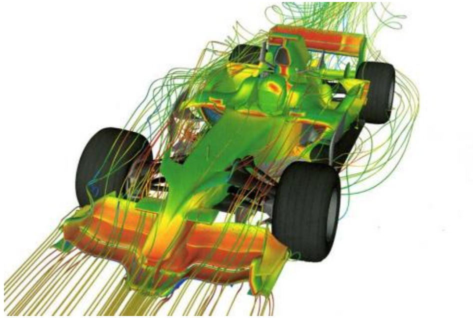
Symmetry-informed modeling and reinforcement learning control of partial differential equations

Sebastian Peitz (Safe Autonomous Systems)

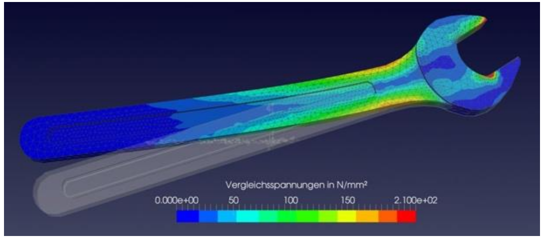
Lamarr Lab Visits – 02/18/2025

Joint work with:
H. Harder, F. Nüske, F. Philipp, M. Schaller, K. Worthmann
S. L. Brunon, J. N. Kutz, J. Stenner, O. Wallscheid

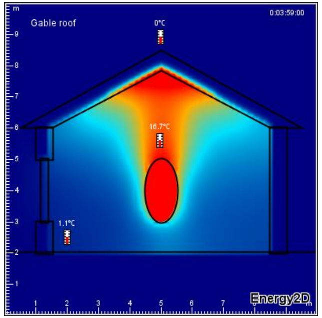
Modeling and control of complex systems



Aerodynamics



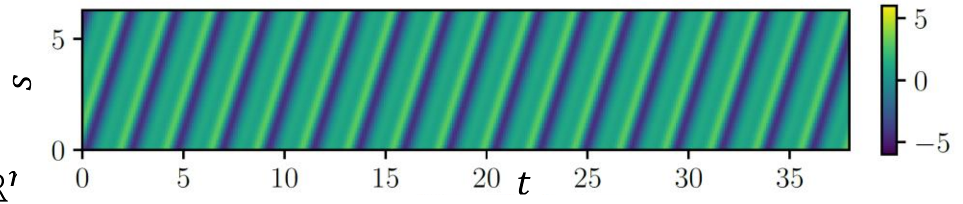
Mechanics

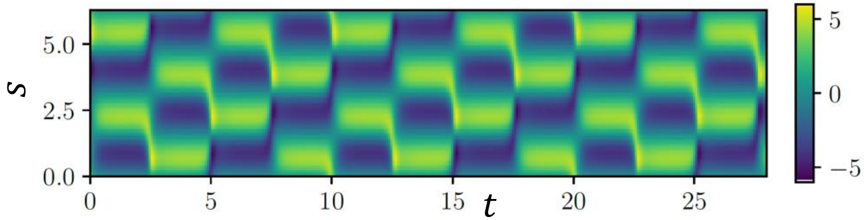


Heat flow

- Dynamics governed by partial differential equations (PDEs) depending on space x and time t
  → Expensive so simulate
  → Even more expensive to solve optimization, control or general multi-query problems
- **Aims**: 1. Learn efficient models or control laws directly from data
  2. Exploit structures (in particular symmetries) in the system dynamics

Notation and setting





- Dynamics for the PDE state $x: \Omega \times [0,T] \to \mathbb{R}^{n}$

$$\frac{\partial x}{\partial t} = \mathcal{N}(x)$$

plus appropriate boundary and initial conditions

- Example Kuramoto-Sivashinsky equation (note the translational symmetry!):

$$\frac{\partial x}{\partial t} = -\mu \left[x \frac{\partial x}{\partial s} - \frac{\partial^2 x}{\partial s^2} \right] - 4 \frac{\partial^4 x}{\partial s^4}, \qquad \Omega = (0,2\pi).$$

- Discretization in time: $\Phi(x_k) = x_k + \int_{k\Delta t}^{(k+1)\Delta t} \mathcal{N}\big(x(\cdot,t)\big)\, dt = x_{k+1}$

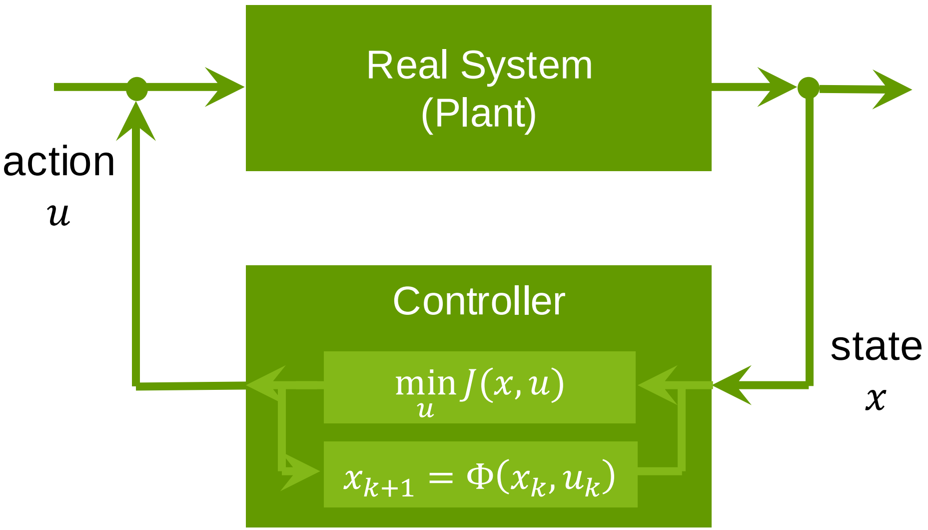
- Control: The right-hand-side has an additional input (or control / action) $u: \Omega \times [0,T] \to \mathbb{R}^m$:

$$\frac{\partial x}{\partial t} = \mathcal{N}(x,u) \qquad \text{or} \qquad x_{k+1} = \Phi(x_k, u_k)$$

→ Optimize some cost functional: $\min_u J(x,u) = \sum_{k=1}^{p} \ell(x_k, u_k) \;\; s.t. \;\; x_{k+1} = \Phi(x_k, u_k)$

