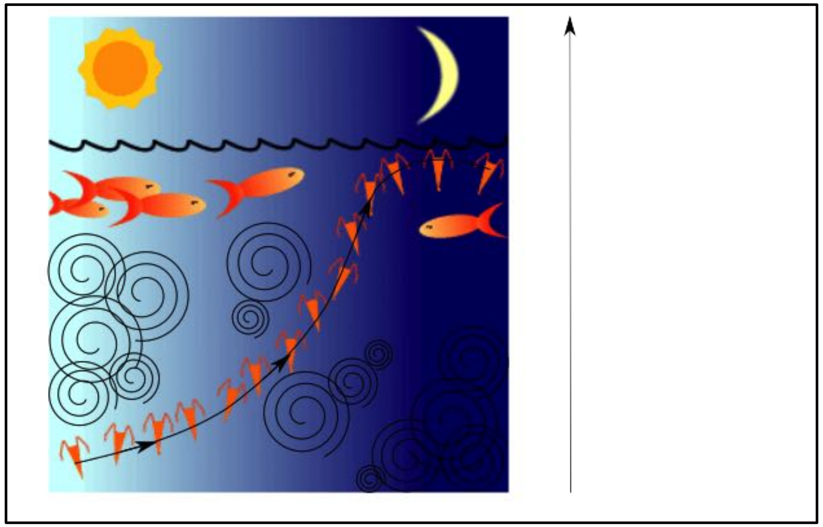
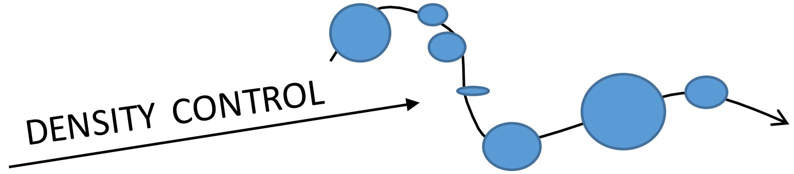
Flow navigation by smart particles via Reinforcement Learning

Luca Biferale
Dept. Physics, INFN & CAST
University of Rome 'Tor Vergata'
biferale@roma2.infn.it
SUSTech June 2018

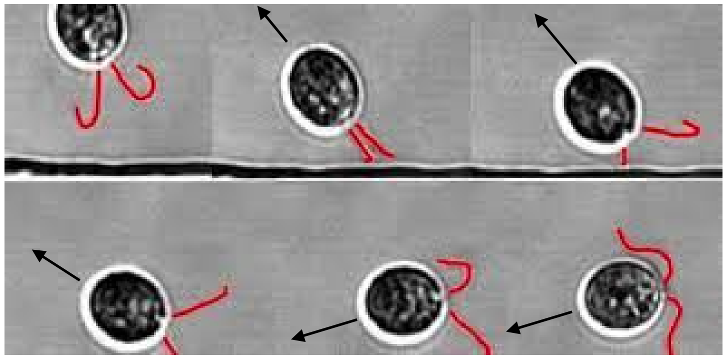


CREDITS: SIMONA COLABRESE (TOR VERGATA UNIV. ROME-IT); ANTONIO CELANI (ICTP TRIESTE-IT); KRISTIAN GUSTAVSSON (GOTHEBORG UNIV. SWEDEN)



- PARTICLES IN COMPLEX FLOWS I: **SMART INERTIAL PARTICLES**
- PARTICLES IN COMPLEX FLOWS II: **SMART MICROSWIMMERS**

SWIMMING DIRECTION CONTROL

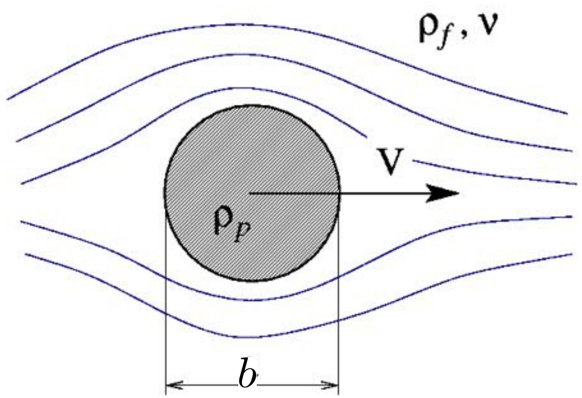


- **Flow navigation by smart microswimmers via reinforcement learning**
S Colabrese, K Gustavsson, A Celani, L Biferale
Physical Review Letters 118 (15), 158004, 2017

