Control of tokamak plasmas through Deep Reinforcement Learning: application to magnetic control on TCV

J. Degrave, F. Felici et al., Nature 602, 414–419 (2022)

Article

Magnetic control of tokamak plasmas through deep reinforcement learning

https://doi.org/10.1038/s41586-021-04301-9
Received: 14 July 2021
Accepted: 1 December 2021
Published online: 16 February 2022
Open access

Jonas Degrave[1,3], Federico Felici[2,3]✉, Jonas Buchli[1,3]✉, Michael Neunert[1,3], Brendan Tracey[1,3]✉, Francesco Carpanese[1,2,3], Timo Ewalds[1,3], Roland Hafner[1,3], Abbas Abdolmaleki[1], Diego de las Casas[1], Craig Donner[1], Leslie Fritz[1], Cristian Galperti[2], Andrea Huber[1], James Keeling[1], Maria Tsimpoukelli[1], Jackie Kay[1], Antoine Merle[2], Jean-Marc Moret[2], Seb Noury[1], Federico Pesamosca[2], David Pfau[1], Olivier Sauter[2], Cristian Sommariva[2], Stefano Coda[2], Basil Duval[2], Ambrogio Fasoli[2], Pushmeet Kohli[1], Koray Kavukcuoglu[1], Demis Hassabis[1] & Martin Riedmiller[1,3]

Federico Felici

Swiss Plasma Center, EPFL Lausanne

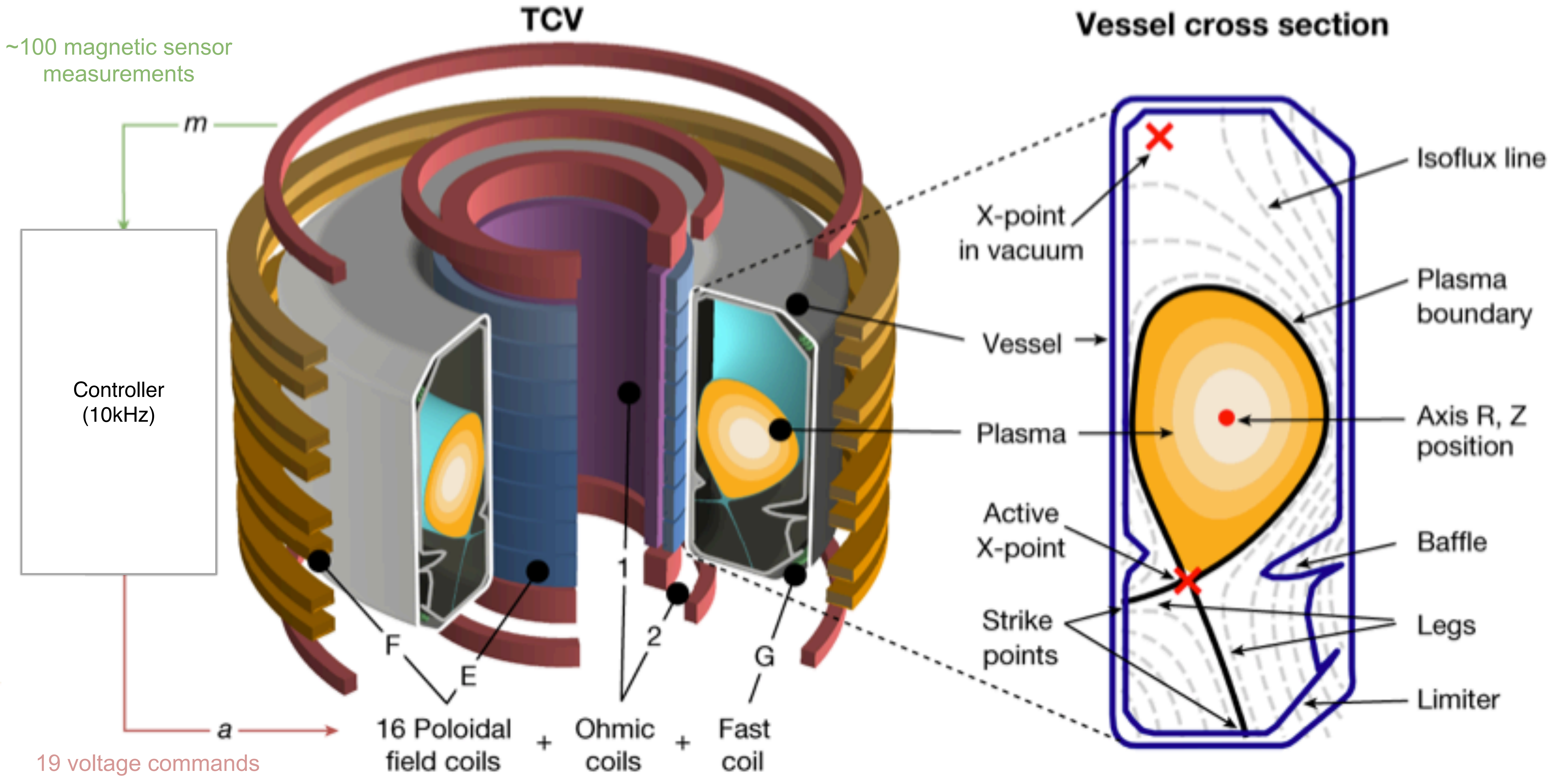
on behalf of the DeepMind-EPFL team

Jonas Degrave, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de las Casas, Craig Donner, Leslie Fritz, Cristian Galperti, Andrea Huber, James Keeling, Maria Tsimpoukelli, Jackie Kay, Antoine Merle, Jean-Marc Moret, Seb Noury, Federico Pesamosca, David Pfau, Olivier Sauter, Cristian Sommariva, Stefano Coda, Basil Duval, Ambrogio Fasoli, Pushmeet Kohli, Koray Kavukcuoglu, Demis Hassabis & Martin Riedmiller

3rd Fusion HPC workshop (Virtual)
15th-16th December 2022

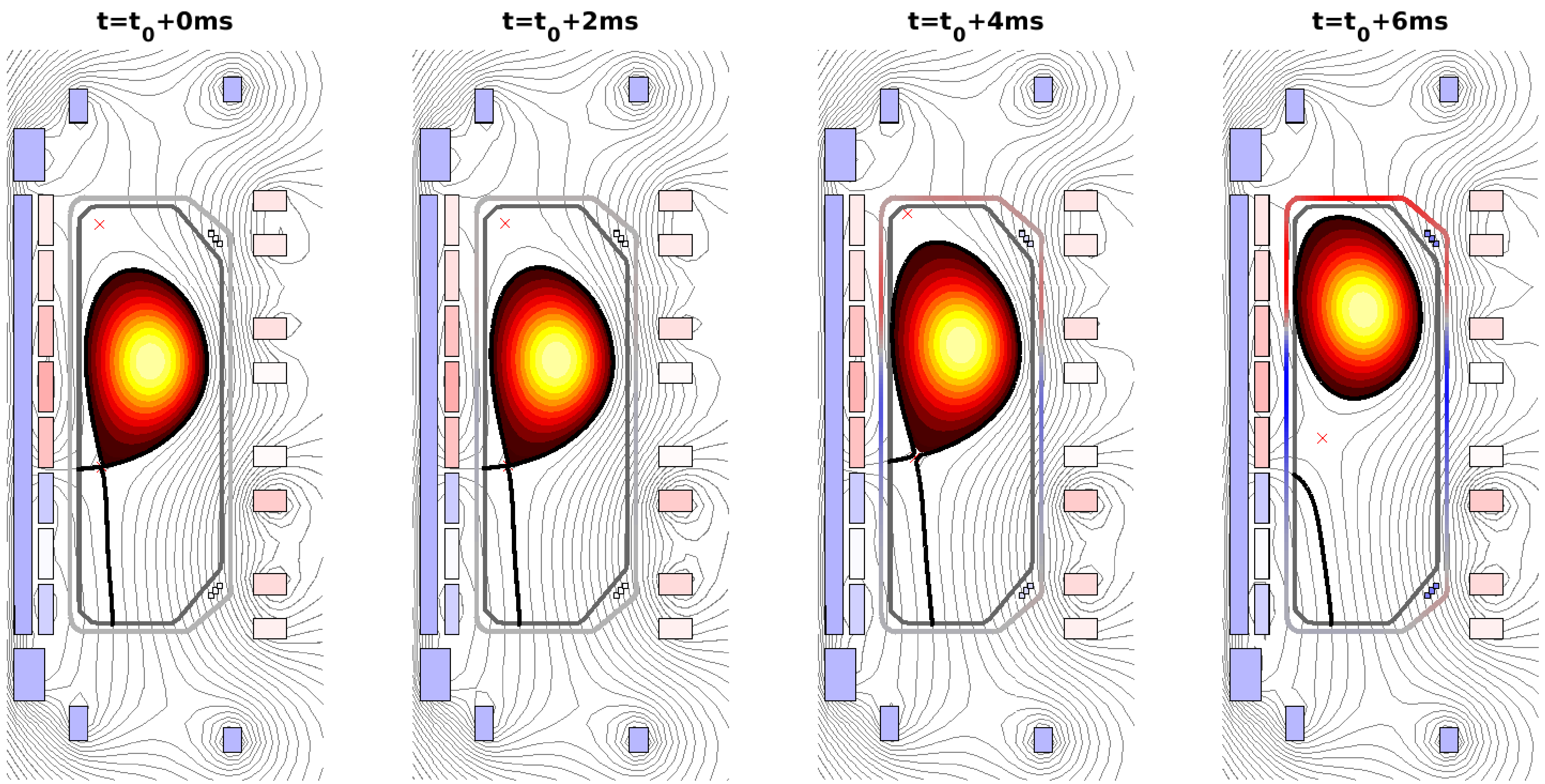
SWISS
PLASMA
CENTER

The axisymmetric equilibrium control problem



The axisymmetric equilibrium control problem

- **Need to control:**
  - **Total plasma current I_p (maintained by induced voltage caused by transformer effect)**
  - **Radial position R (by vertical magnetic fields)**
  - **Vertical position Z (by radial magnetic fields - *unstable for elongated plasmas*)**
  - **Plasma shape: <u>last closed flux surface</u> distribution**

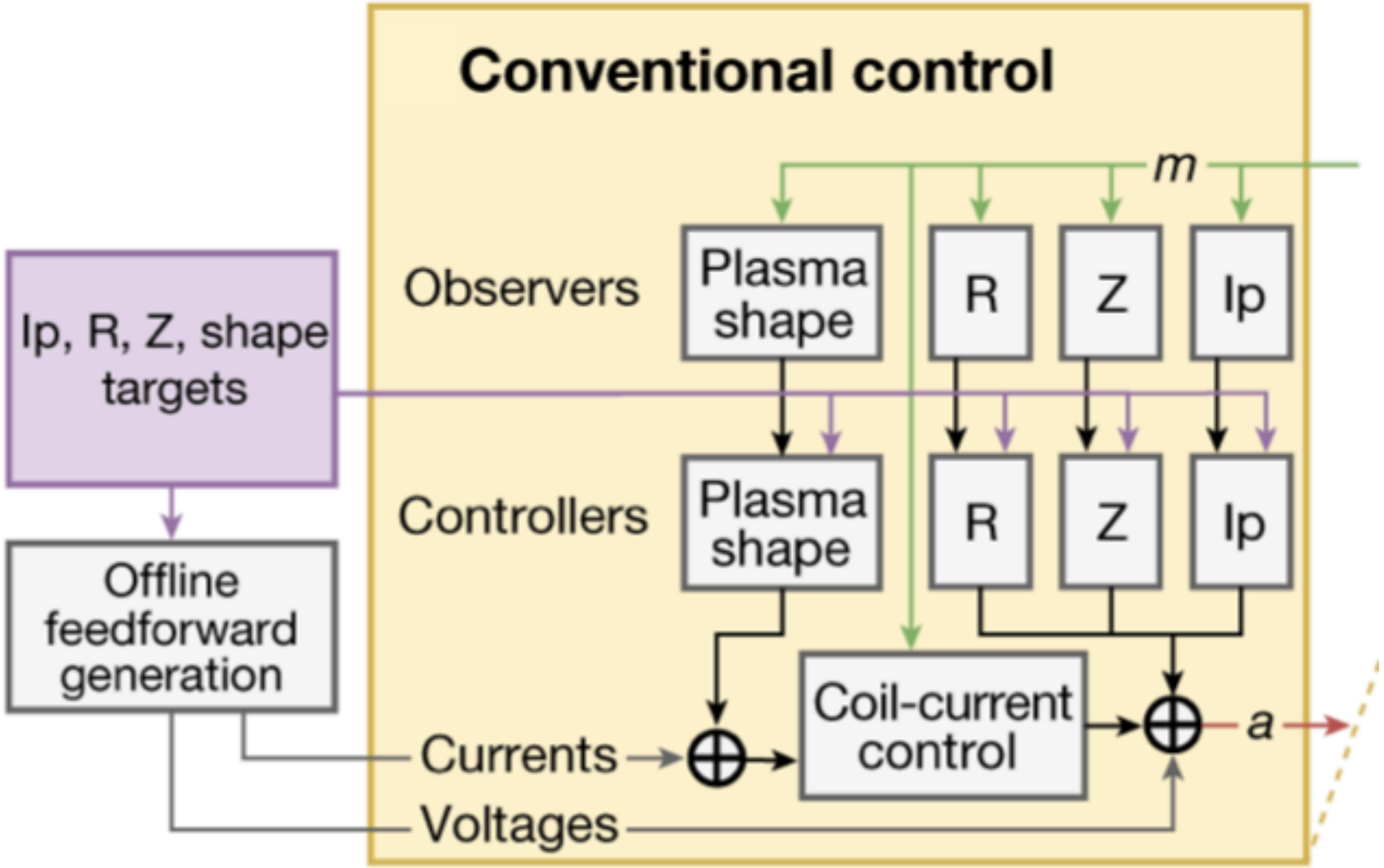


Traditional solutions

- **Pre-shot**
  - **Pre-compute feedforward coil currents & voltages**
  - **Design feedback controllers for stabilization & tr**

