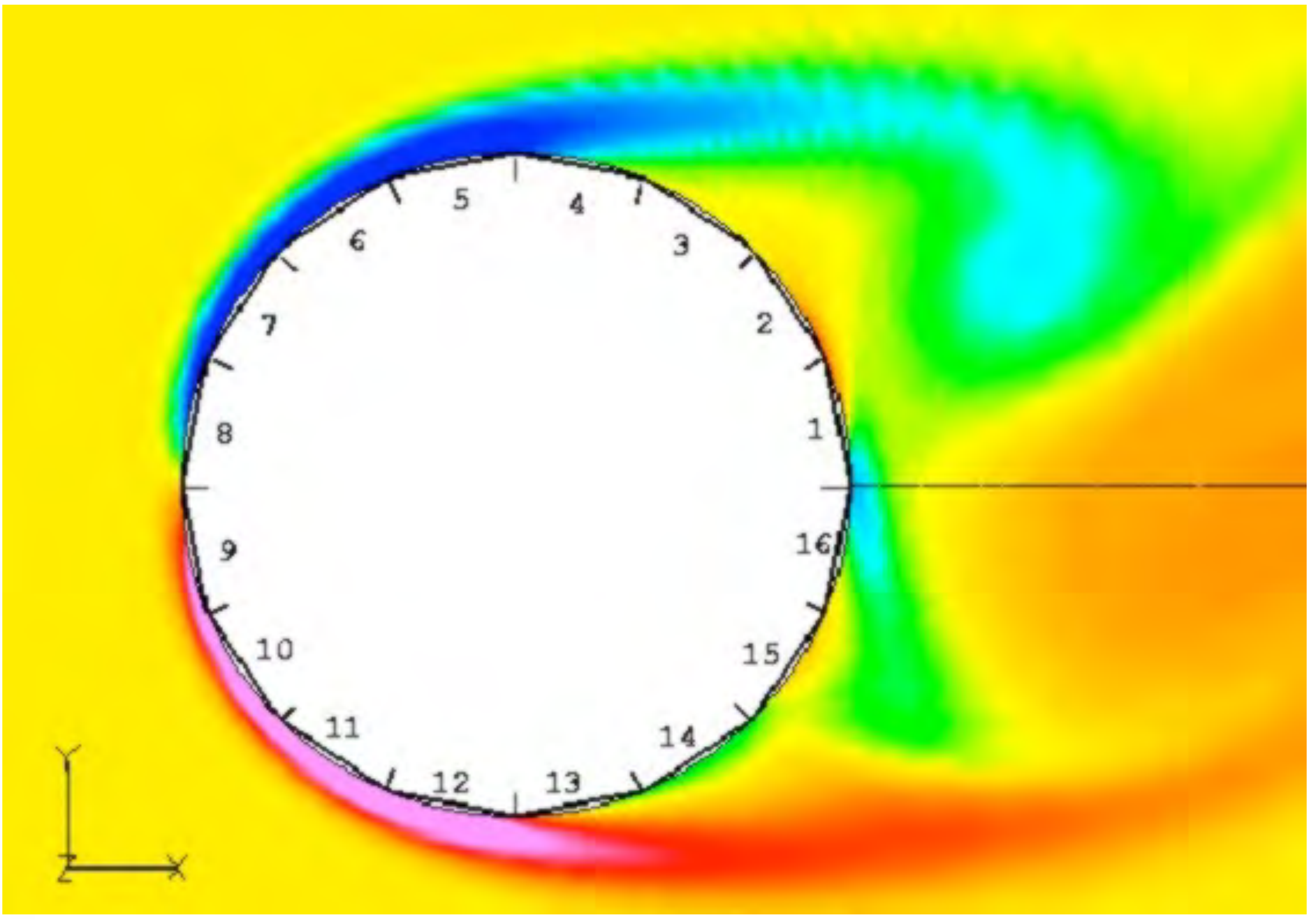


$$\mathbf{X^g} = (x_1^g, x_2^g, ..., x_{16}^g) \quad \text{with} \ \ x_i^g \in [-1, +1], \quad i = 1, ..., 16$$