-**Smart Inertial Particles**
S Colabrese, K Gustavsson, A Celani, L Biferale
arXiv preprint arXiv:1711.05853, 2017

- **Finding efficient swimming strategies in a three-dimensional chaotic flow by reinforcement learning**
K Gustavsson, L Biferale, A Celani, S Colabrese
The European Physical Journal E 40 (12), 110, 2017

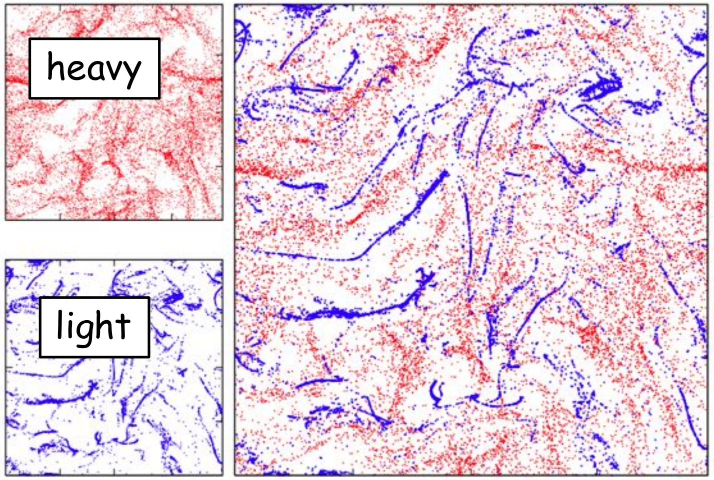
PARTICLES IN COMPLEX FLOWS I: INERTIAL PARTICLES



$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{V}$$

$$\frac{d\boldsymbol{V}}{dt} = \beta \frac{D\boldsymbol{u}(\boldsymbol{X},t)}{Dt} + \frac{\boldsymbol{u}(\boldsymbol{X},t) - \boldsymbol{V}}{\tau}$$

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \partial)\mathbf{u} = -\partial P + \nu \nabla \mathbf{u}$$



$$\beta = \frac{3\rho_f}{\rho_f + 2\rho_p}$$

$\beta<1$ heavy particles

$\beta>1$ light particles

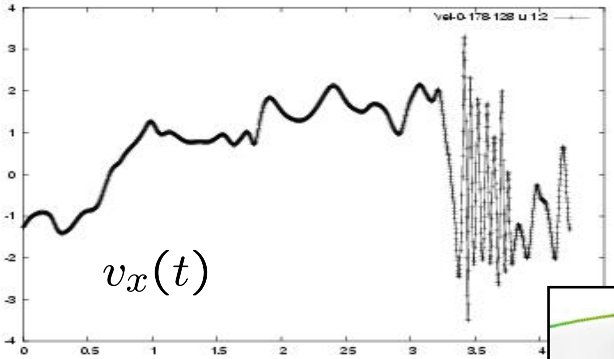
$$\tau = \frac{b^2}{3\nu\beta}$$

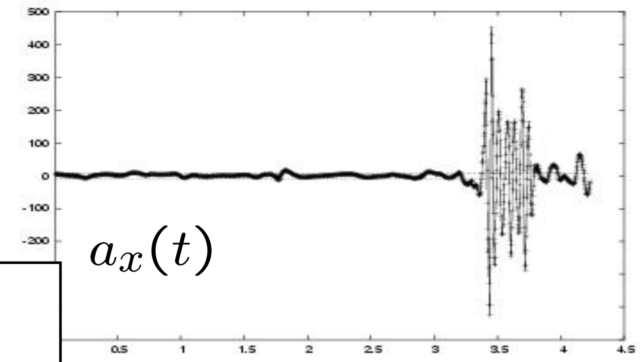
Drag: **Stokes Time**

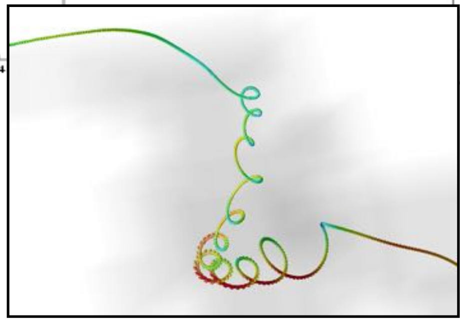
Preferential concentration!