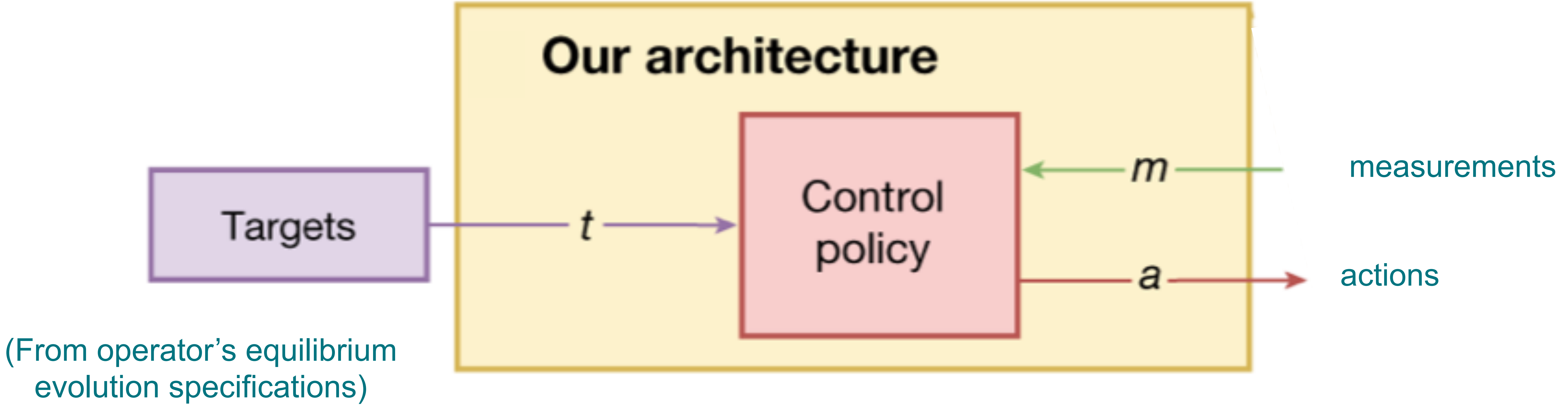
- **During shot:**
  - **Real-time position estimators**
  - **Real-time equilibrium reconstruction**
  - **Separate real-time controllers per ‘channel’**

- **Today mostly done using traditional control engineering**
  - **‘Model-based design’ + sometimes hand tuning of gains**



See e.g. De Tommasi, G. Plasma Magnetic Control in Tokamak Devices. *J Fusion Energ* **38,** 406–436 (2019)

A single controller?



- **No separate calculation of controlled variables / equilibrium reconstruction**
- **No separate design of various control loops**

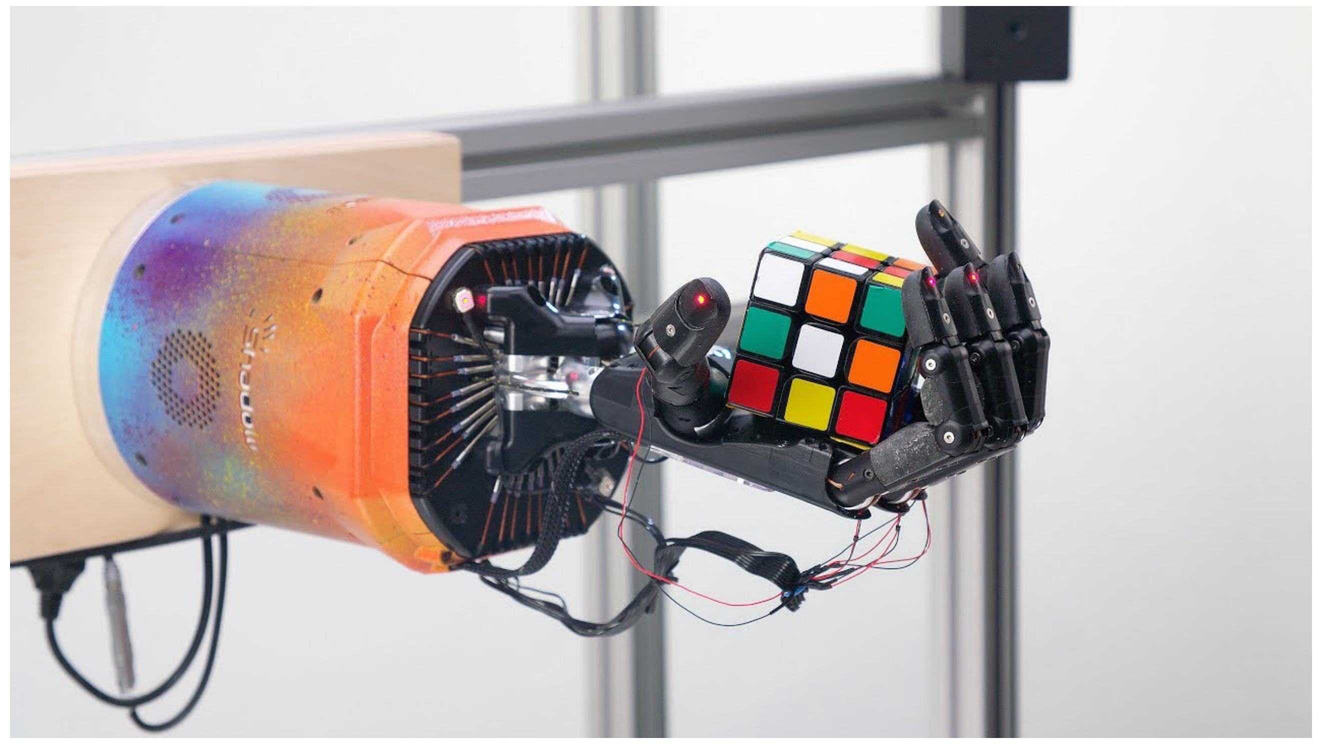
- **Use Reinforcement Learning**

Reinforcement learning

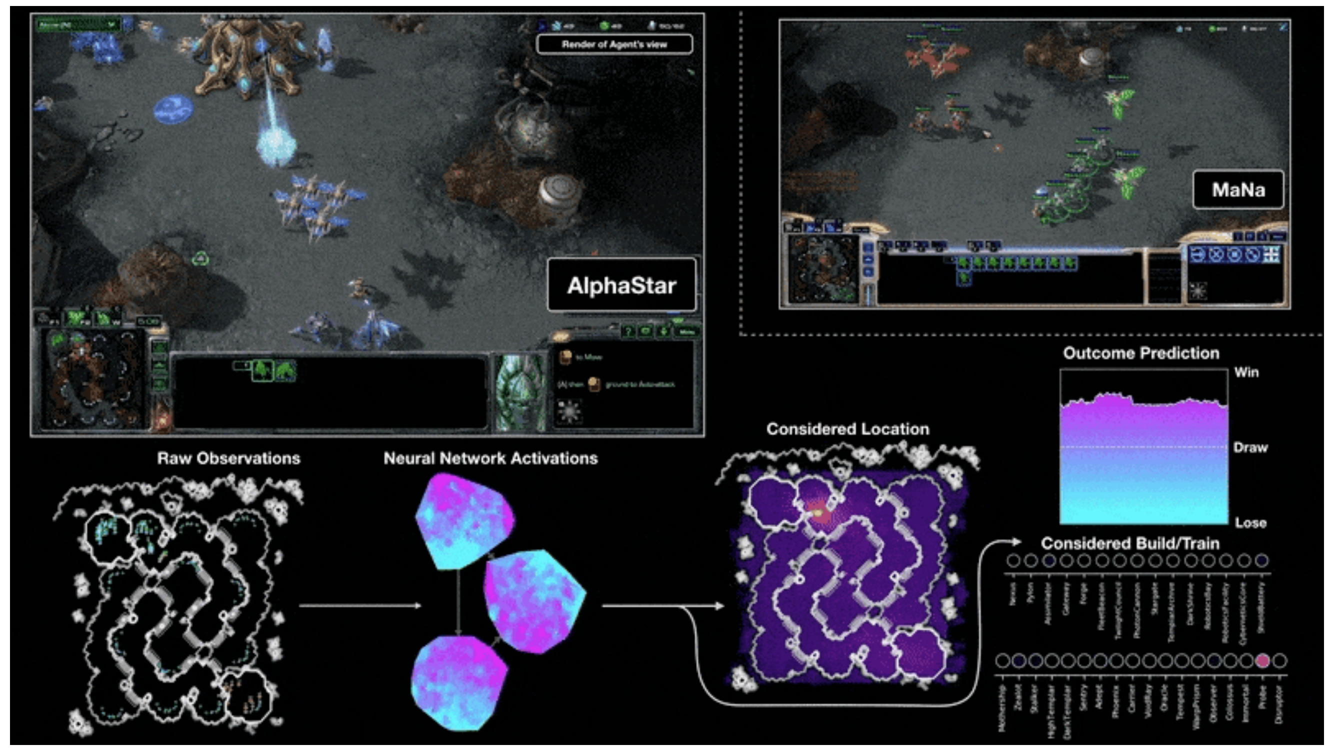
- **How do we (humans) learn to solve problems?**
  - Trial and error interaction with the environment
- **Reinforcement learning (RL) is a general framework to express how this process is performed.**
- **There are two important aspects to the paradigm**
  - It allows us to specify the goal (Reward function)
  - It can deal with long-term dependencies (dynamical systems)