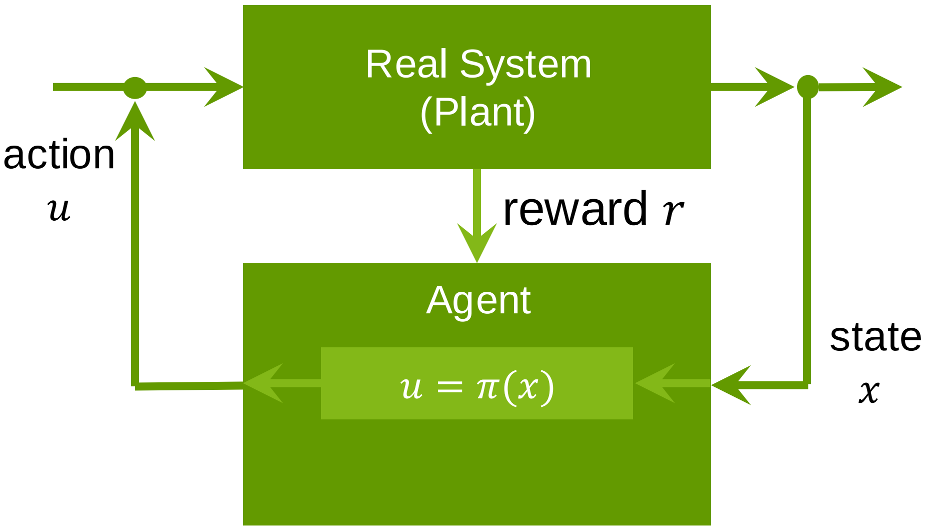
Autonomous Systems: model predictive control vs. reinforcement learning

MPC



- Learn a model Φ from data (offline)
- Solve control problem online → action u

RL



- Learn a policy π which maximizes the reward (Bellmann principle)
- Evaluate π online → action u

<u>real time capability</u> ← **Challenges** → <u>training effort</u>

Surrogate model from data: Koopman operator [Koopman 1931, Mezic 2005, Rowley et al. 2009]

Bernhard Koopman and John von Neumann proposed already in the 1930s to transfer operator theoretic concepts from quantum mechanics to classical mechanics:

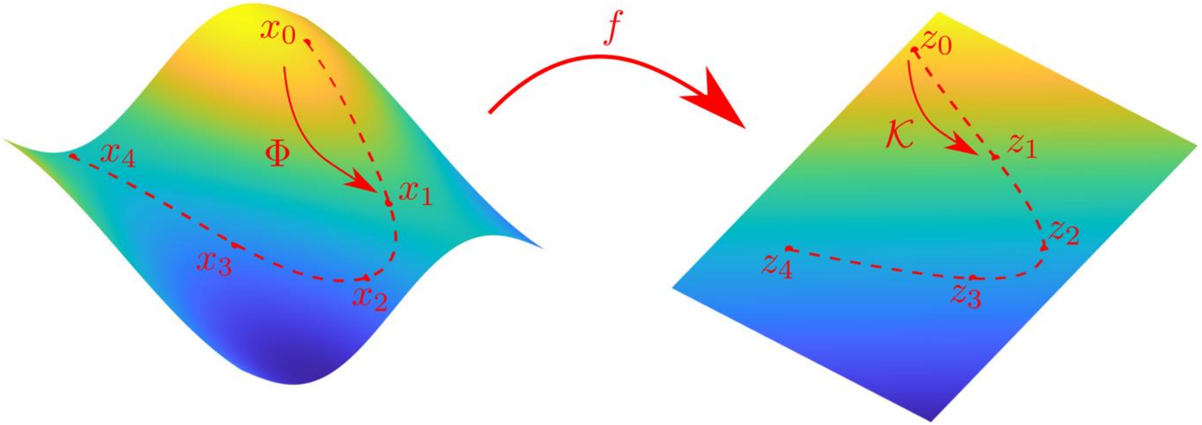
- We do not directly study the state x of a system, but instead an

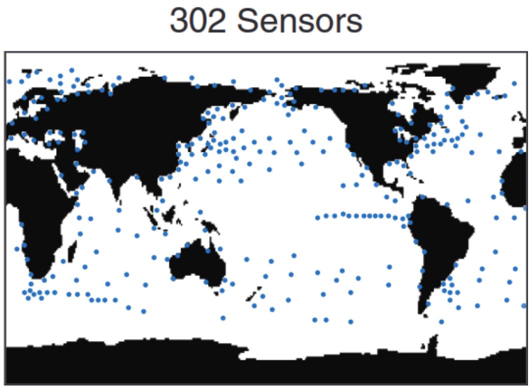
  observable function $f \in \mathcal{F}$ with $z = f(x)$

- The Koopman operator $\mathcal{K}: \mathcal{F} \to \mathcal{F}$ then acts linearly on these observables:

$$(\mathcal{K}f)(x) = f(\Phi(x))$$

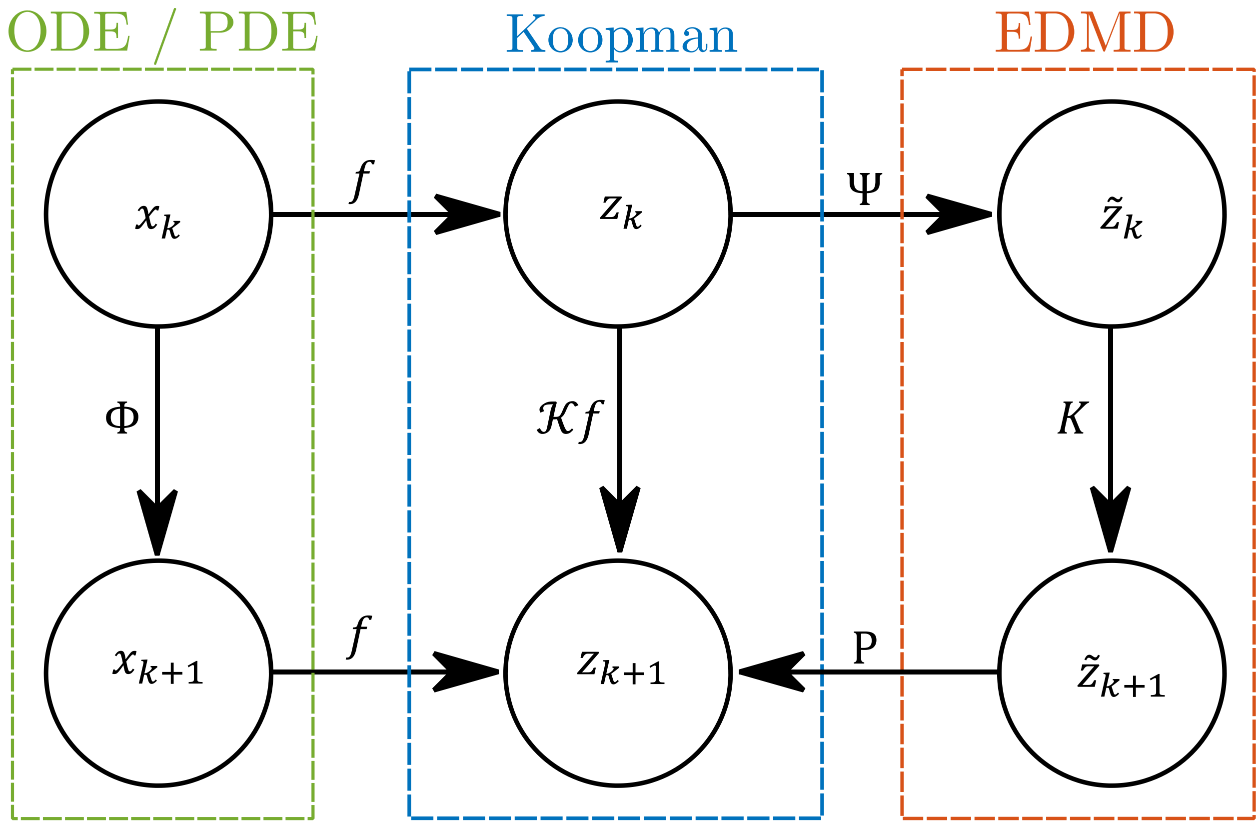
- It is a linear yet infinite-dimensional operator, even if the original system Φ is nonlinear!