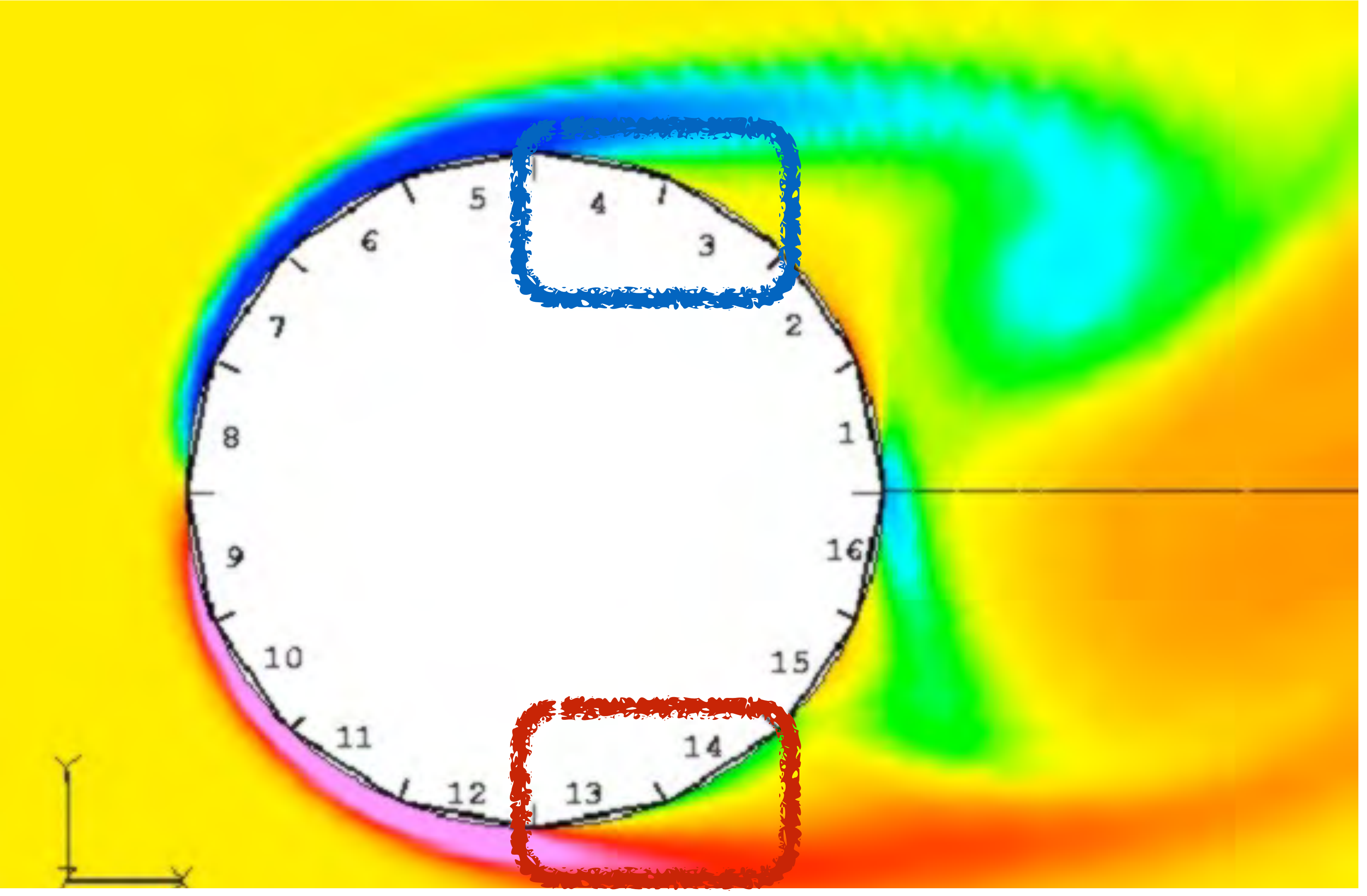
Histograms of population values (over each panel)

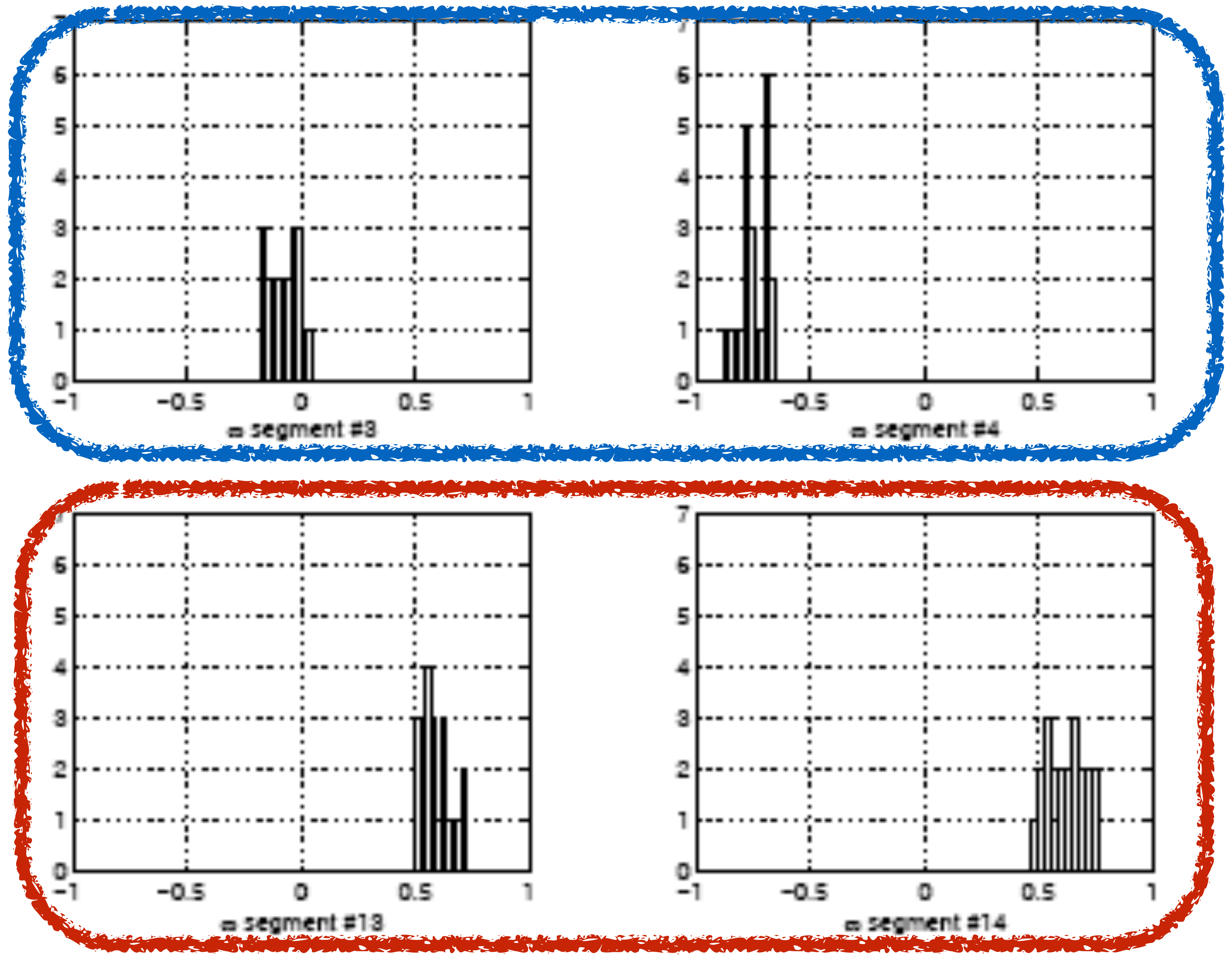