https://www.deepmind.com/blog/alphazero-shedding-new-light-on-chess-shogi-and-go

StarCraft II



https://openai.com/blog/solving-rubiks-cube/

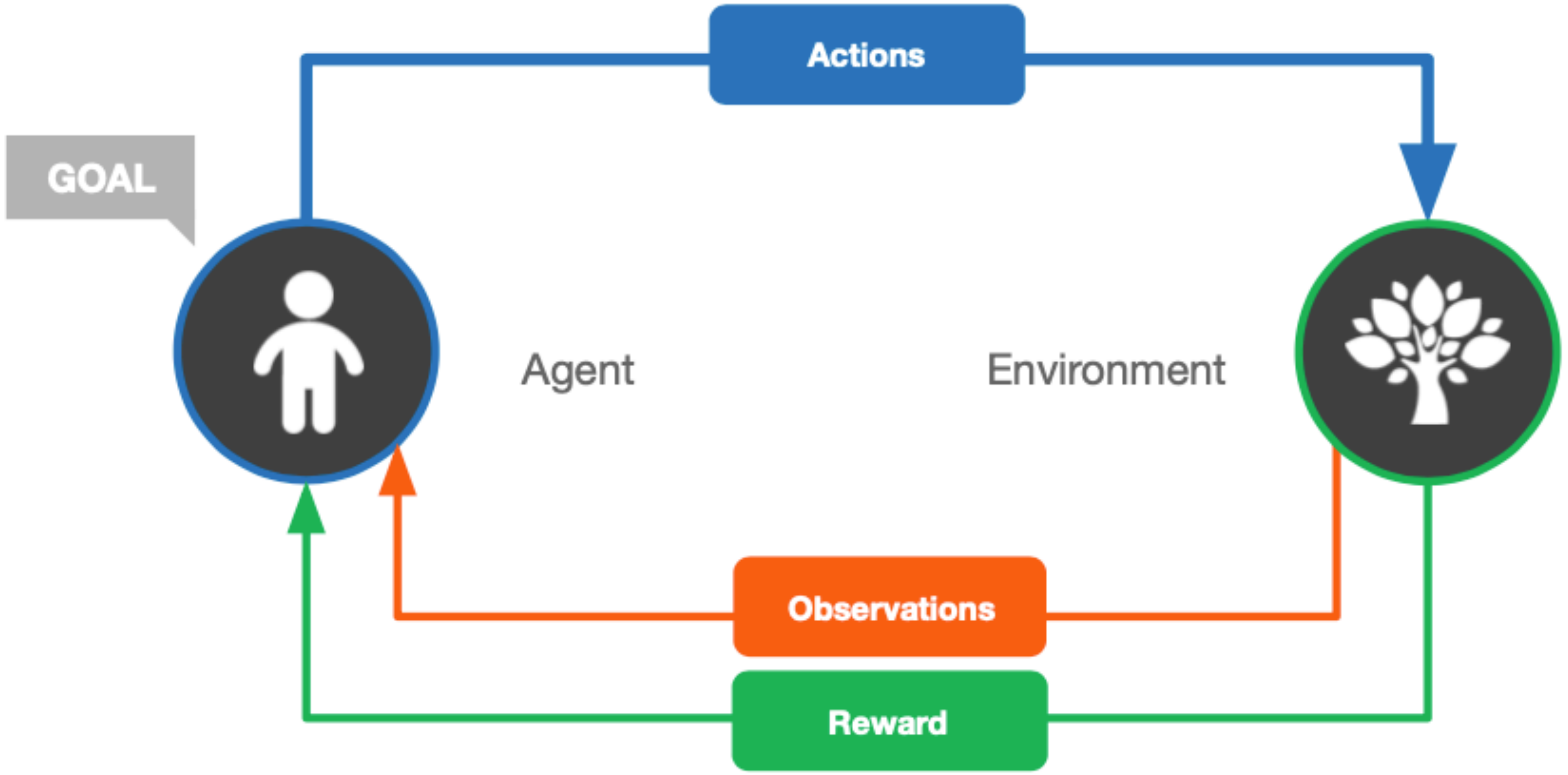


https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii

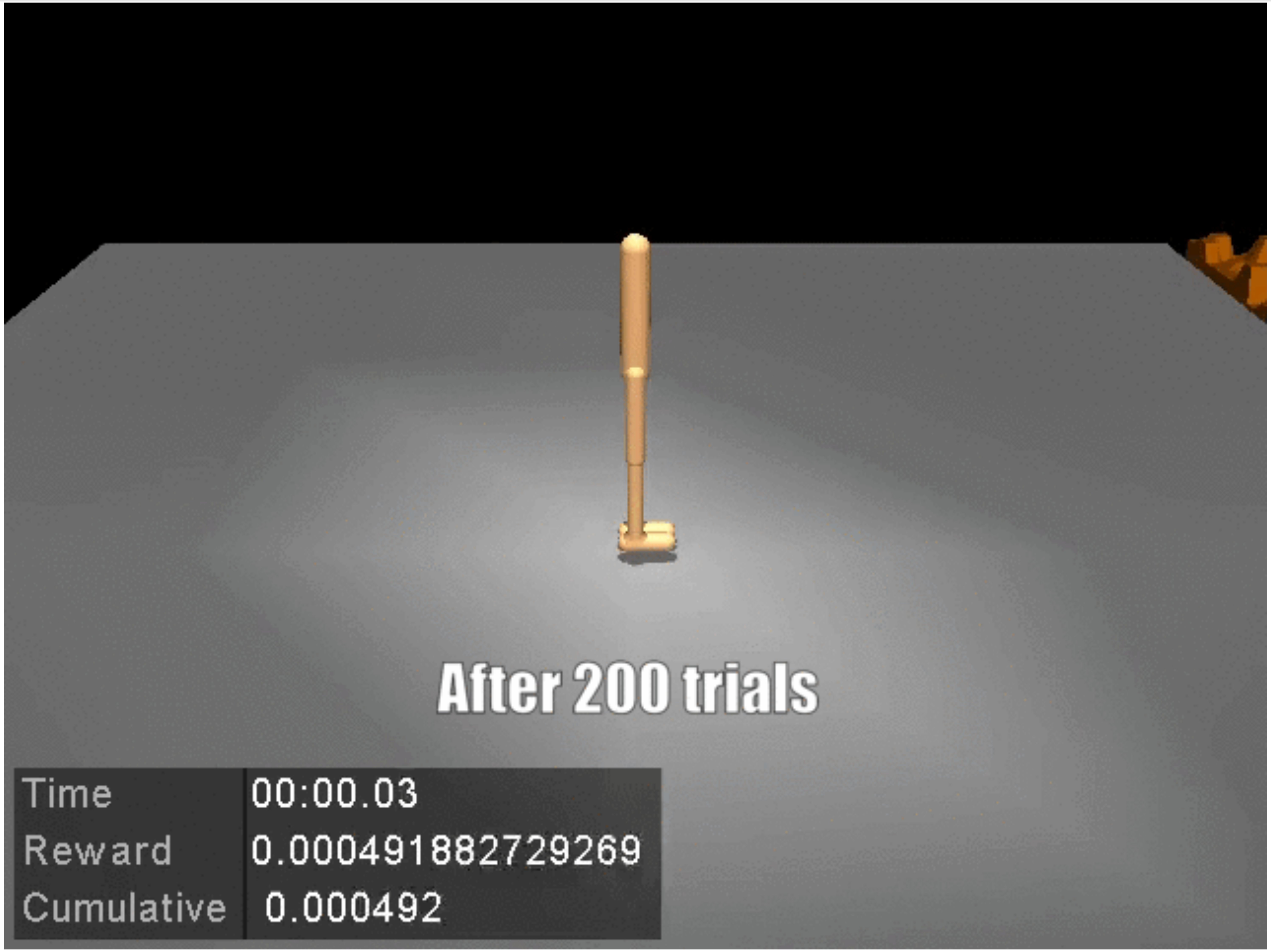
Reinforcement learning versus other learning

- **Supervised learning: Learn to classify data based on labeled examples**
- **Unsupervised learning: Learn to separate data based on similarities or differences**
- **Reinforcement learning: Learn by <u>trial-and-error</u> how to act on an environment to achieve high reward**
  - **<u>Exploration</u> to gather experience + <u>learning from</u> the experience**

[Figure and RL slide material from hereon: courtesy A. Abdolmaleki]



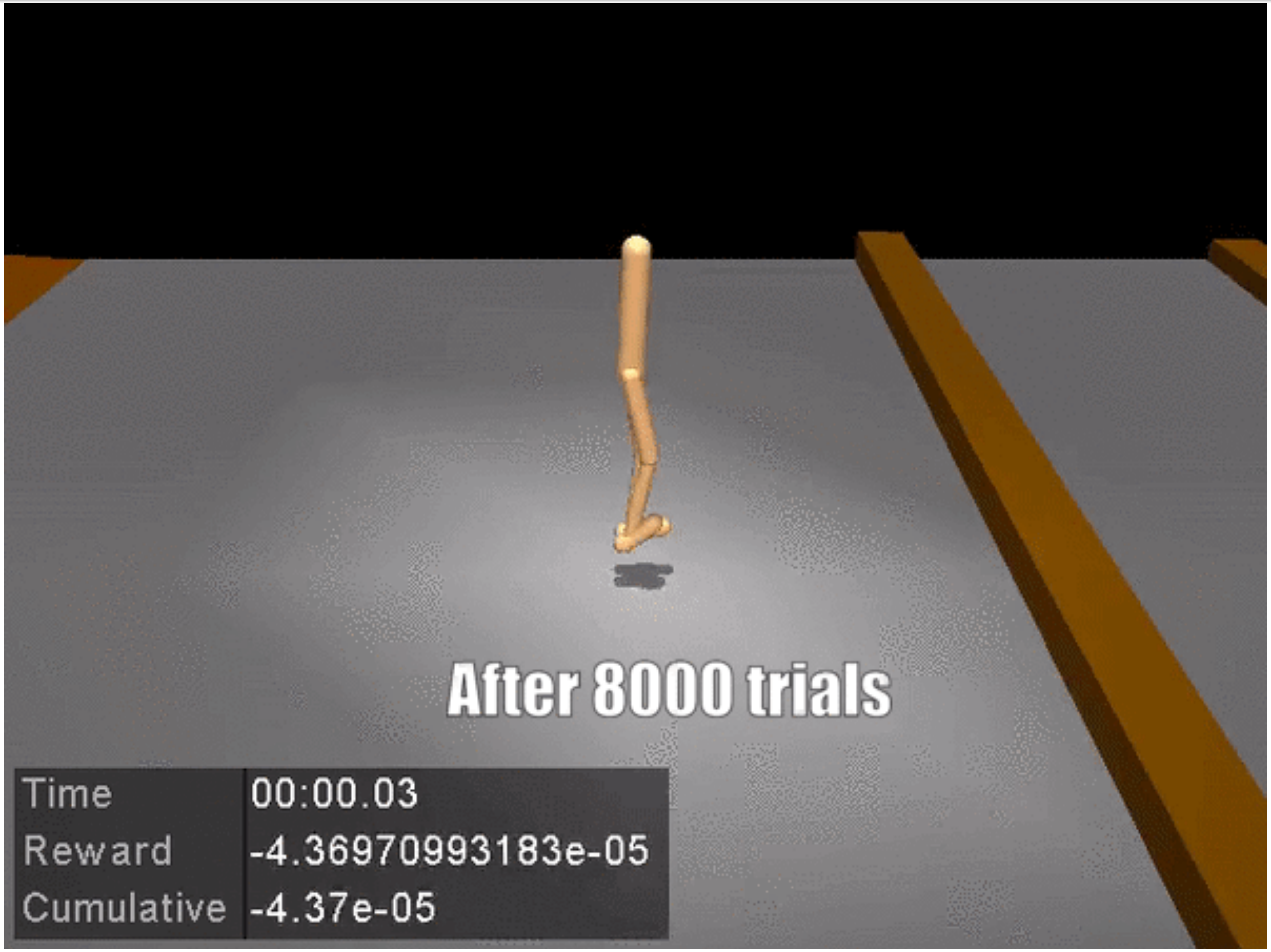
See also: [Sutton and Barto, *Reinforcement Learning, an Introduction*. MIT Press]



[Credits A. Abdolmaleki, DeepMind]


After 2000 trials
Time 00:00.03
Reward -0.000388528895386
Cumulative -0.000389


After 6000 trials
Time
00:00.03
Reward
-0.00096222858493
Cumulative
-0.000962


After 8000 trials
Time 00:00.03
Reward -4.36970993183e-05
Cumulative -4.37e-05

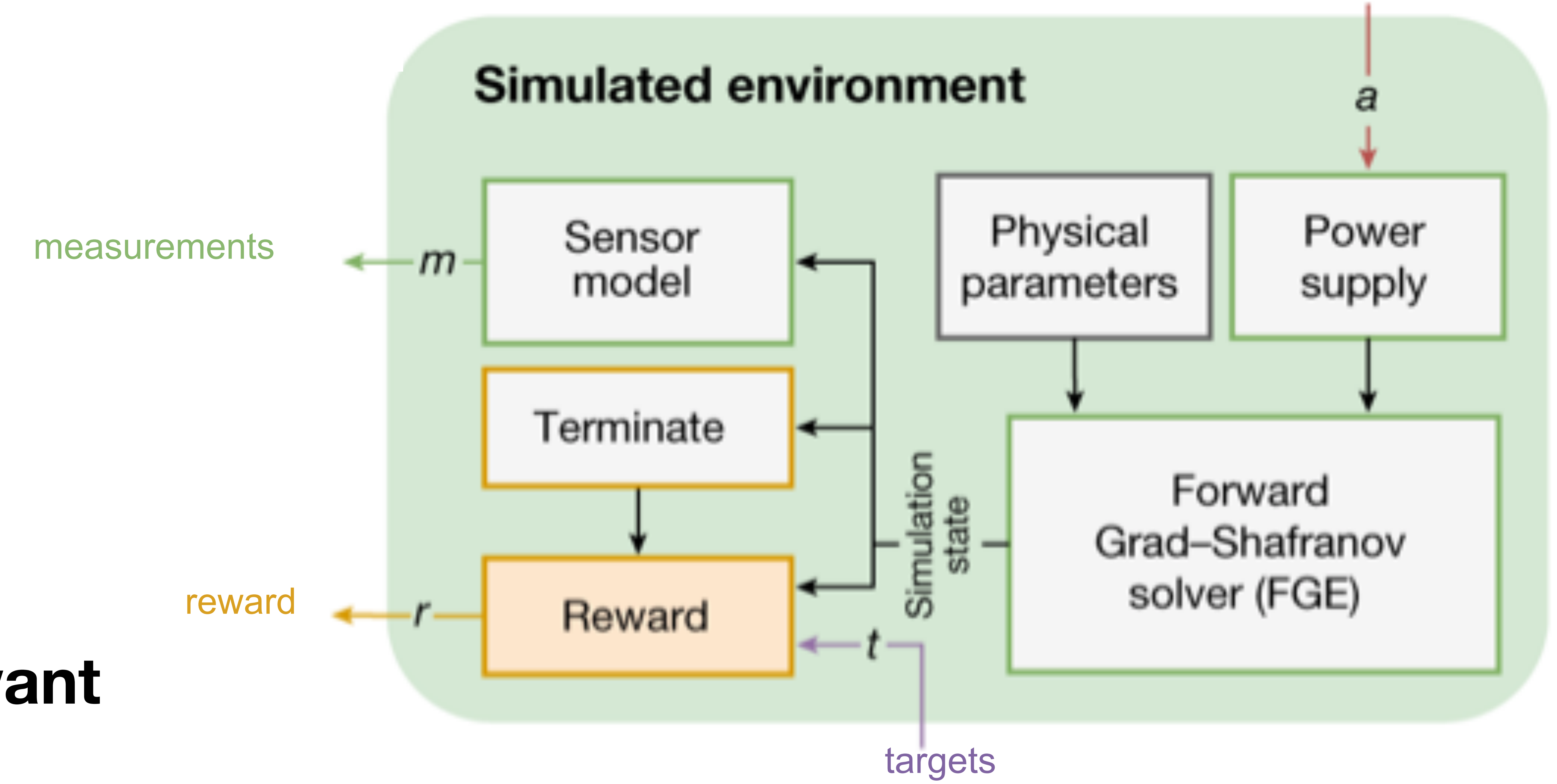

After 15000 trials
Time 00:00.03
Reward -0.000562359414637
Cumulative -0.000562

Environment for learning tokamak axisymmetric control

- Free-boundary simulator: Grad-Shafranov equation coupled to circuit equations
  - In: Voltages on coils. Out: Conductor currents, plasma current distribution, synthetic measurements
  - Used FGE code - part of SPC's Matlab EQuilibrium suite (MEQ) [F. Carpanese, EPFL thesis 2020]
    - Typically ~hours for simulating a few seconds of plasma evolution (50,000 steps/s) - optimisation underway

- Prescribe physical parameters not predicted by model:
  - Plasma conductivity σ_{plasma}
  - Plasma normalised pressure β_p
  - Plasma current profile shape (q_{Axis})

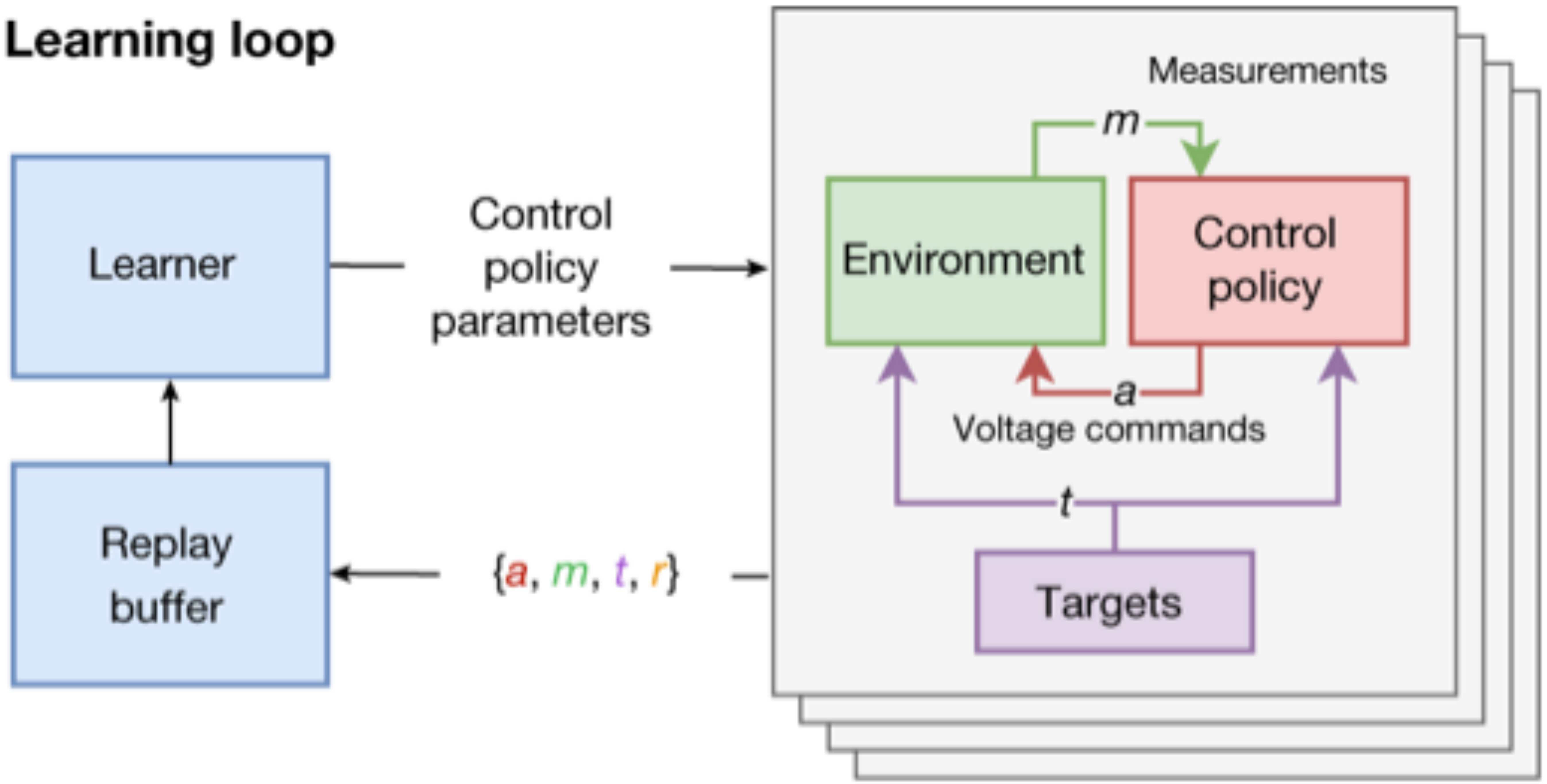
- Termination criteria, examples:
  - Exceeds limits on currents
  - Plasma too far away from target
  - Simulator solver did not converge
- Reward function - specify what we want