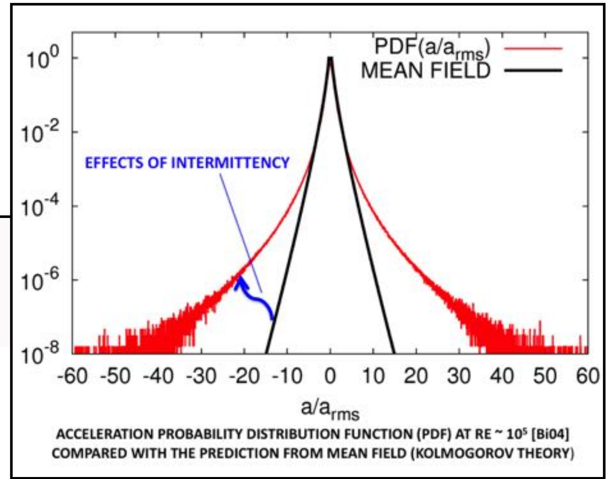
Light(heavy) particles accumulate inside(outside) highly vortical regions

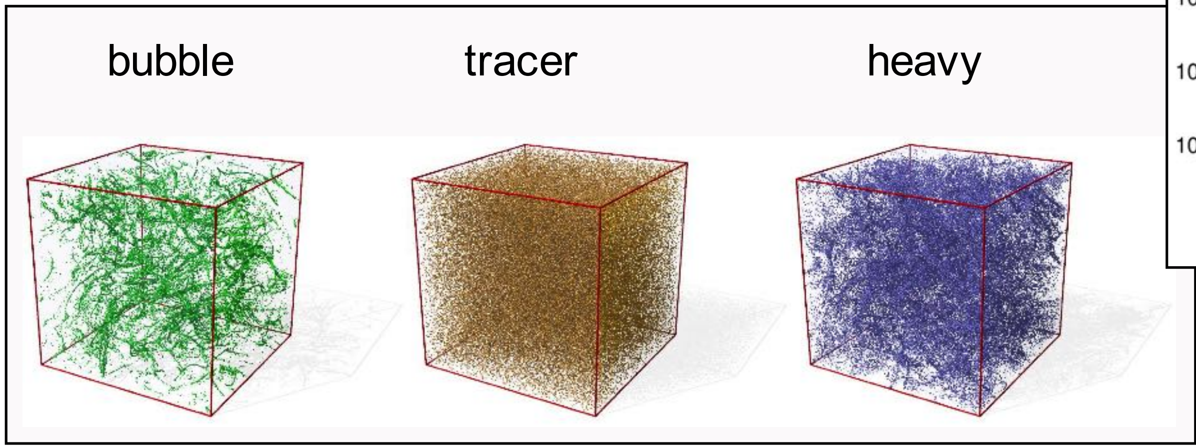
Maxey, *J. Fluid Mech.* **174**, 441 (1987); Falkovich *et al*, *Phys. Rev. Lett.* **86**, 2790 (2001)