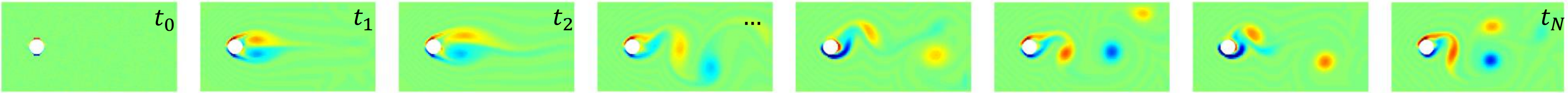


Surrogate model from data: Koopman operator

$$(\mathcal{K}f)(0x) = f(\Phi(x))$$



Dynamic Mode Decomposition (DMD) [Schmid 2010]



- Learn from time series data $x_0, \dots, x_N$, mit $x_{k+1} = \Phi(x_k)$

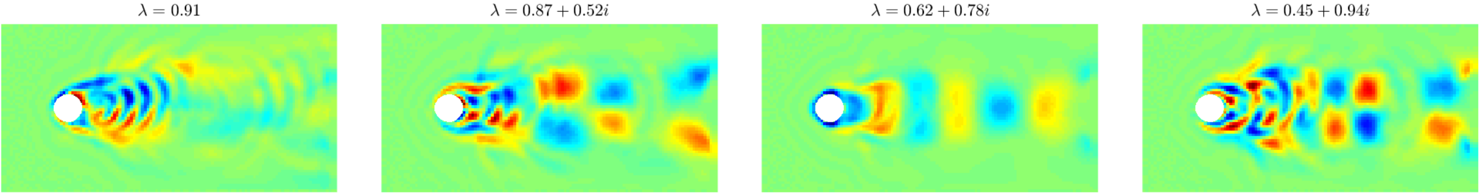
$$X = [x_0 \quad \dots \quad x_{N-1}], \qquad \hat{X} = [x_1 \quad \dots \quad x_N]$$

- Suppose there exists a linear Operator K such that $\hat{X} = KX$.

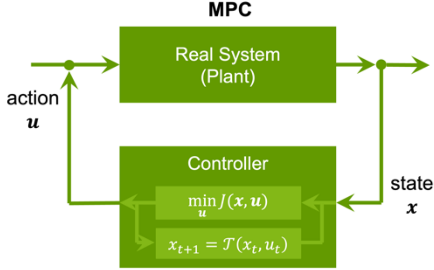
Then $\mathrm{K} = \hat{X}X^\dagger$ minimizes $\left\|\hat{X} - KX\right\|_F$ $\Rightarrow$ Simple linear regression

→ This matrix K is a finite-dimensional approximation of the Koopman operator!

- the eigenvectors of $\boldsymbol{K}$ approximate eigenfunctions of the Koopman operator
- the complex eigenvalues indicate frequency & growth/decay



Surrogate model from data: Koopman operator



- The assumption of linearity breaks down quickly!
- But: if we do not study the state x, but sensor data $z = f(x)$ and define a basis Ψ for f,

$$f(x) = \sum_{i=1}^{q} c_i \psi_i(x) = c^\top \Psi(x),$$

Then our regression problem becomes anot

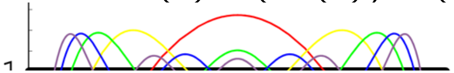
$$\min_{K \in \mathbb{R}^{q \times q}} \sum_{k=1}^{N-1} \|\Psi(x_{k+1}) - K\Psi($$

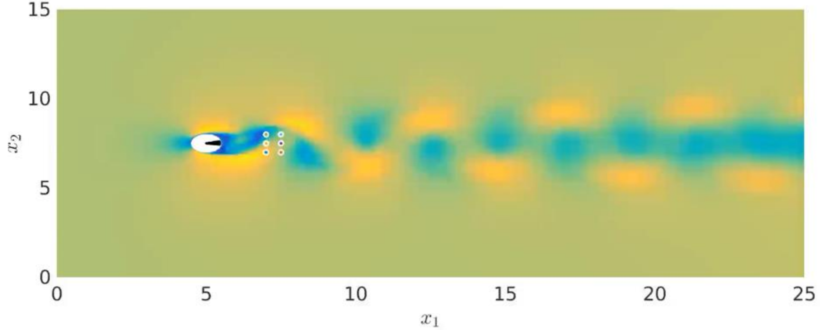
→ This matrix is a much better approxima

→ Convergence results

→ Error bounds

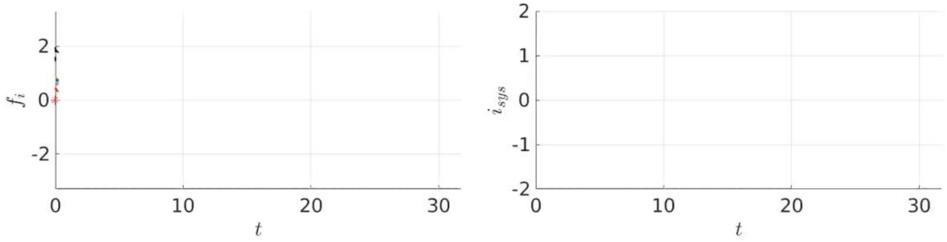
Example: Fourier series expansion of f:
$\Psi(x) = (\sin(x), \cos(x), \sin(xs), \ldots)$