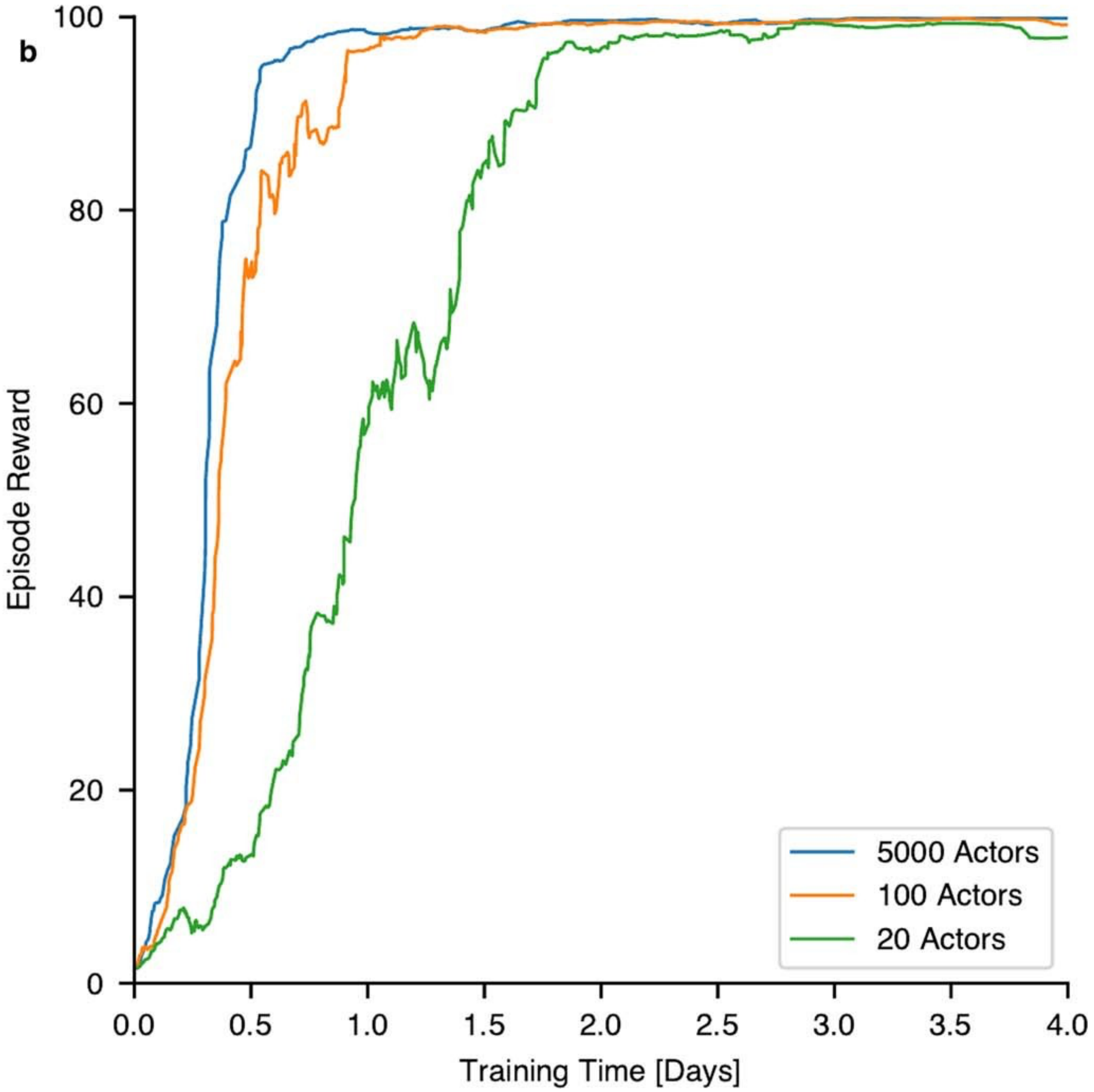


Reward formulation - components

Reward component	Description
Diverted	Whether the plasma is limited by the wall or diverted through an X-point.
E/F Currents	The currents in the E and F coils, in amperes.
Elongation	The elongation of the plasma, this is its height divided by its width.
LCFS Distance	The distance in meters from the target points to the nearest point on the last closed flux surface (LCFS).
LCFS Normalized Flux	The difference in the normalized flux at target points.
Legs Normalized Flux	The difference in normalized flux from the flux at the LCFS at target leg points.
Limit Point	The distance in meters from the actual limit point (wall or X-point) and target limit point.
OH Current Diff	The difference in amperes between the two OH coils.
Plasma Current	The plasma current in amperes.
R, Z	The radial/vertical position of the plasma axis, in meters.
Radius	Half with of the plasma, in meters
Triangularity	The upper triangularity is defined as the radial position of the highest point relative to the median radial position.
Voltage Out of Bounds	Penalty for going outside of the voltage limits.
X-point Count	Return the number of actual and requested X-points within the vessel.
X-point Distance	Returns the distance in meters from actual X-points to target X-points. Only X-points within 20cm are considered.
X-point Far	For any X-point that isn't requested, return the distance in meters from the X-point to the LCFS.
X-point Flux Gradient	The gradient of the flux at the target location with a target of 0 gradient.
X-point Normalized Flux	The difference in normalized flux from the flux at the LCFS at target X-points

The learning setup

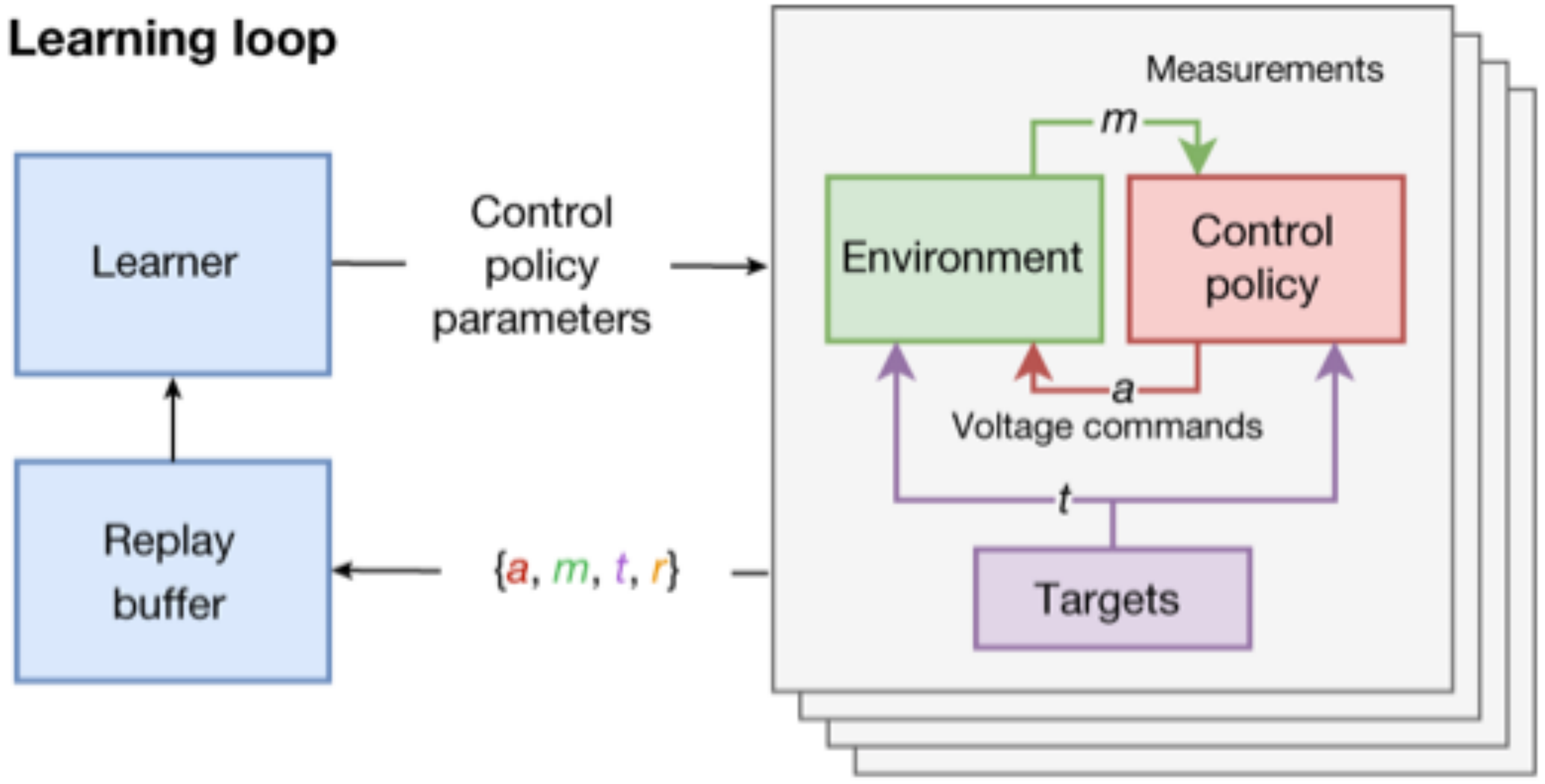




- **Algorithm details:**
  - **Actor-Critic RL: good for control of environments with continuous-valued states**
  - **Distributed implementation: many actors in parallel, results fed to replay buffer continuously for learning**

More learning details

- All learning ran on Google datacenters
  - Computation graph defined using [launchpad]
  - Learner (Critic) ran on TPU - optimized for linear algebra involved in training/evaluating deep neural networks
  - Simulations (Actors) ran on CPUs - easier since single-thread application



Actor-critic method for reinforcement learning

- Critic learns the *Q function* from data generated by actors interacting with environment
- Actor learns a policy π by taking policy gradients on learned *Q function*

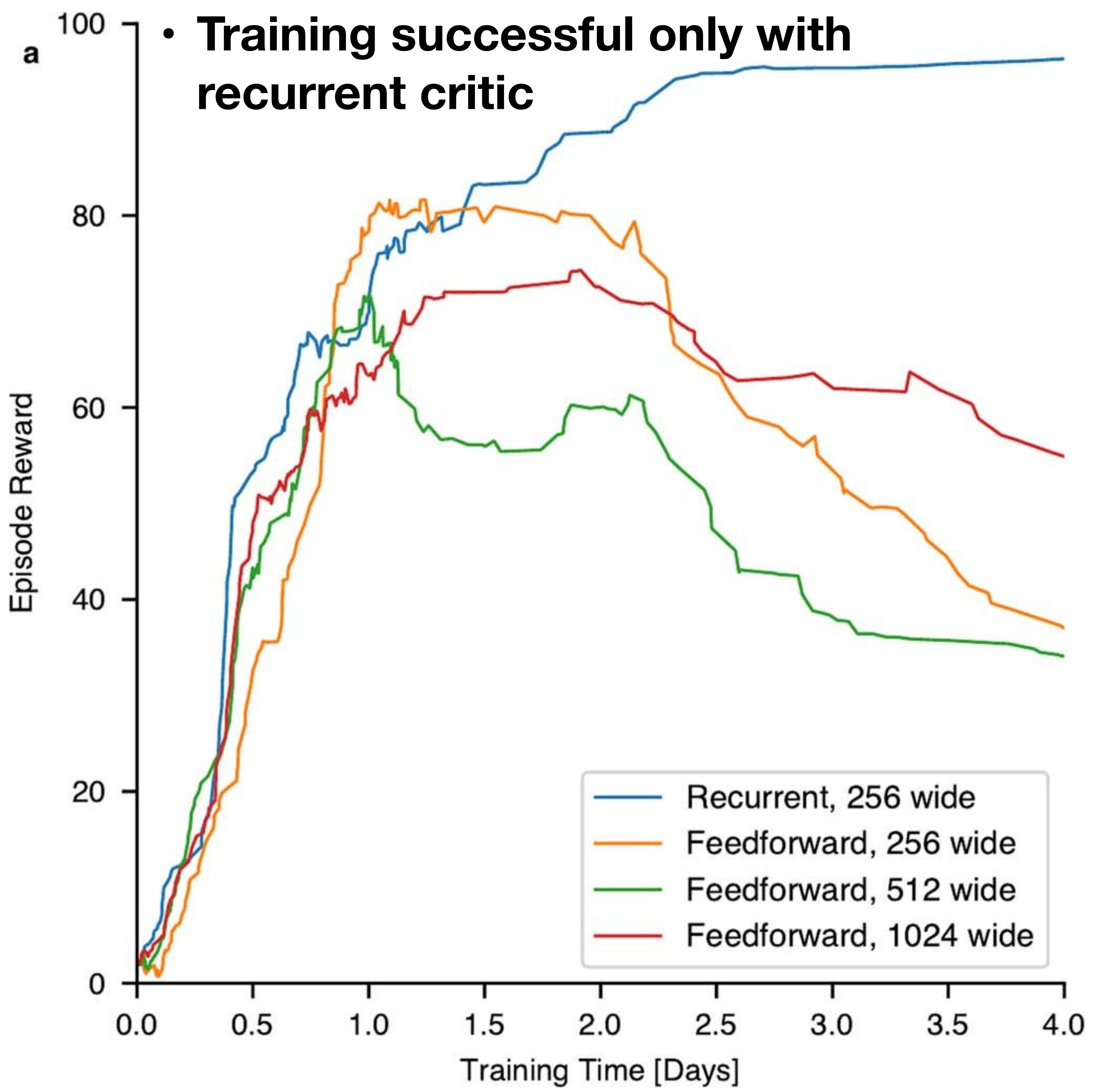
- Advantages (in general)
  - Stability in training, flexibility, efficient use of data
- Advantages (for our problem):
  - Value function: can be large & have access to privileged information (e.g. full state)
  - Actor (Policy) can be small for real-time applications