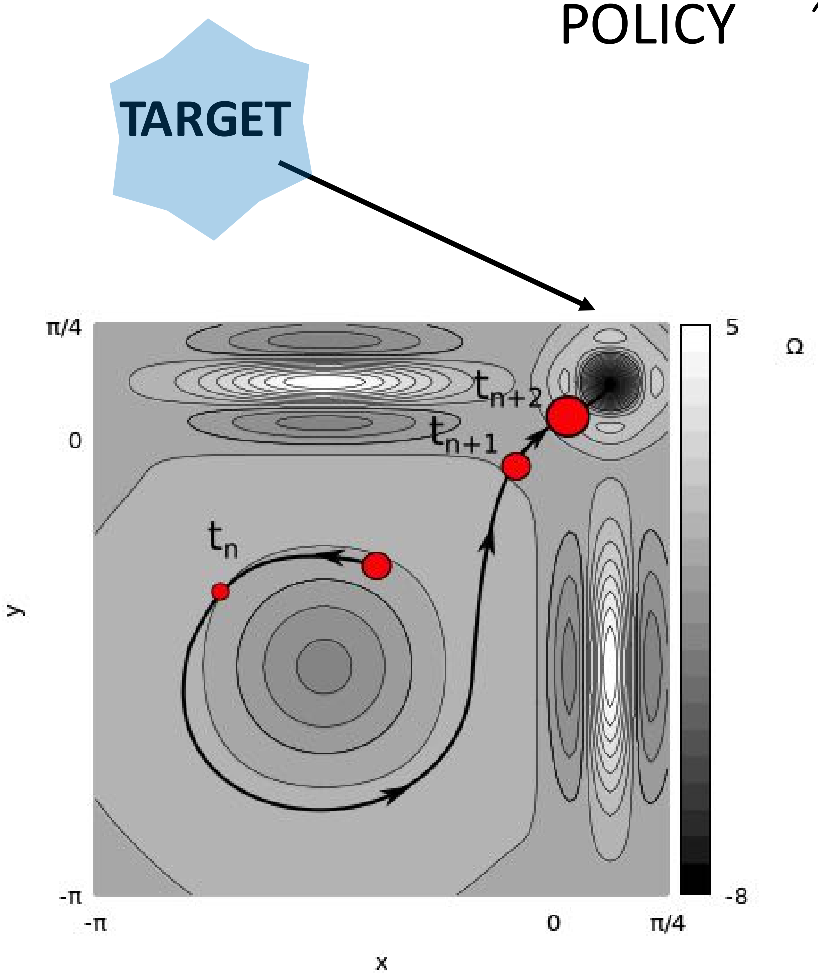


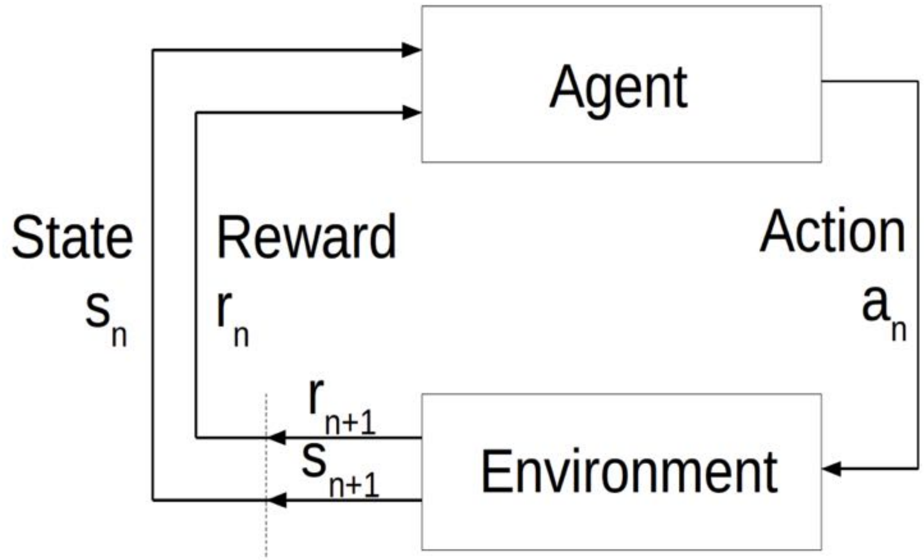




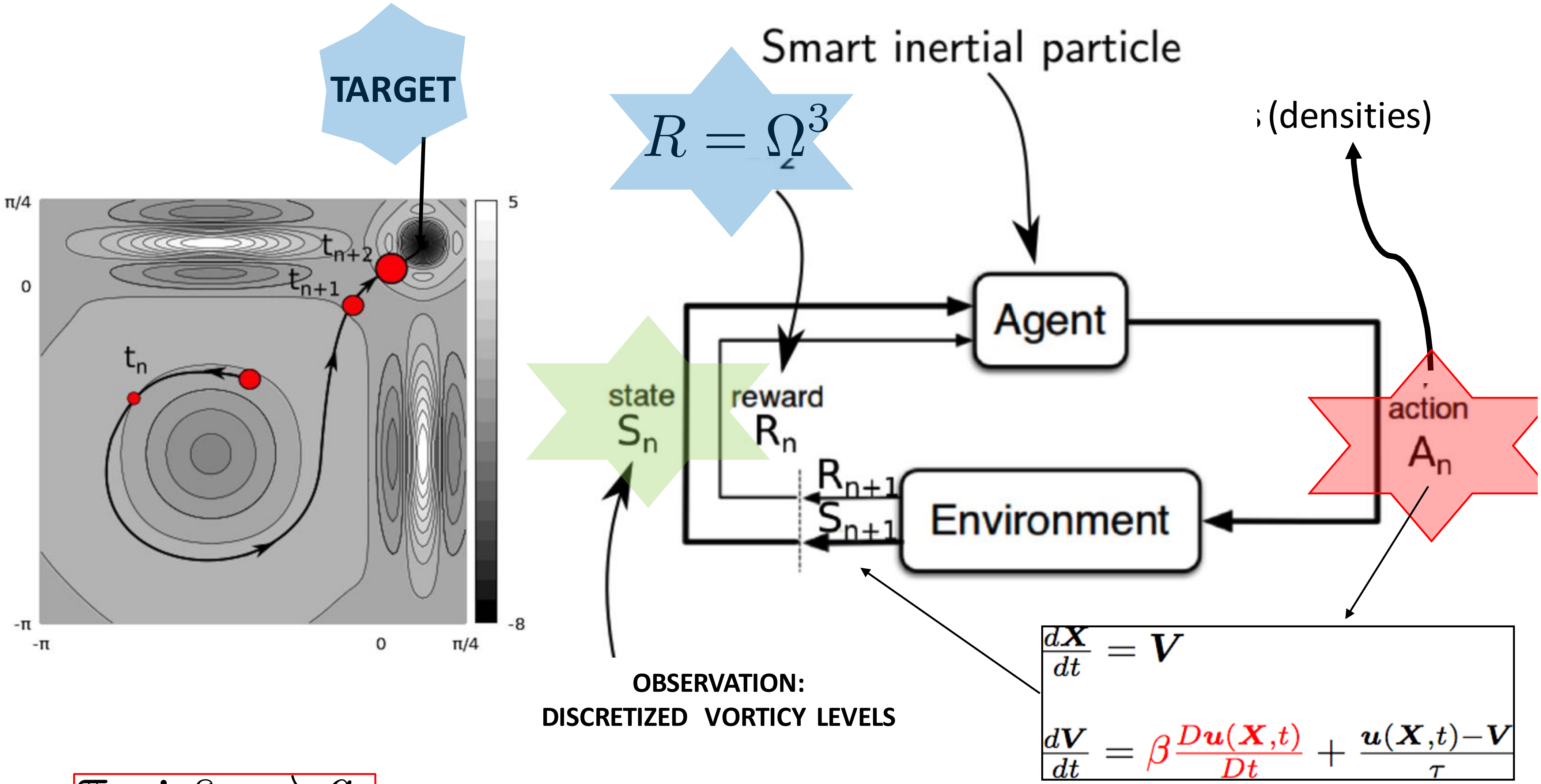
<u>**Particle trapping in three-dimensional fully developed turbulence**</u>
L.B., G Boffetta, A Celani, A Lanotte, F Toschi
Physics of Fluids 17 (2), 021701