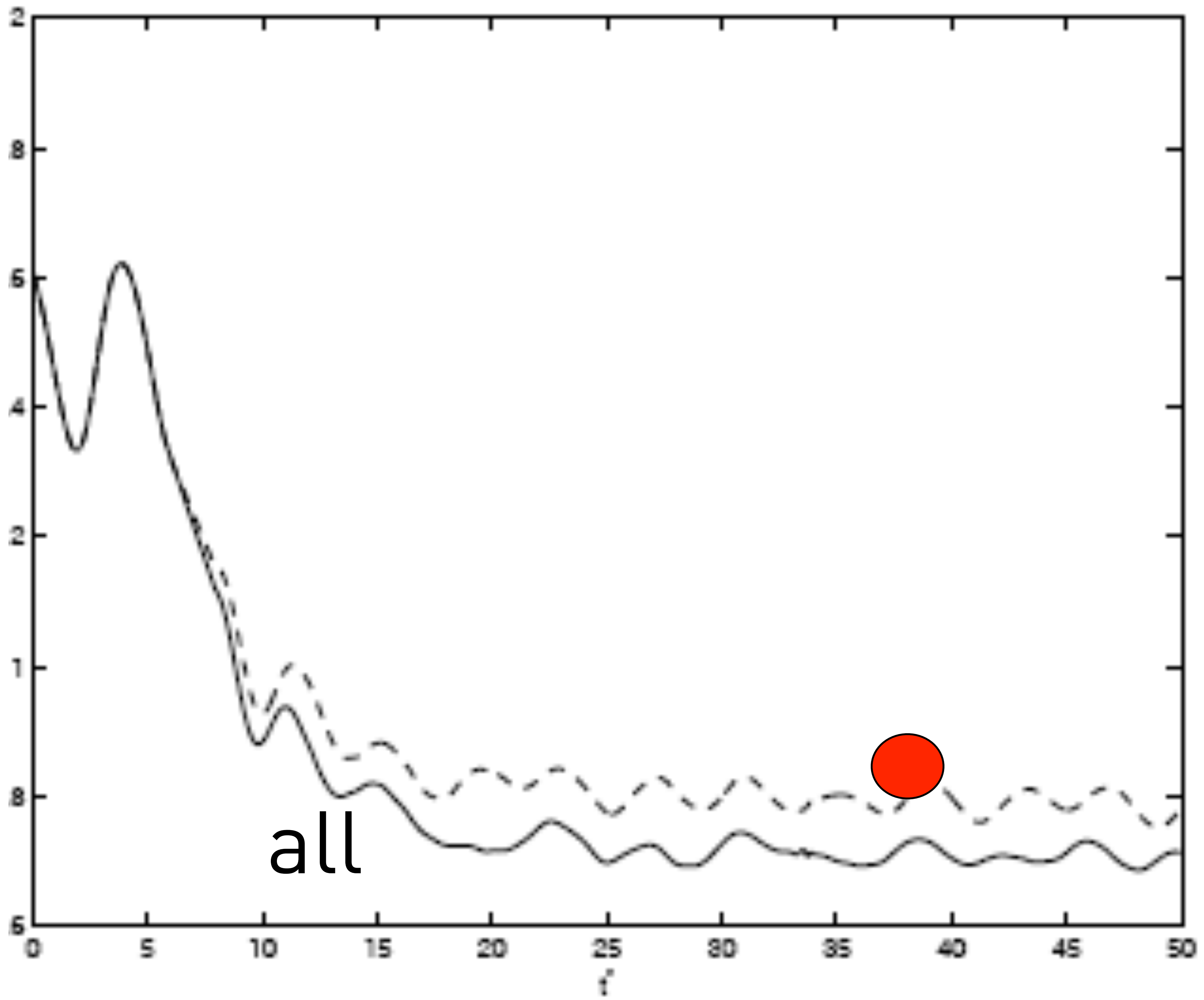


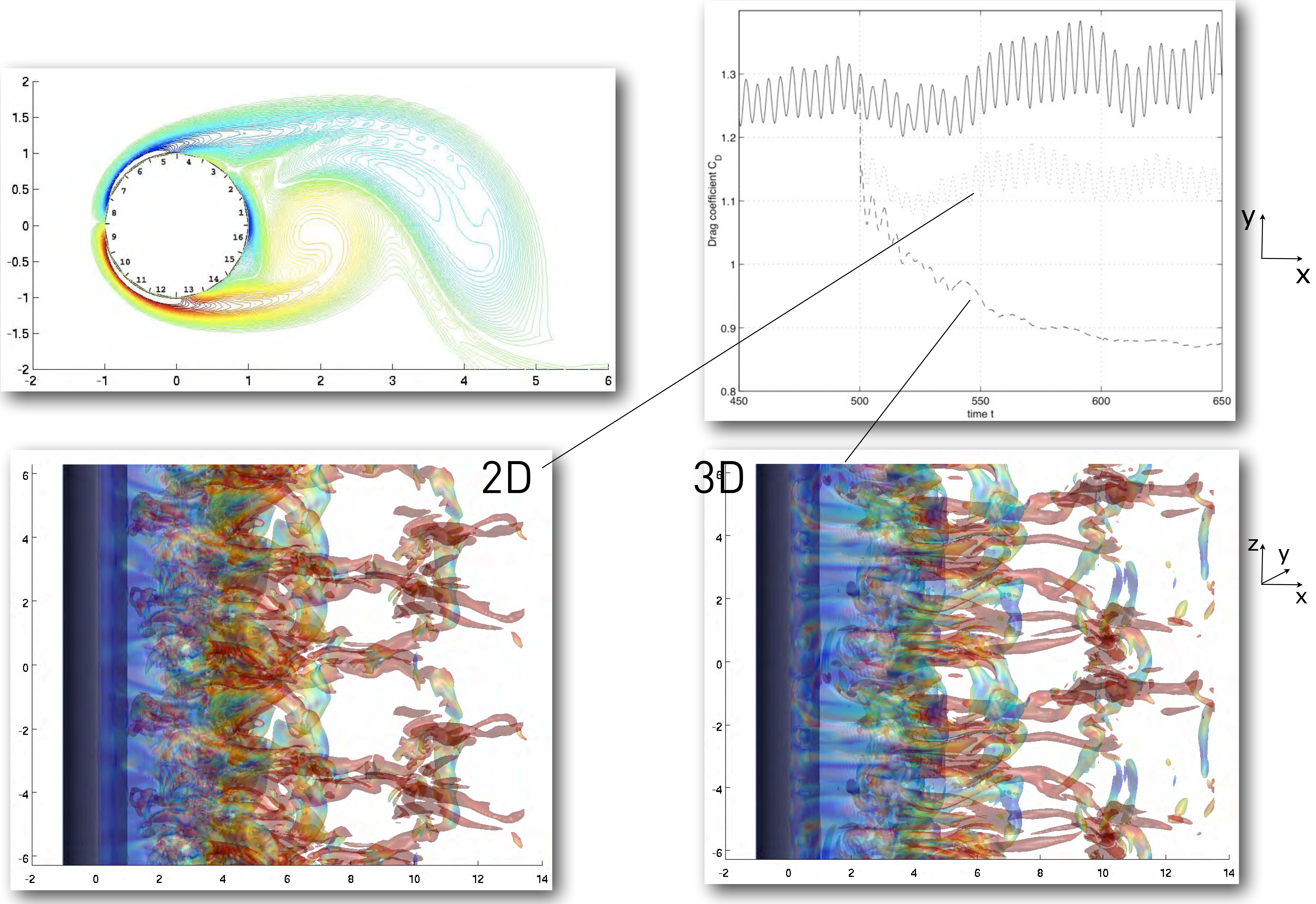
Identify and Optimize **Critical** actuators





M. Milano, P. Koumoutsakos, and J. Schmidhuber, "Self-organizing nets for optimization," *IEEE Trans. on neural networks*, vol. 15, iss. 3, 2004.





Poncet Ph., Hildebrand R., Cottet G.H., Koumoutsakos P., Spatially distributed control for optimal drag reduction of the flow past a circular cylinder, **J. Fluid Mechanics**, 599, 111-120, 2008