- Deep Reinforcement Learning means using (Deep) Neural Networks for both the Value function and Policy

Importance of an asymmetric, recurrent critic

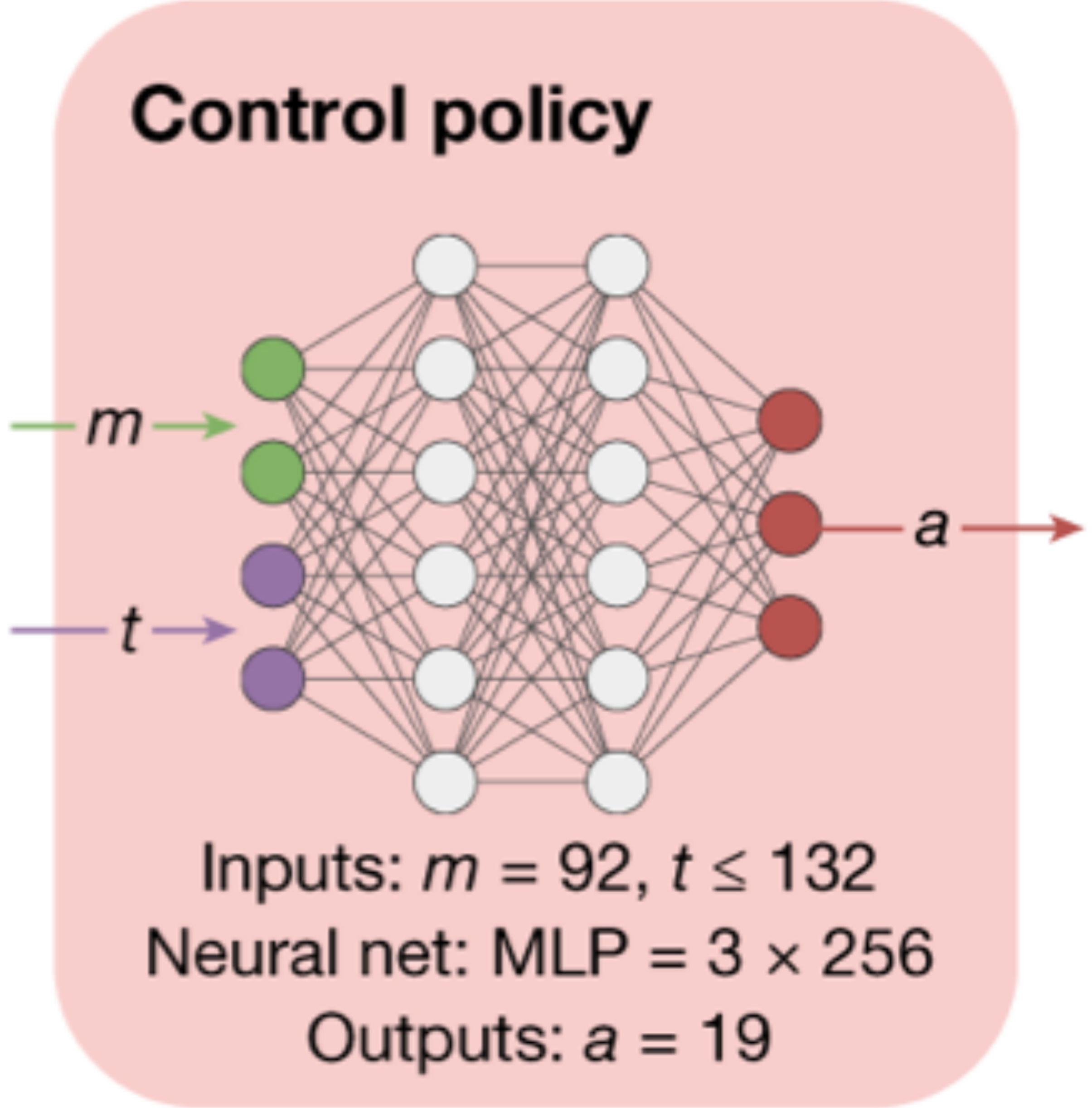
- **Actor-critic RL allows asymmetry**
  - **Large, recurrent critic**
    - Only used in training
    - Sees entire simulator state
  - **Small, feedforward actor**
    - Runs in real-time
    - Sees only measurements

- **Training successful only with recurrent critic**



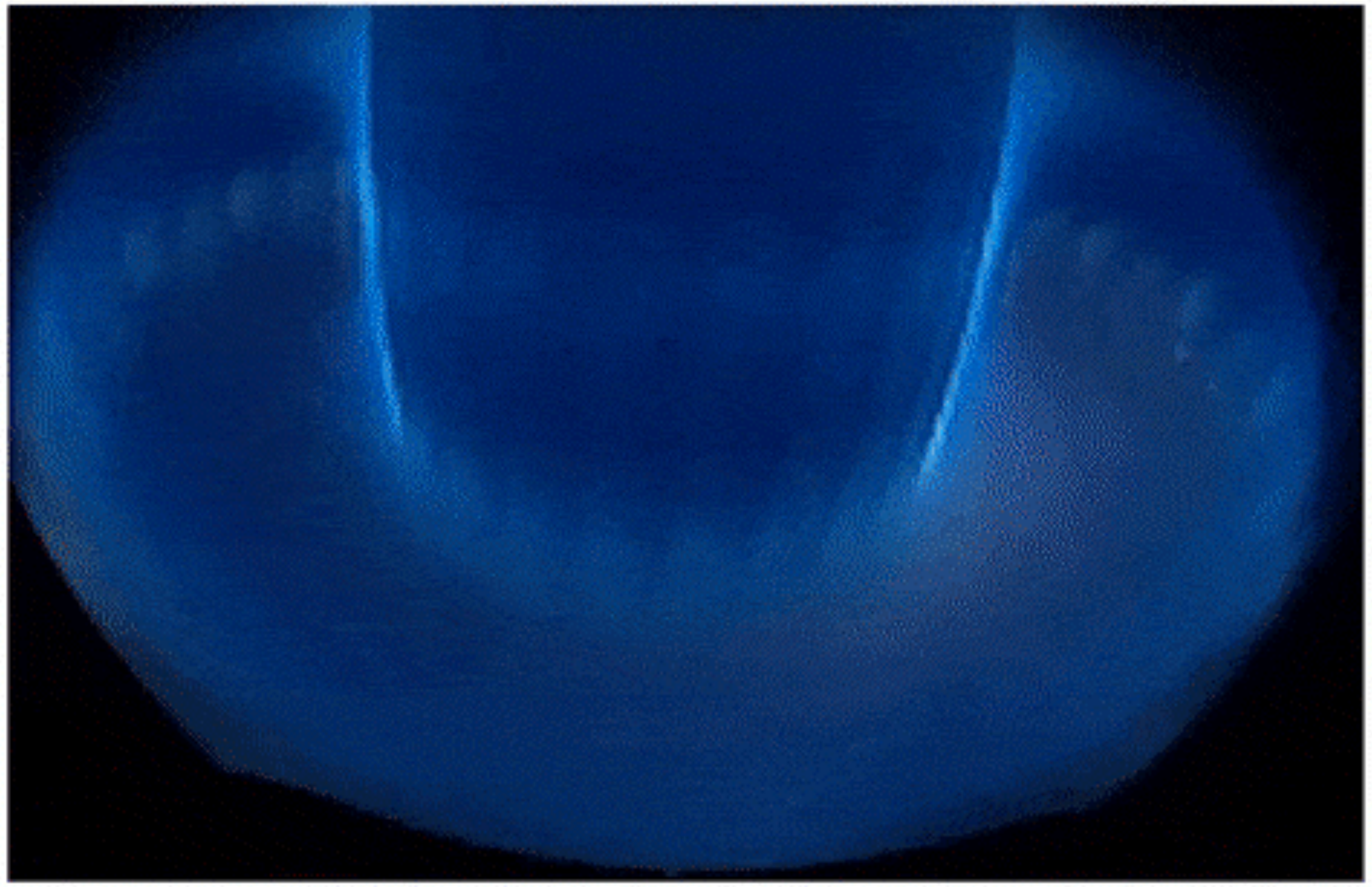
Bringing learned agents to the TCV tokamak hardware

- **Deploy agent to TCV control system**
  - Compiled Agent NN to binary - wrapped in Simulink S-function
  - TCV control system code does signal routing & calibration & traditional controller
  - Use automated code generation to deploy to real-time environment
    - One CPU thread, 10kHz
    - MARTe2 real-time framework *[https://vcis.f4e.europa.eu/marte2-docs]*

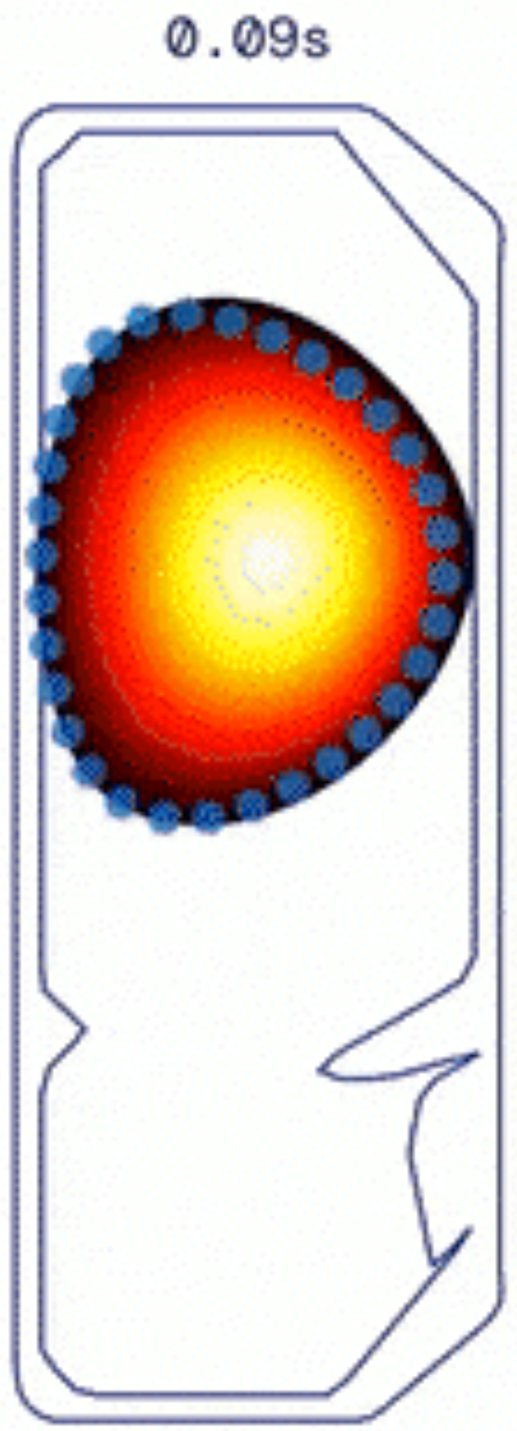


- **Needed to add randomization in training to make the controller robust (address 'sim2real gap')**
  - Perturb plasma internal parameters (q_A, β_p, σ_{plasma}) that are external inputs to the GS equation
  - Perturb observations (measurements) and actions (input voltages).
- **Needed some trial-and error to design the reward function**
  - Finally we found 1 reward function that worked for most cases