POLICY $\pi : s \rightarrow a$



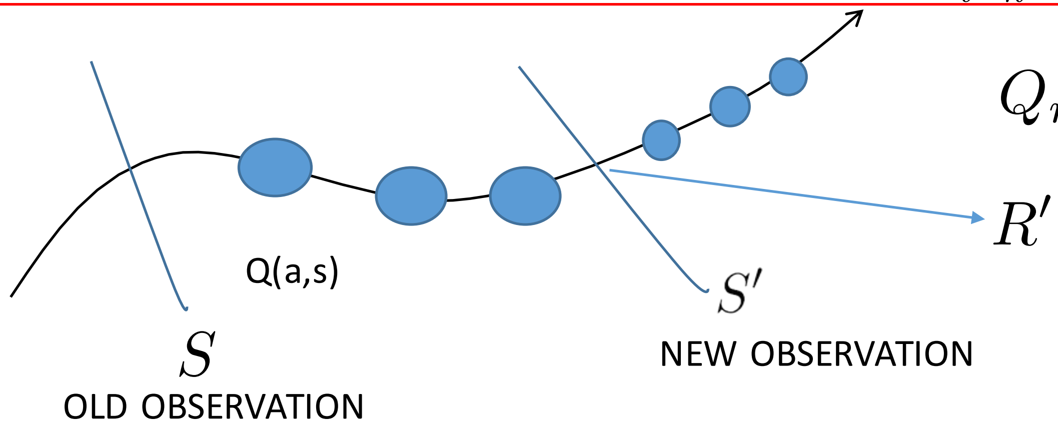


Reinforcement learning is a framework to find a good (optinal) POLICY for achieving given long-term tasks. It is widely used in artificial intelligence and machine learning. It is based on the interaction between a decision-maker (in our case the inertial particle) and the environment. The decision maker can change its behaviour in response to inputs from the system. By trial and error the decision maker progressively learns how to behave optimally



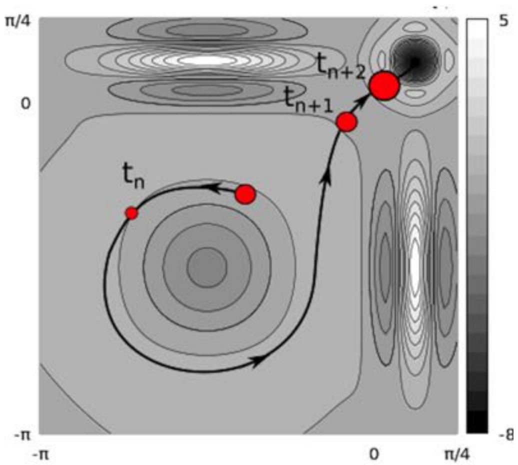
$$\pi_n : s_i \rightarrow a_j$$

$$Q_n(s_i, a_j) = R_n + \gamma R_{n+1} + \gamma^2 R_{n+2} + \gamma^3 R_{n+3} + \cdots = \sum_{t=n}^{\infty} \gamma^t R_t$$



$$Q_n(s, a) = R_n + \gamma Q_{n+1}(s', a')$$

$$Q(s, a) \leftarrow Q(s, a) + \alpha[R' + \gamma \max_{a'} Q(s', a') - Q(s, a)]$$



$$\pi_n \rightarrow \pi_{n+1} \rightarrow \cdots \pi_{opt}$$

1-step Q-LEARNING ALGORITHM

QUALITY MATRIX AT STEP n → $Q_n(a_j, s_i)$

EXPECTED DISCOUNTED FUTURE RETURN IF ACTION **a_j** is taken after observation of state **s_i**

$$Q_n(s_i, a_j) = R_n + \gamma R_{n+1} + \gamma^2 R_{n+2} + \gamma^3 R_{n+3} + \cdots = \sum_{t=n}^{\infty} \gamma^t R_t$$

MYOPIC→ $\gamma = 0$

FAR-SIGHTED→ $\gamma = 1$

GREEDY POLICY AT STEP n:

$$\pi_n : a = arg \max_{a'} Q_n(a', s)$$

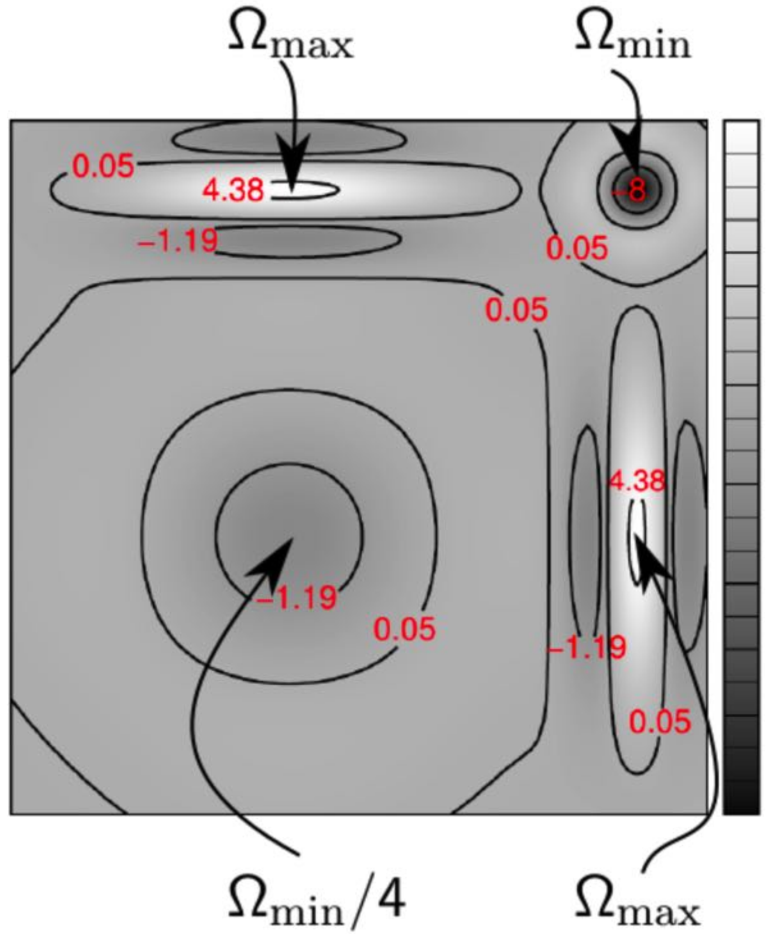
$$\begin{array}{c} s_1 \\ s_2 \\ s_3 \end{array} \begin{bmatrix} 1.2 & 0.3 & 0.1 \\ 2.2 & 4.3 & 10.1 \\ 2.0 & 8.1 & 2.0 \end{bmatrix}$$