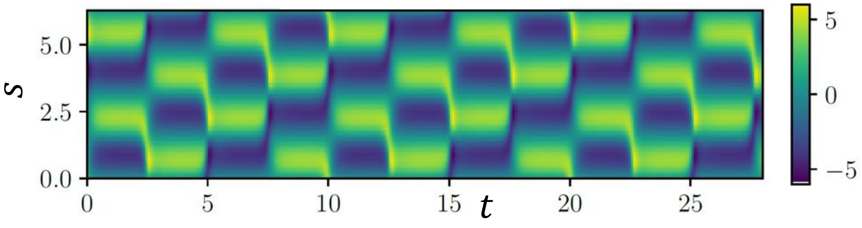


[Williams et al. 2015], [Korda & Mezic 2018], [Klus, Nüske, P

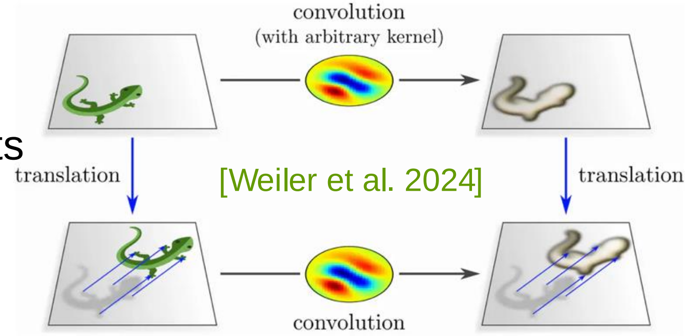
- Analogy to Support Vector Machines (SVMs) for classification:
- Many applications
- Extensions for control problems



Group Convolutional EDMD [Harder et al. 2024]



- Assume that we have symmetries in our system
- Examples: Equivariance to
  - Shift, Flips, Rotation (discrete, continuous)
- Formally defined by a symmetry group $\mathcal{G}$ and its group actions $g \in \mathcal{G}$. Examples



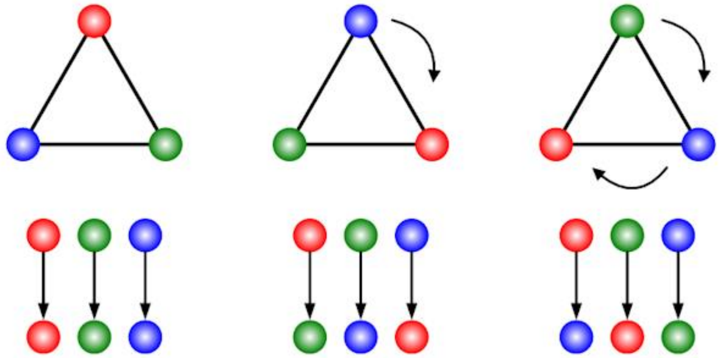
[Weiler et al. 2024]

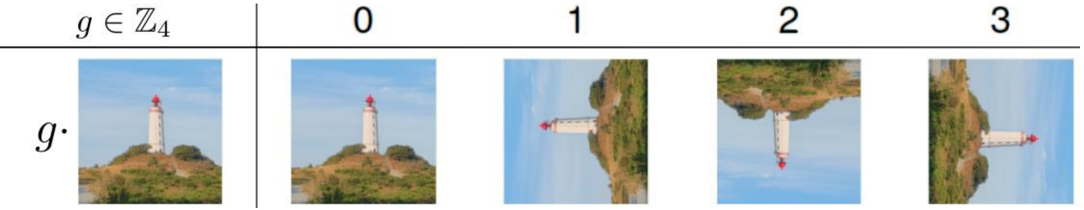
  - $\mathcal{G} = (\mathbb{R}, +)$: the translation group of continuous shifts

$$\mathrm{g} \cdot s = s + g$$
$$(g \cdot x)(s) = x(g^{-1} \cdot s) = x(s - g)$$

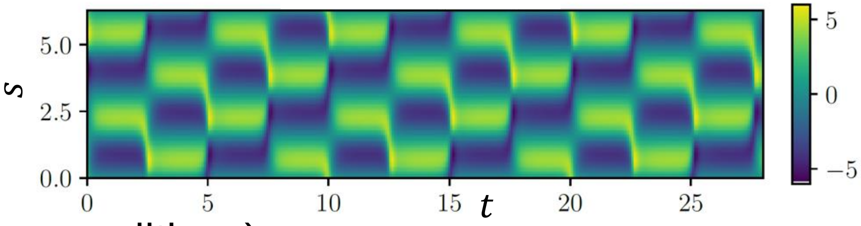
  - The cyclic group $\mathbb{Z}_n$ of integer shifts:

$$g \cdot s = (\mathrm{s} + \mathrm{g}) \bmod \mathrm{n}$$



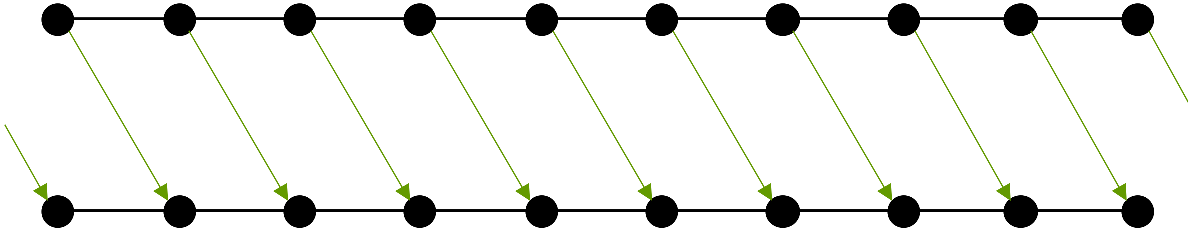


Group Convolutional EDMD [Harder et al. 2024]

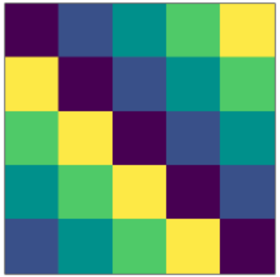


- The cyclic group can also encode shifts…
- … on a discretized domain (with periodic boundary conditions)