Result - demonstration shot

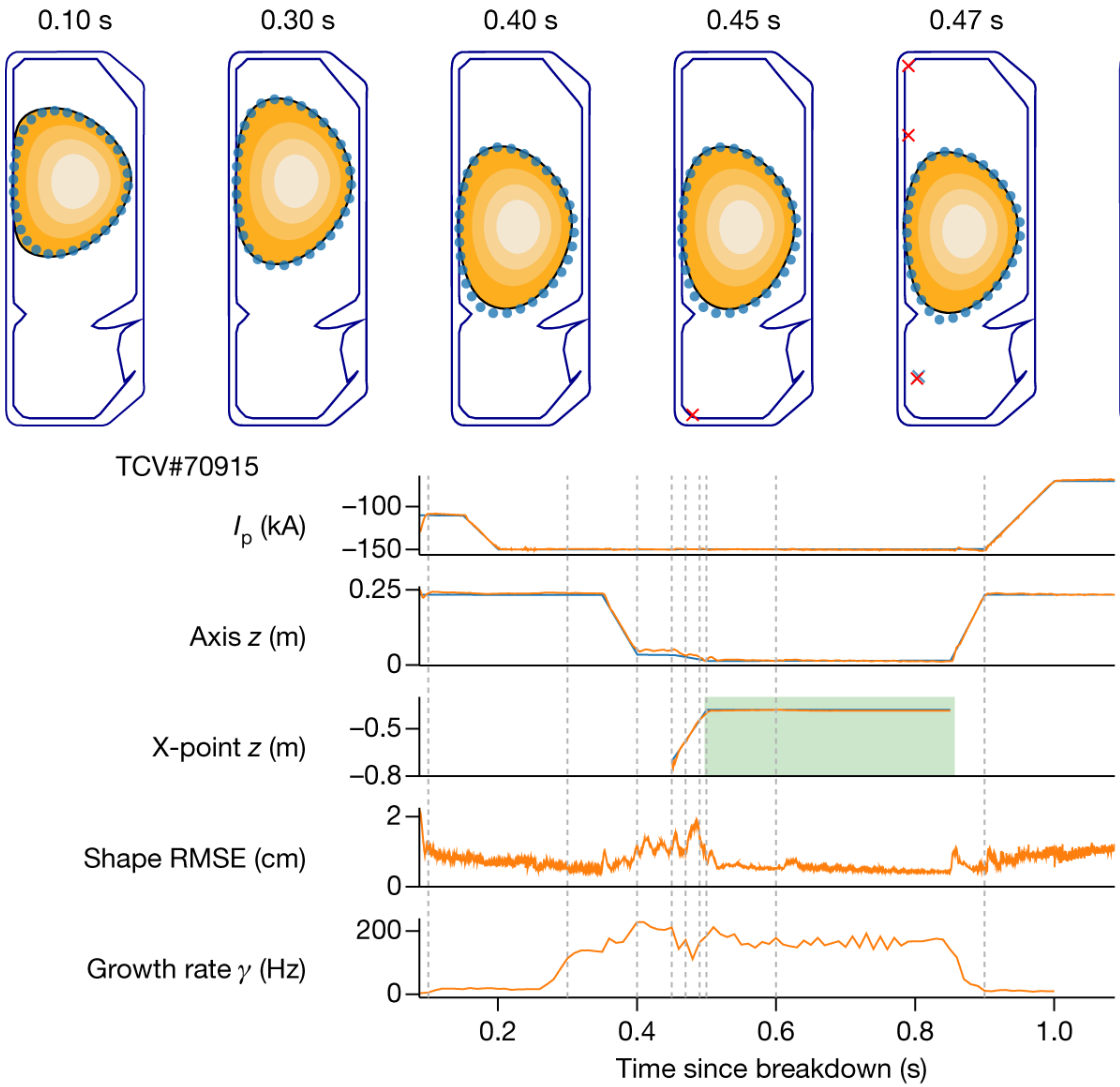


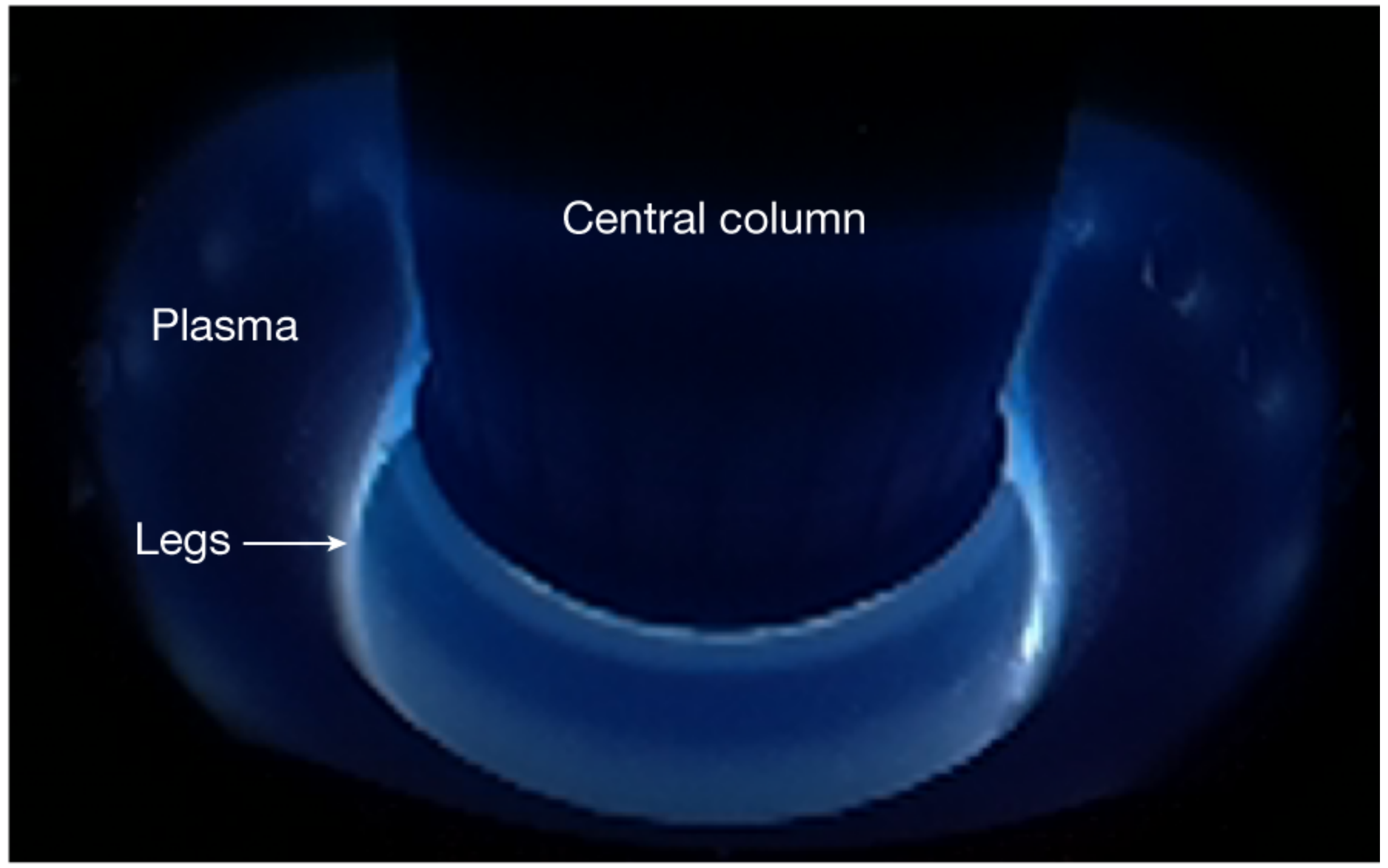
View from inside the tokamak



Plasma state reconstruction

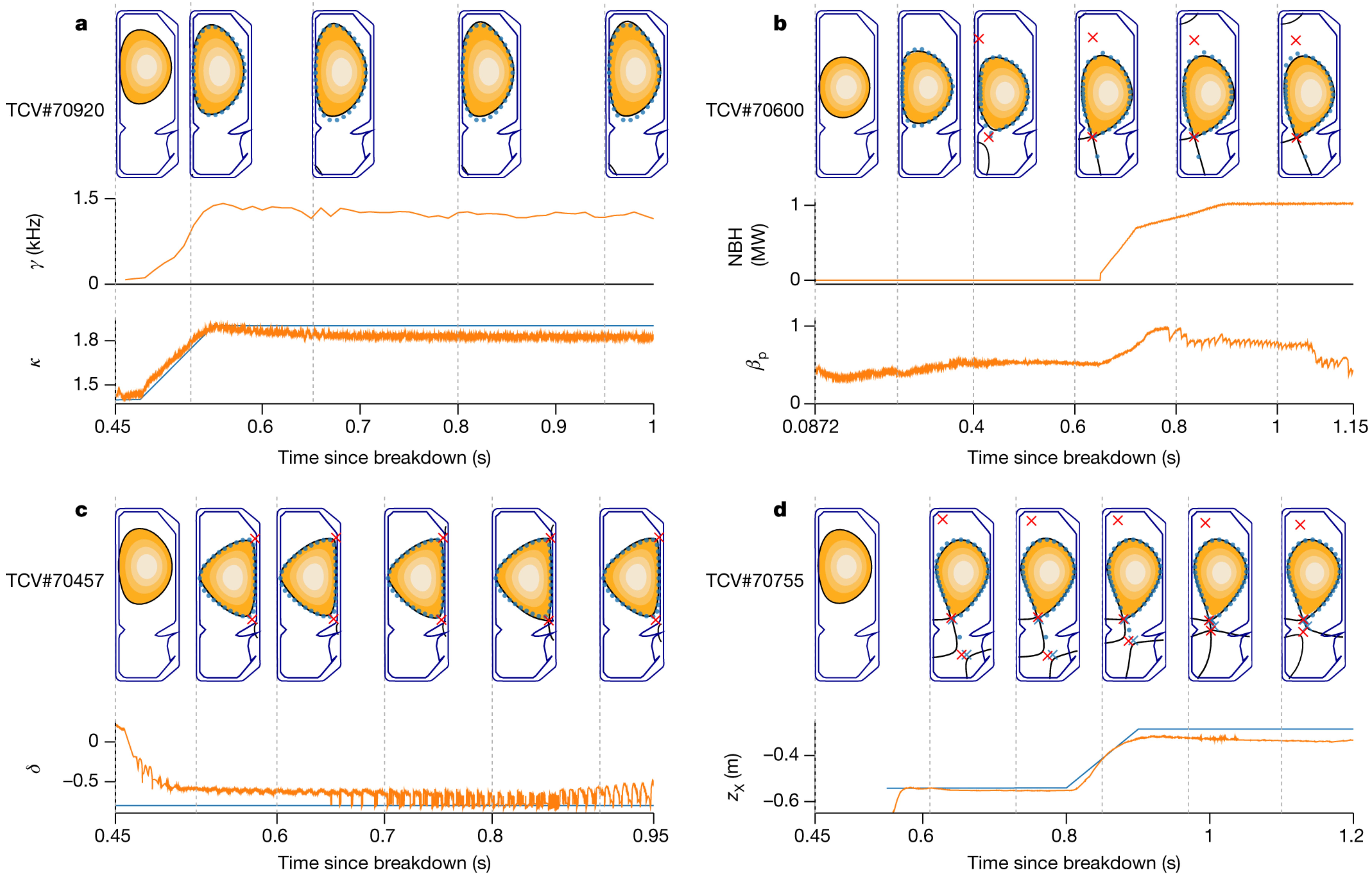
Result - demonstration shot



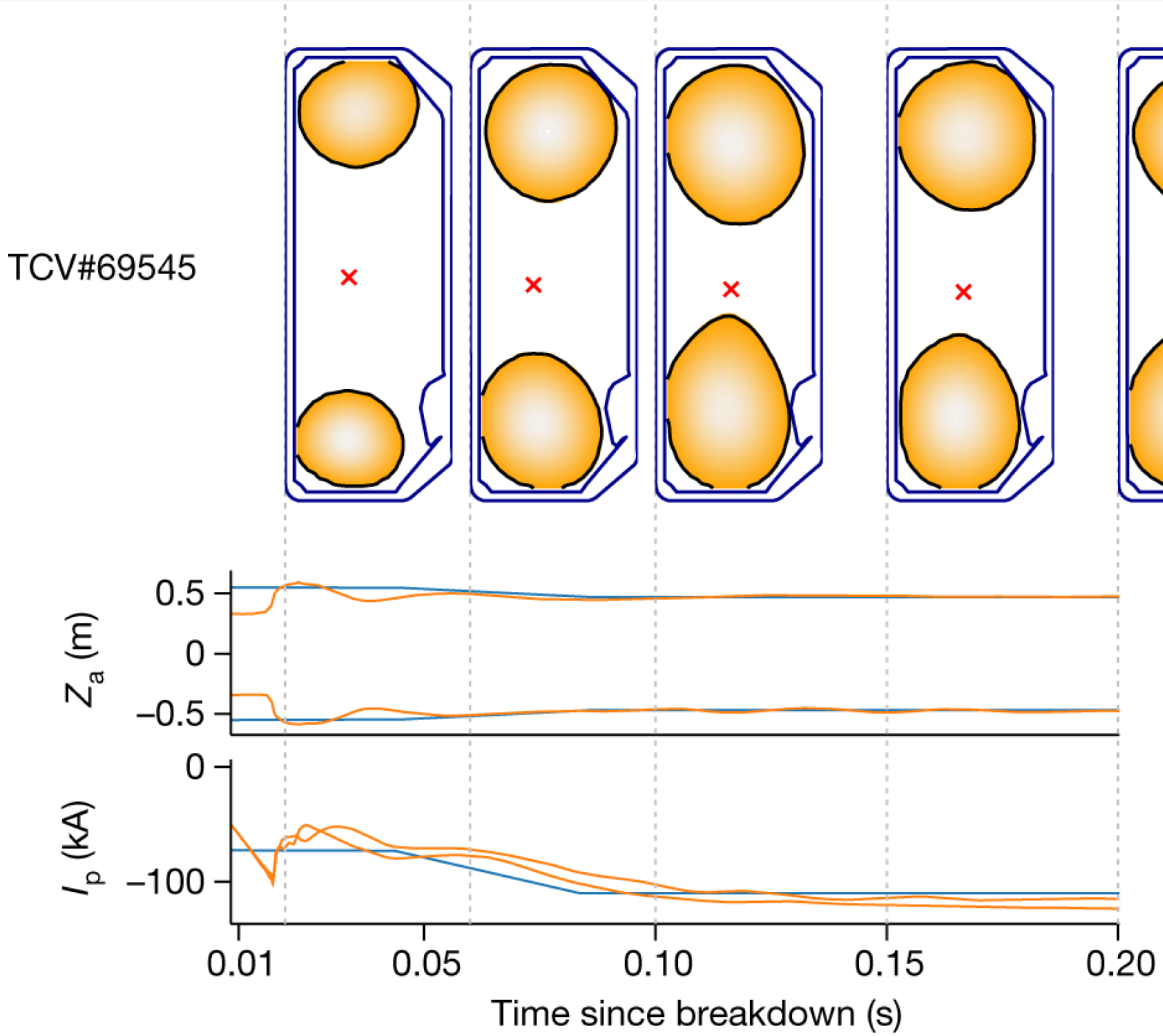


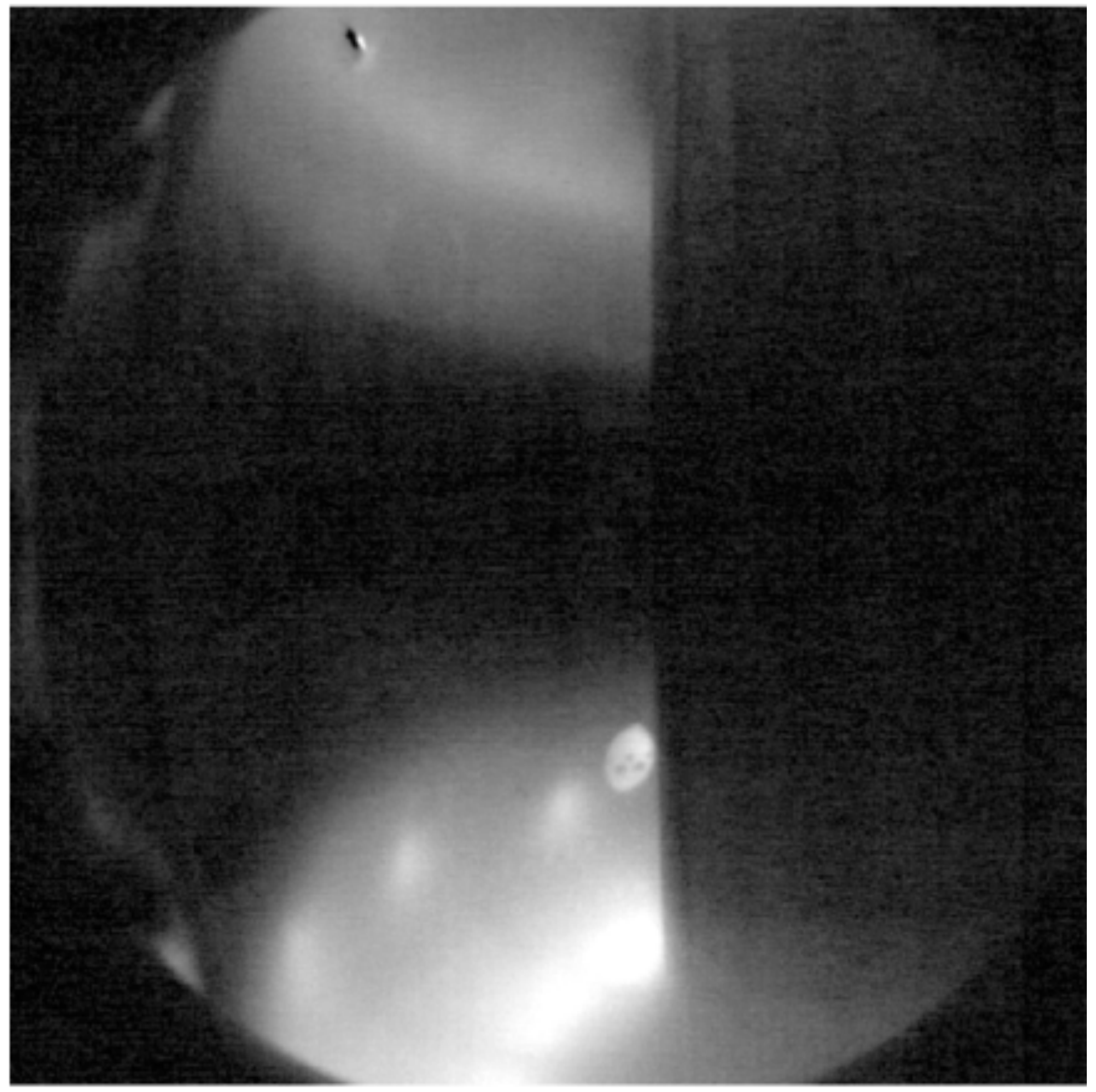
Inside view at 0.6 s

Various plasma shapes controlled in in TCV with reinforcement Learning



Opening new frontiers for TCV: droplet plasmas





Features of traditional / RL controllers

Traditional controllers (MIMO PID)	Our Reinforcement Learning implementation
Separate error for each control loop, need to compute error online	Single reward function, no explicit error signals or state estimation
Need separate tuning of various control loops, using linear control techniques assuming (local) linearity	Joint solution to entire stabilization/control problem including any nonlinearities
Need domain knowledge to break down control problems, design separate controllers	Domain knowledge is in simulator. Just define reward functions
Tuning of several control parameters	Reward function engineering
(Usually) Clear relation between parameters and aspects of control performance	Black-box agent

Conclusions

- **Demonstrated RL for closed-loop magnetic control of tokamak plasmas, trained in free-boundary simulations and <u>tested on TCV experiments</u>**
  - **Implementing 10kHz controller with 100+ measurements, 20 actions is a milestone for RL on real-world systems in terms of complexity**

- **Key challenges:**
  - **Accurate and fast <u>models</u> of system to control**
  - **Reinforcement learning with scarce data**
  - **Reward function engineering**
  - **Real-time implementation & interfacing with existing PCS**
  - **Domain randomization for controller robustness**

J. Degrave, F. Felici et al., Nature 602, 414–419 (2022)

Article

Magnetic control of tokamak plasmas through deep reinforcement learning

https://doi.org/10.1038/s41586-021-04301-9
Received: 14 July 2021
Accepted: 1 December 2021
Published online: 16 February 2022
Open access

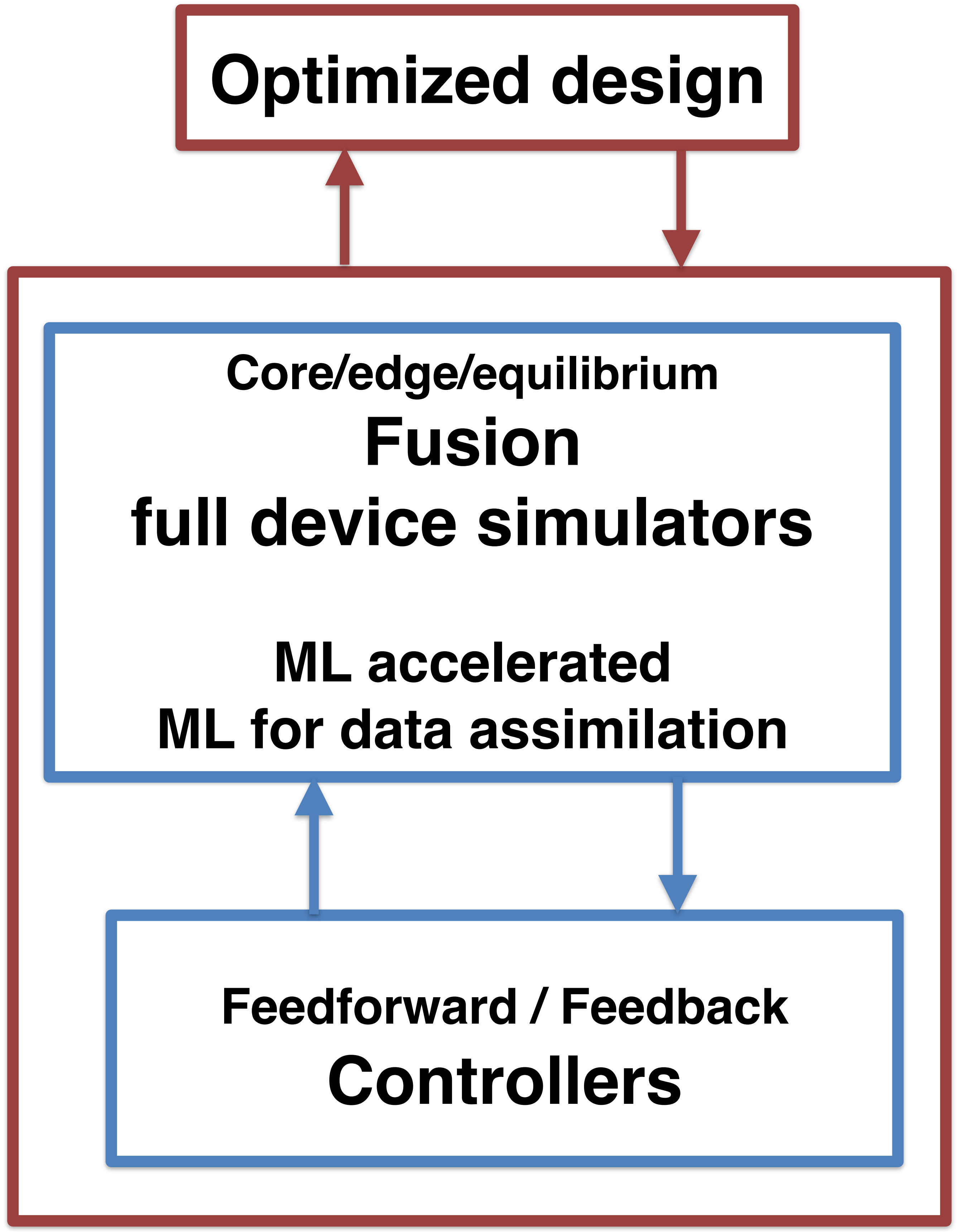