$$\quad a_1 \; a_2 \; a_3$$

$$s_1 \xrightarrow{\pi_n} a_1$$

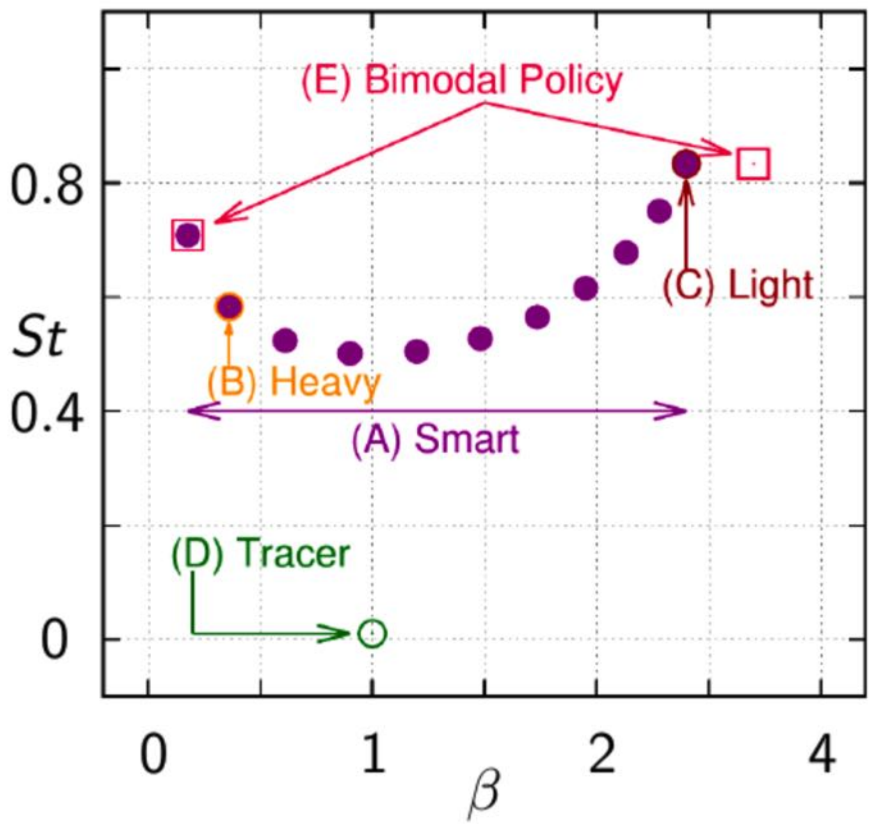
$$s_2 \rightarrow a_3$$

$$s_3 \rightarrow a_2$$



$Ns = 21$
$Na = 11$

CHANGING RADIUS b_n



$\beta \to \beta(b_n)$

$St \to St(b_n)$

Comparison with naive behaviors

Learning gain $\tilde{\Sigma}(E) = \sqrt[3]{\frac{\sum_{n=1}^{N} R_n}{N}}$