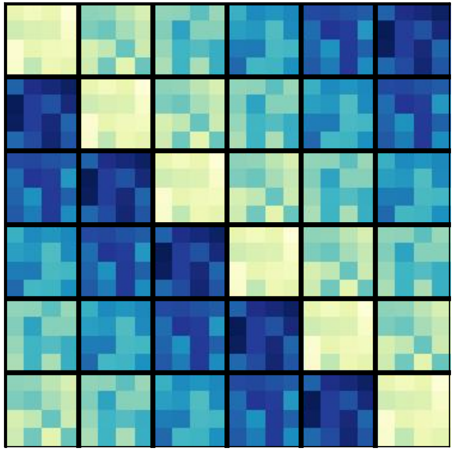
$$g \cdot s_i = s_{(g \cdot i)} = s_{i+1}$$



- For DMD, this shift equivariance implies that the DMD matrix is circulant.
- Instead of learning a matrix, we can learn a convolution kernel!

 $x =$  $* \, x$

- In Extended DMD, this kernel simply has multiple channels

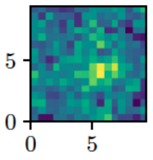


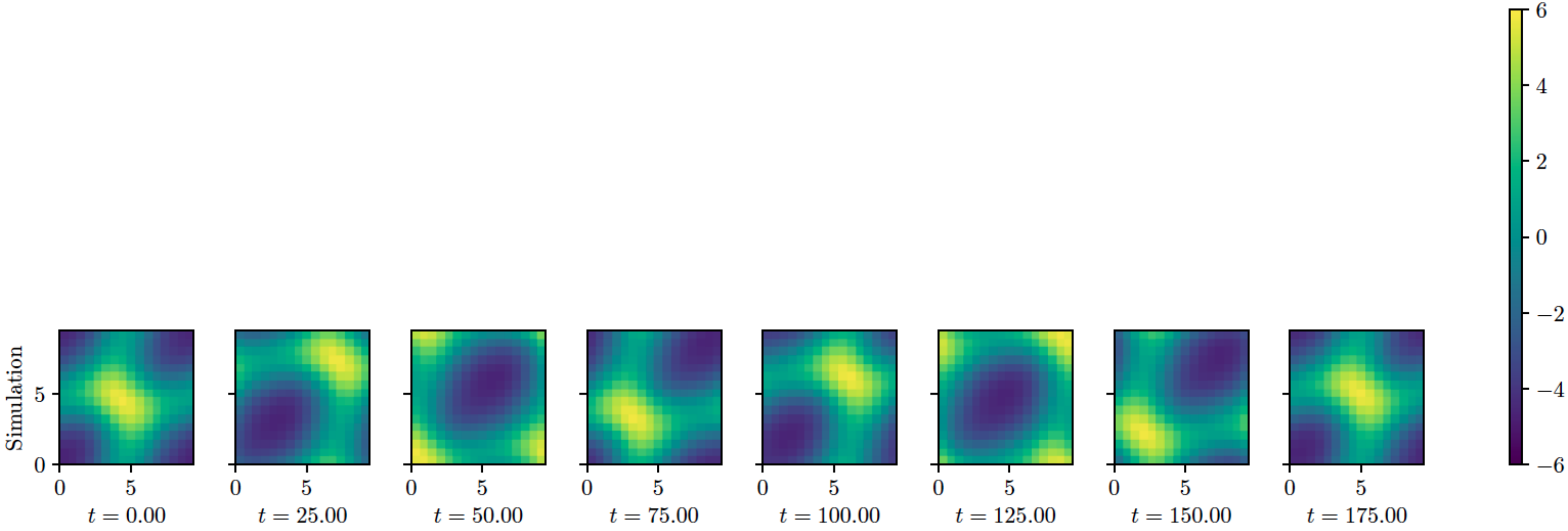
Group Convolutional EDMD [Harder et al. 2024]

- Example: the Kuramoto-Sivashinsky equation in 2D

$$\frac{\partial x}{\partial t} = \mathcal{N}(x) = -\Delta x - \Delta^2 x - \frac{1}{2}\|\nabla x\|^2$$

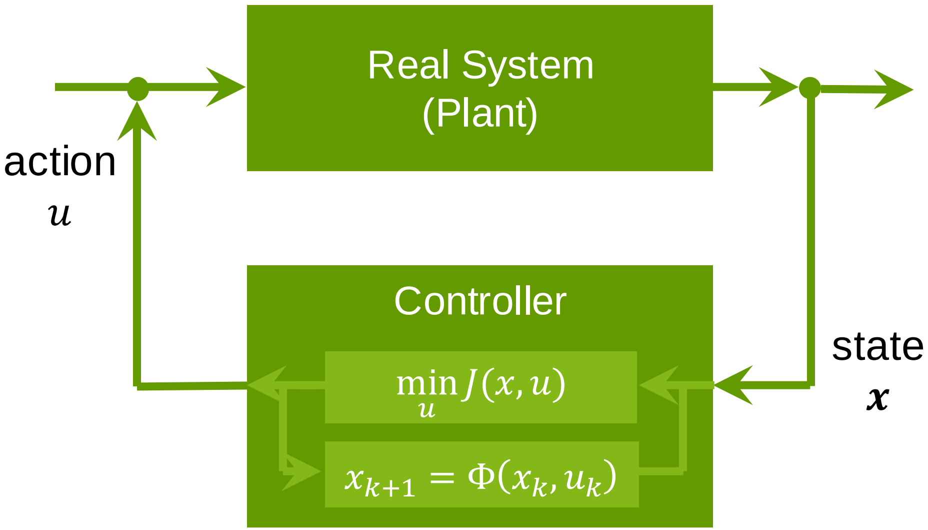
- Equivariance w.r.t. shifts in both directions (the system is also equivariant w.r.t. continuous rotations and flips (i.e., $E(2)$), but we only consider $\mathbb{Z}_n$ here)