Jonas Degrave[1,3], Federico Felici[2,3]✉, Jonas Buchli[1,3]✉, Michael Neunert[1,3], Brendan Tracey[1,3]✉, Francesco Carpanese[1,2,3], Timo Ewalds[1,3], Roland Hafner[1,3], Abbas Abdolmaleki[1], Diego de las Casas[1], Craig Donner[1], Leslie Fritz[1], Cristian Galperti[2], Andrea Huber[1], James Keeling[1], Maria Tsimpoukelli[1], Jackie Kay[1], Antoine Merle[2], Jean-Marc Moret[2], Seb Noury[1], Federico Pesamosca[2], David Pfau[1], Olivier Sauter[2], Cristian Sommariva[2], Stefano Coda[2], Basil Duval[2], Ambrogio Fasoli[2], Pushmeet Kohli[1], Koray Kavukcuoglu[1], Demis Hassabis[1] & Martin Riedmiller[1,3]

Outlook

- Improve on current RL controller implementations for tokamak magnetic control
  - Recurrent policies / Tunable performance / Use experimental data

- Use RL for optimizing plasma performance (= fusion power)
  - Need to include physics of internal plasma (temperature, density) evolution in environment, much more complex physics models.
    - 'Whole device simulations' - digital twins / flight simulators
  - Learn better operating scenarios, or co-design device and plasma scenario

- Bright future for more applications of reinforcement learning
  - For accelerating fusion science: improving plasma performance, control & design new devices
  - For application to more complex real-world systems, in particular where good models exist



Some other recent ML applications in Nuclear Fusion

- **ML accelerated 'surrogates' of integrated simulation bottlenecks**
  - Model of neutral beam heating sources (NUBEAMNET, [M.D. Boyer 2019])
  - Models of turbulent transport fluxes (QLK-NN, [I. Van de Plassche 2020])
  - Models of edge plasma pressure gradients (EPED-NN [O. Meneghini *et al,* 2021])

- **Learning macroscopic tokamak behaviour from large datasets: apply for state reconstruction and control**
  - RL for feedforward control of internal plasma quantities [J. Seo et al 2021]
  - Learning plasma internal profile evolution [J. Abbate, 2020]

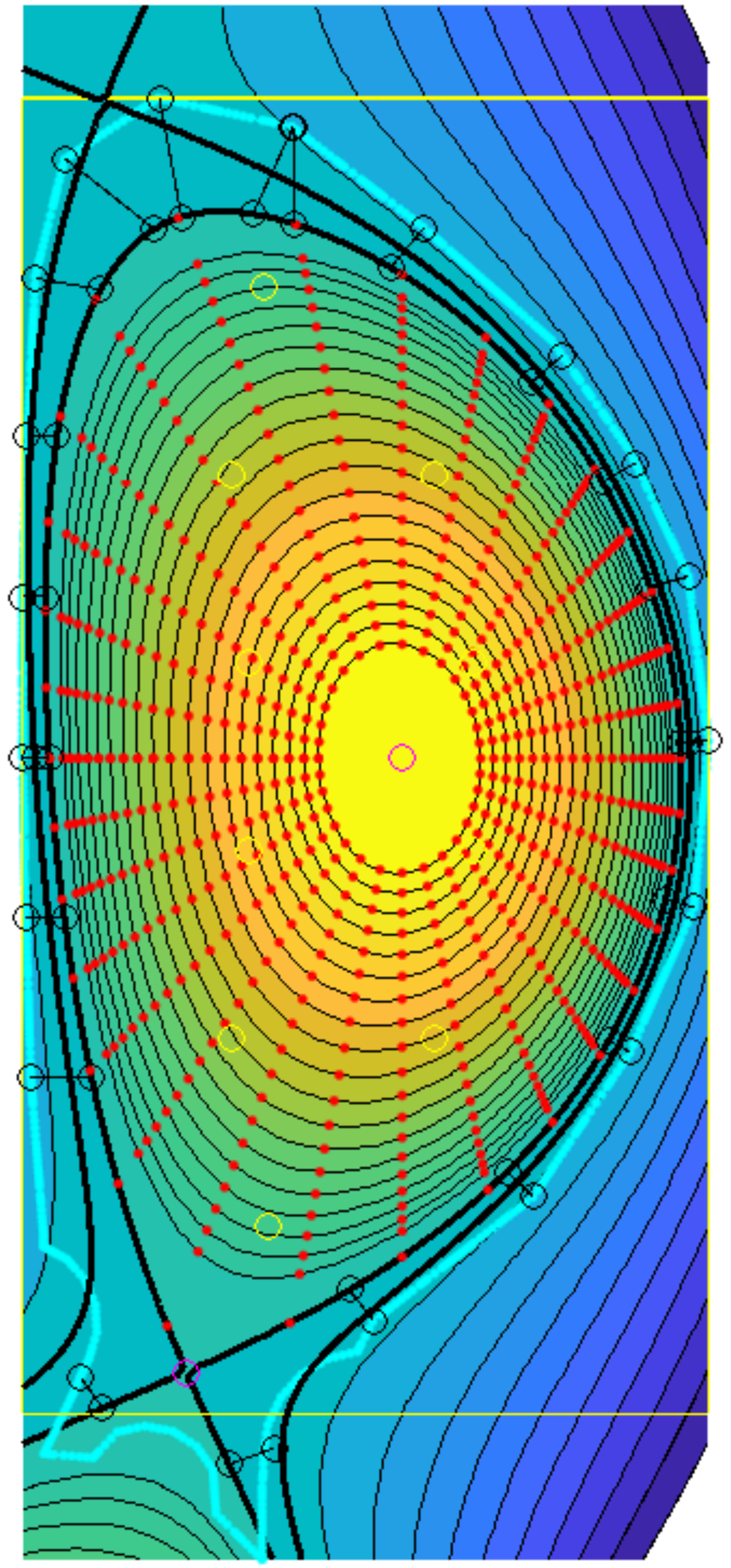
- **(Real-time) plasma event detection based on diagnostic signals - for plasma state interpretation and machine protection**
  - Confinement state detection: [F. Matos 2020]
  - Alfvén Eigenmode classification: [A. Jalalvand 2021]