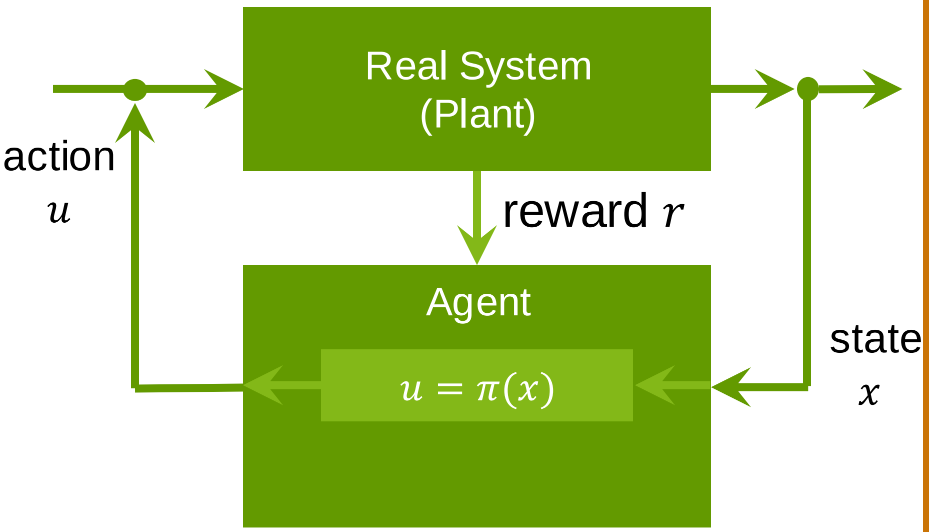
Autonomous Systems: model predictive control vs. reinforcement learning

MPC



- Learn a model Φ from data (offline)
- Solve control problem online → action u

RL

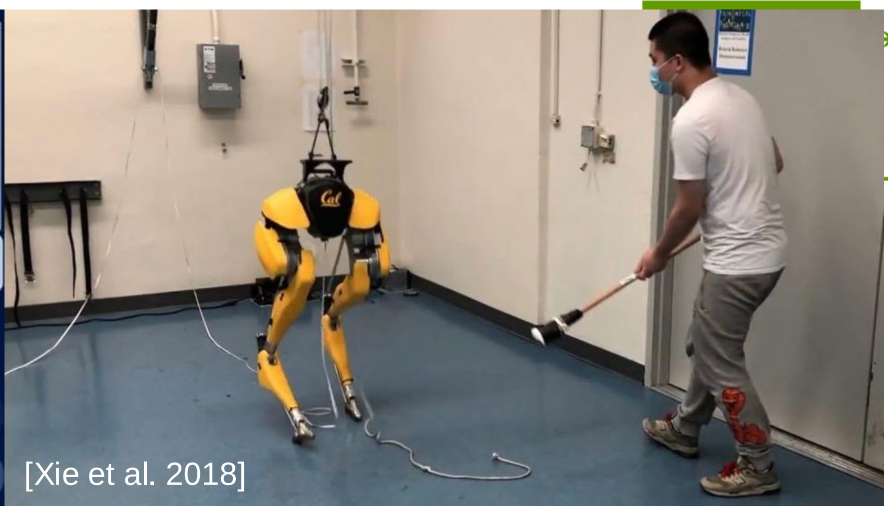


- Learn a policy π which maximizes the reward (Bellmann principle)
- Evaluate π online → action u

<u>real time capability</u> ← **Challenges** → <u>training effort</u>

Re

[Silver et al. 2016]


[Xie et al. 2018]

state
x

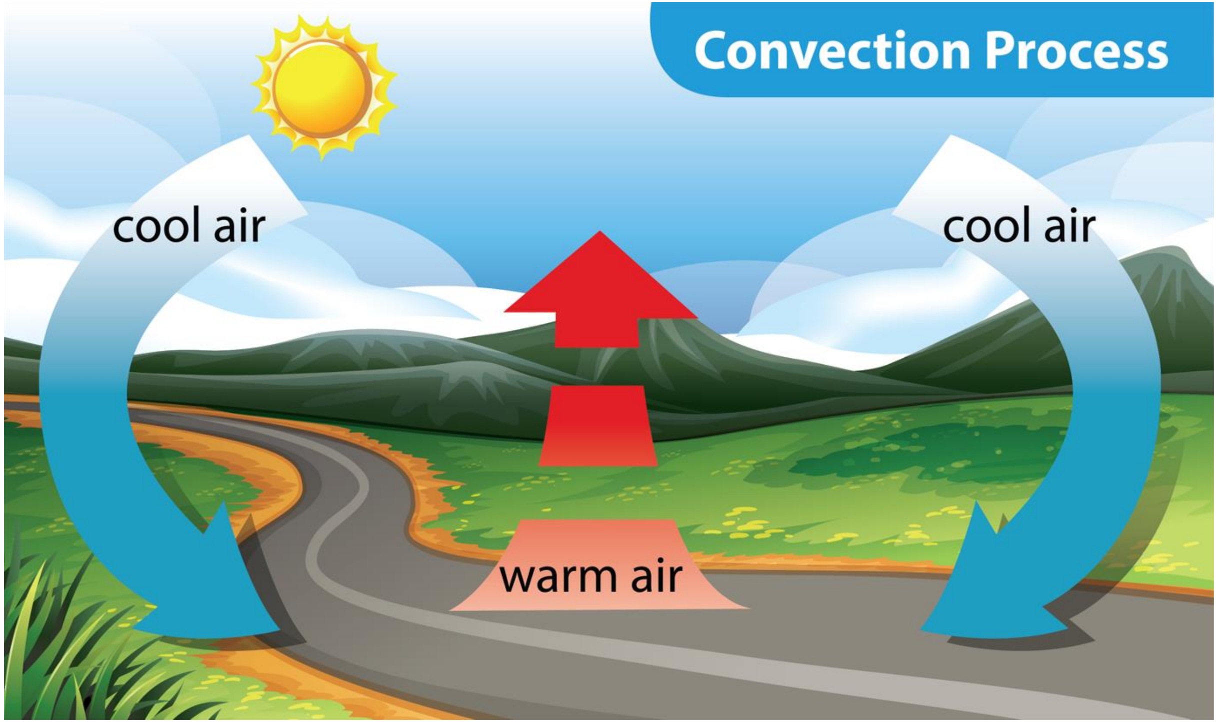
- The reward function $\mathcal{R}\colon \mathcal{X} \times \mathcal{U} \to P(\mathbb{R}) \rightarrow r_\tau$ determines whether the taken action was favorable or bad

- Goal in RL: Find a policy $\pi\colon \mathcal{X} \to P(\mathcal{U})$ which maximizes the expected value of the discounted sum of future rewards (the so-called value) :

$$\pi^* = \arg\max_{\pi} V^\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{\tau+k} \middle| s_\tau = s \right], \qquad \text{where } \gamma \in [0,1]$$

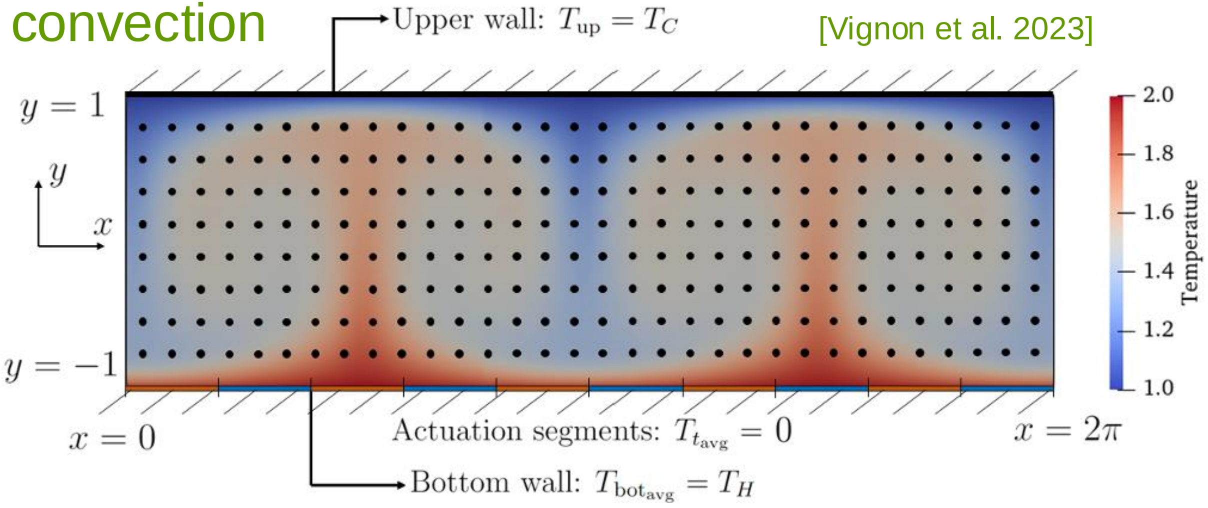
- Challenge: Training can be very expensive!

Example: Rayleigh Bénard convection



Example: Rayleigh Bénard convection

[Vignon et al. 2023]

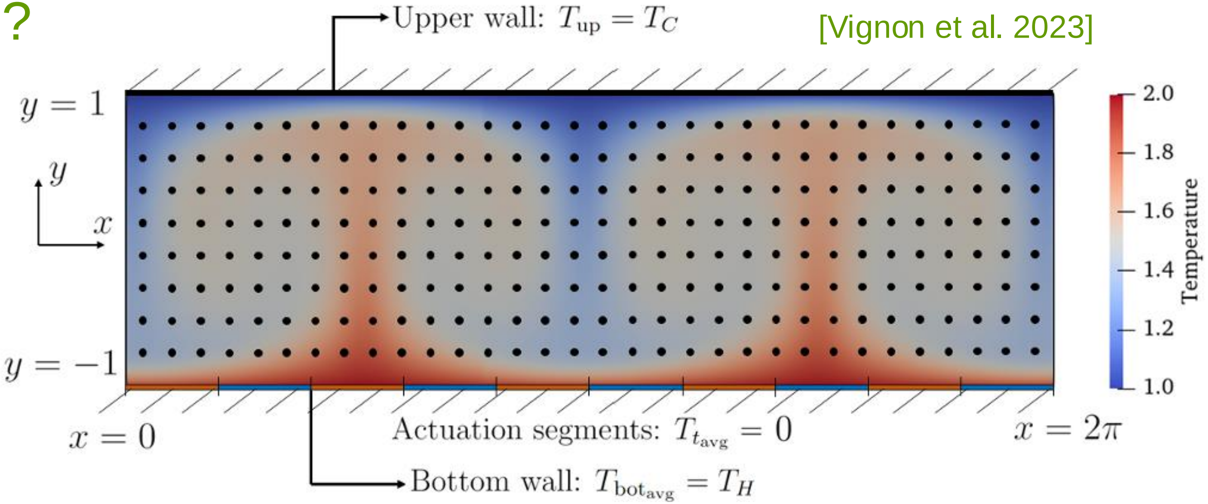


Setup: Compressible flow between two flat plates
→ heated at the bottom
→ cooling at the top

Goal: reduction of the convective heat transport by modification of the bottom temperature
→ A single agent has many states and 10 actions → expensive

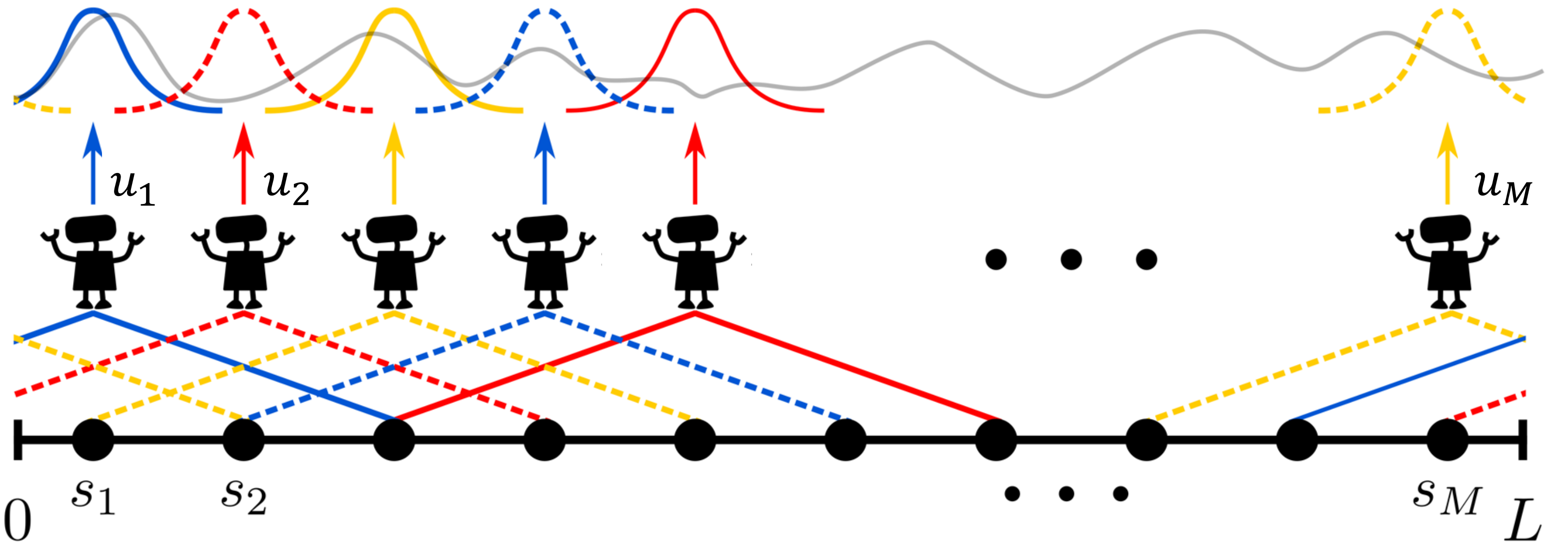
Can we exploit symmetries?