Backup slides

Real-time equilibrium reconstruction

- **Find plasma equilibrium that best fits magnetic measurements**
- **Formulate least-squares problem: min || y - A(x)*x ||^2**
  - **Fit quantities x:**
    - **Parameters of internal plasma profiles (p’,FF’)**
    - **Unknown passive & active currents**
  - **Measurements y:**
    - **Magnetic and other measurements**
  - **A(x) regression matrix: contains a plasma-dependent part**
- **Bottlenecks:**
  - **Solving GS equation in real-time**
  - **Build plasma-dependent part of regression matrix**
- **Modern RT GS solvers can do this routinely in <1ms, e.g.**
  - **LIUQE [J-M. Moret, Fus. Eng. Des 2015]**
  - **PEFIT [Y. Huang et al, Fusion Eng. Des 2016]**

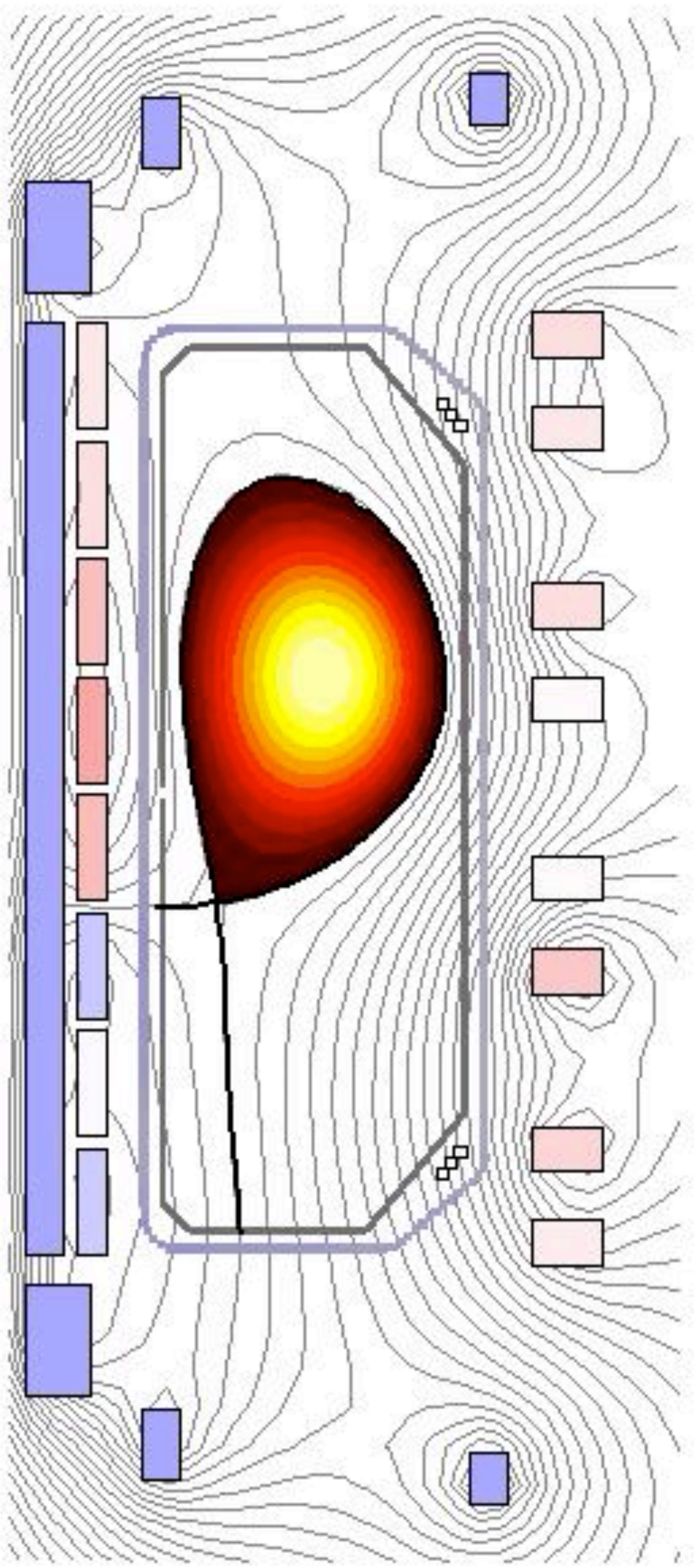
LIUQE-ITER(C)#0 85.0000s/12001
r,z=6.453,+0.525m-0.5mm:4197
Ip=15004kA bp=0.58 li=0.85



Physics model for axisymmetric magnetic control

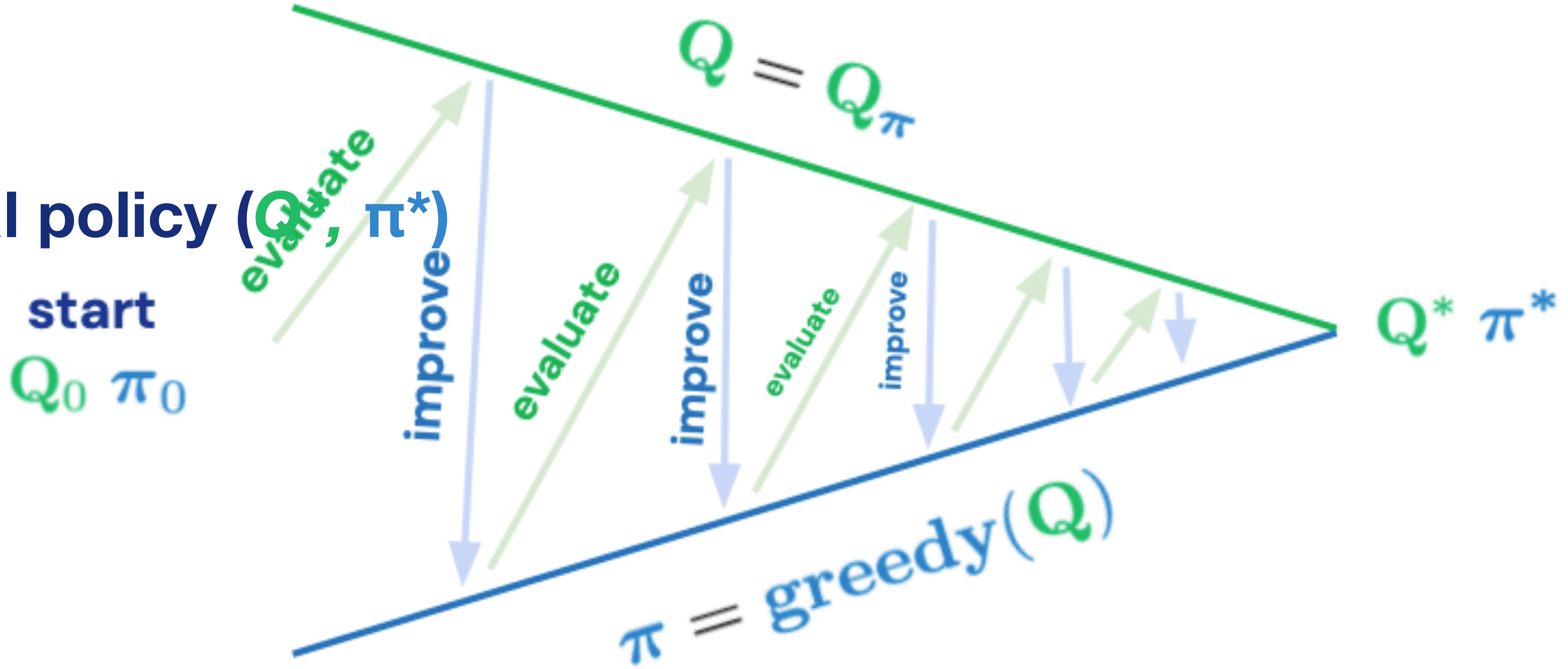
- **Free-boundary Grad-Shafranov equilibrium evolution solvers**
  - **Grad-Shafranov equation**
    - **Static, ideal MHD axisymmetric force balance + Ampère's law + Faraday's law + internal plasma parameters**
  - **(linear) dynamic circuit equations for solid conductors**
    - **Current evolution in passive + active conductors due to induced and imposed voltages**
    - **Current evolution in plasma: 0D lumped equation or 1D current diffusion**
- **Power supply models (often simplified)**
- **(Magnetic) sensor models**

Example of TCV feedback controlled simulation using FGE



Flavours of RL (1)

- **Value-based methods**
  - **Learn the *Q function* by interacting with the environment with a given policy π + randomization (to explore)**
  - **Compute the optimal policy π for the new value function.**
  - **Iterate until we find the optimal value function and optimal policy (Q*, π*)**

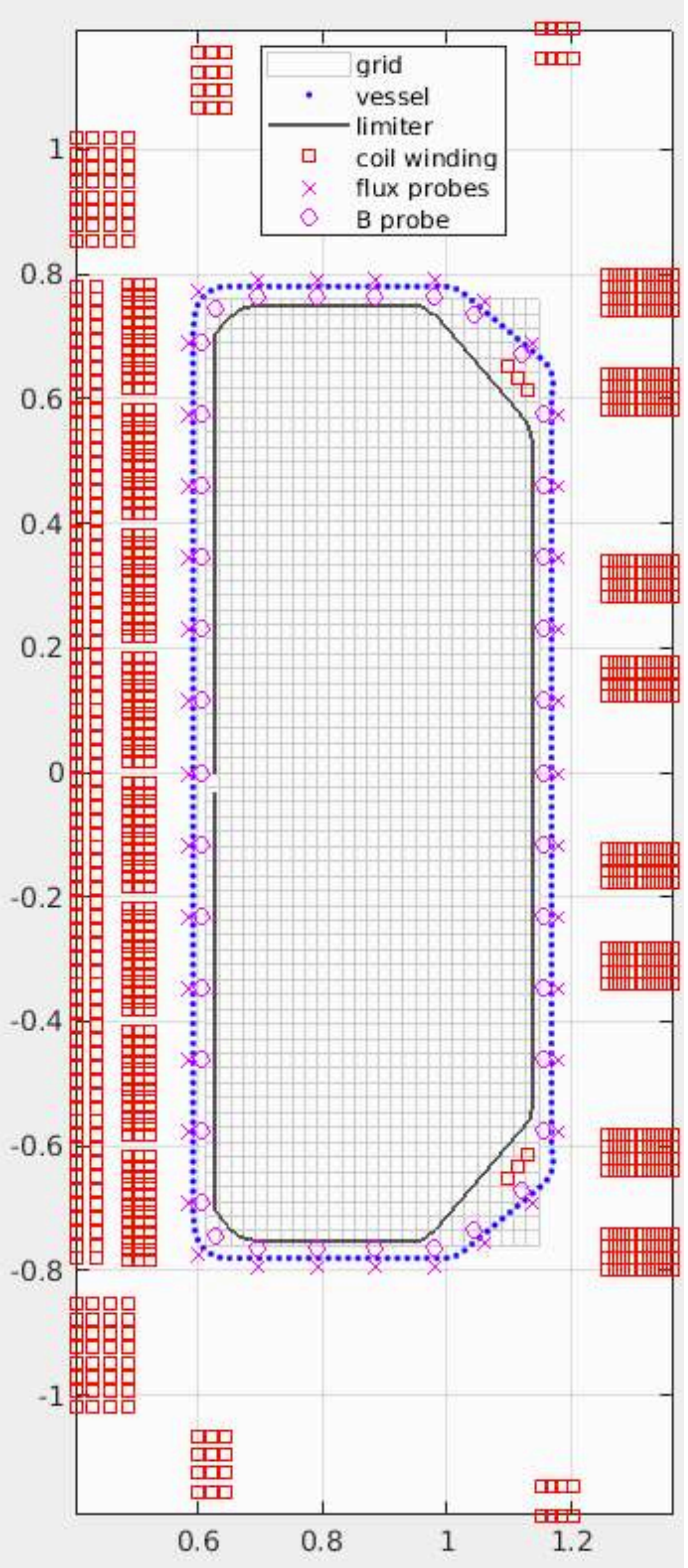


- **Policy-based methods**
  - **Parametrize the policy and learn that directly:**
  - **Interact with the environment while perturbing the policy.**
  - **Computing a policy gradient from the sampled trajectories**

MEQ core model & numerics

- **Model:**
  - **Flux solved on a square computational grid covering region within the limiter**
  - **Effect of external conductors on currents in grid is modelled via Green's functions**
  - **Conductors modelled by a set of discrete filaments**
  - **No ferromagnetic elements, no SOL currents**
- **Numerics:**
  - **Finite Differences for Laplace-like operator**
  - **'Lackner's trick' for effect of plasma currents on flux boundary conditions.**
- **Common modules**
  - **Generic mag. axis, x-point, LCFS finder**
  - **Fast post-processing for integral quantities, contouring, contour integral calculations (q profiles etc), gaps**
  - **Interpolation of fluxes, fields on desired control points**

MEQ geometry representation for TCV



FGE - Forward Grad-Shafranov Evolution solver

Ref: [F. Carpanese EPFL PhD thesis 2020]

- **Solves coupled equations:**
  - **ODEs (Kirchhoff circuit laws) for active and passive circuits including induced voltages from changing plasma distribution.**
  - **Plasma force balance (Grad Shafranov equation).**
  - **Constraint equations for p', TT'**
    - **Presently combinations of scalar constraints like I_p, $\beta_{p/t}$, l_i, q_A**
    - **Optionally: plasma Current Diffusion Equation:**
      - **Ohm's law for the plasma: (presently 0D only)**
- **Modules:**
  - **Feedback controller acting on coil voltages (test shape, current, position control)**
  - **Simple power supply models**
  - **Monitoring of coil current and force limits**
- **Numerical solver:**
  - **Monolithic, Jacobian Free Newton Krylov solver**
  - **Numerical sensitivities for control design/optimisation use**
- **Benchmarked vs FEEQS.M**

Example of TCV feedback controlled simulation