[Vignon et al. 2023]



1. The system is equivariant under horizontal shifts!
   → This is true for any right-hand side $\mathcal{N}(x)$ where the position s does not explicitly show
2. Information (mass, energy, …) is transported with finite velocity
   → For a local decision, it is sufficient to consider only information close by

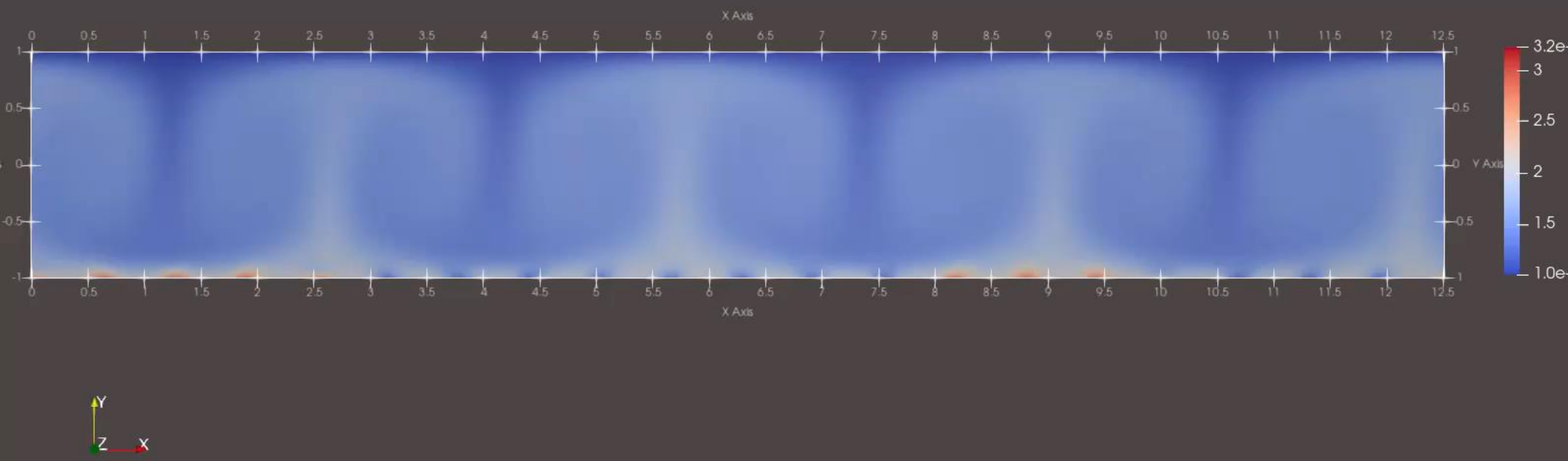
Convolutional reinforcement learning [Peitz, Stenner, Chidananda, Wallscheid, Brunton, Taira 2024]



Advantages:

1. Dimensionality reduction: few inputs, one output
2. Parameter sharing: All agents are identical
3. Transferability to other (i.e., larger) domains!

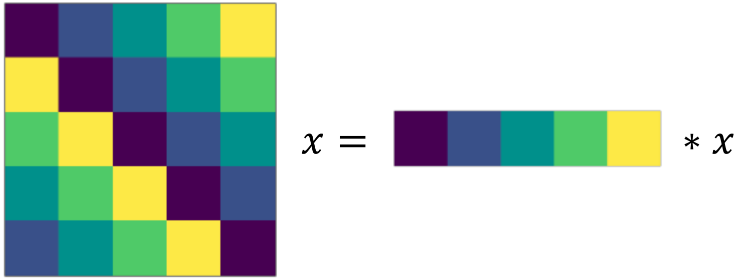
Example: Rayleigh Bénard convection

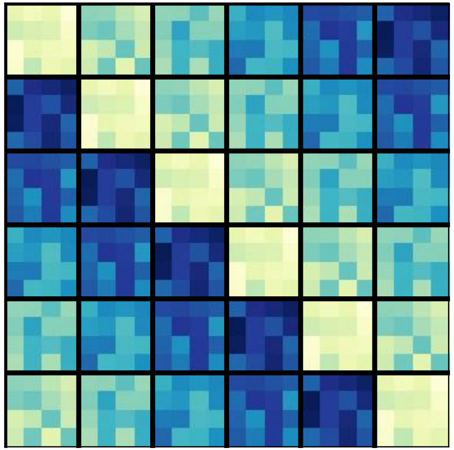


Conclusion

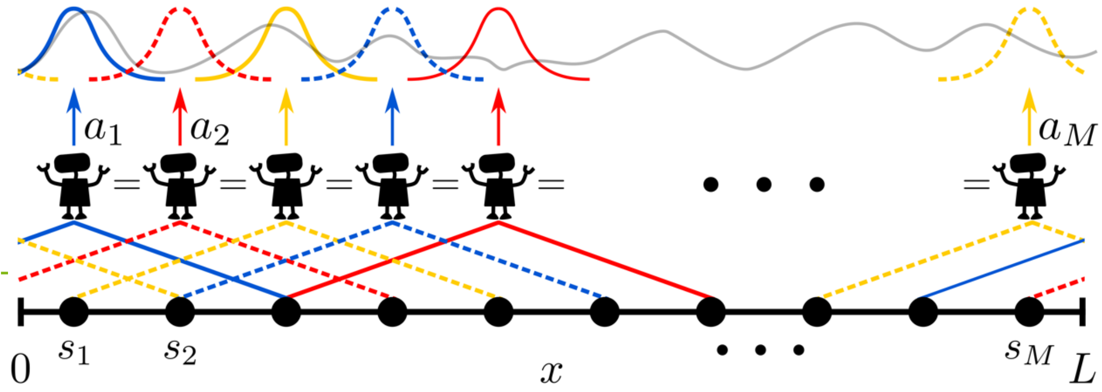
Symmetries can help us to reduce the effort in modeling and control

- Group convolutional DMD





- Reinforcement learning



Thanks for your attention!