Reinforcement Learning for Fusion: Self Driving Cars to Controlled Fusion

				
Jeff Schneider Professor	Ian Char MLD PhD	Youngseog Chung MLD **PhD**	Viraj Mehta RI PhD	Willie Neiswanger Stanford Postdoc

Auton Lab
Carnegie Mellon

Carnegie Mellon
THE ROBOTICS INSTITUTE

Hardware

Side and rear facing cameras

Work in collaboration to construct a continuous view of the vehicle's surroundings

Top mounted lidar units

Provide a 360° 3-dimensional scan of the environment

GPS

IMU

wheel encoders

Forward facing camera array

Focus both close and far field, watching for braking vehicles, crossing pedestrians, traffic lights, and signage

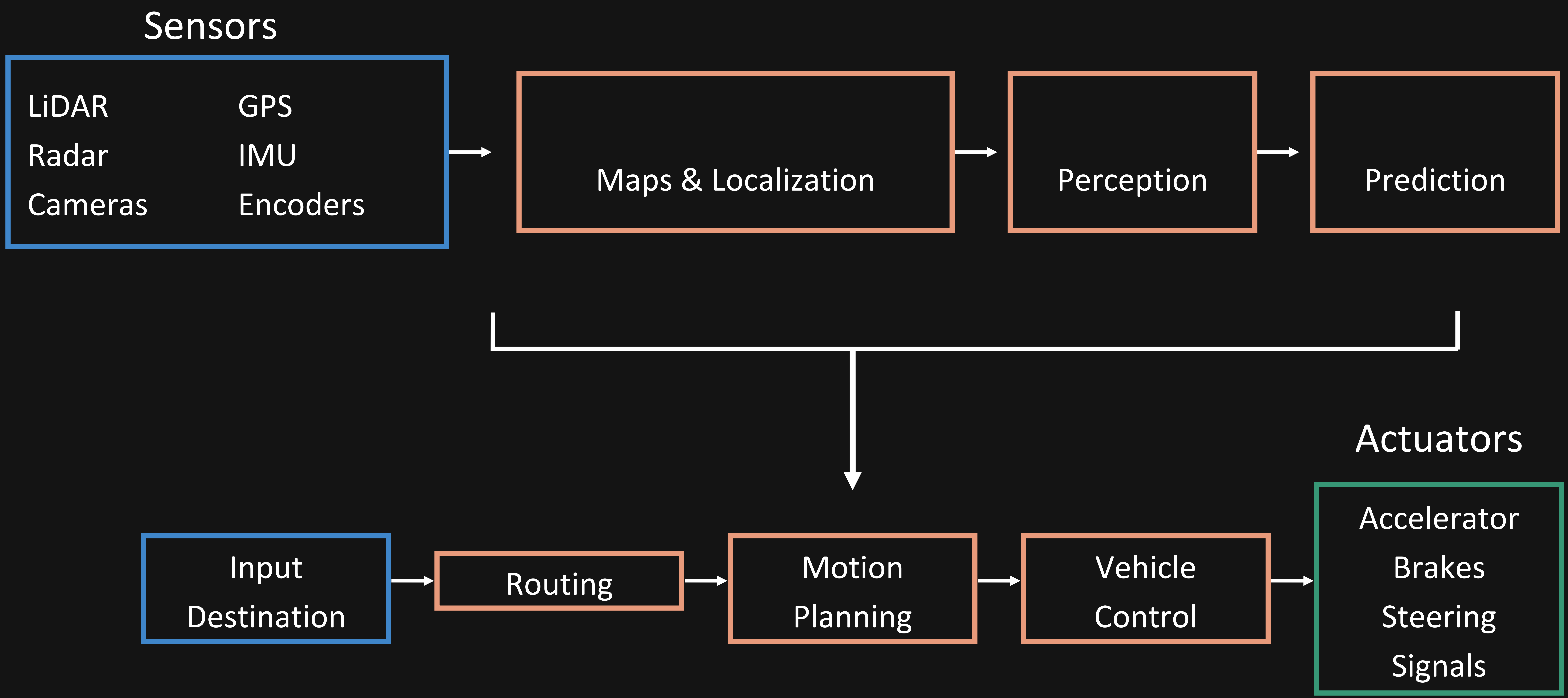
360° radar coverage

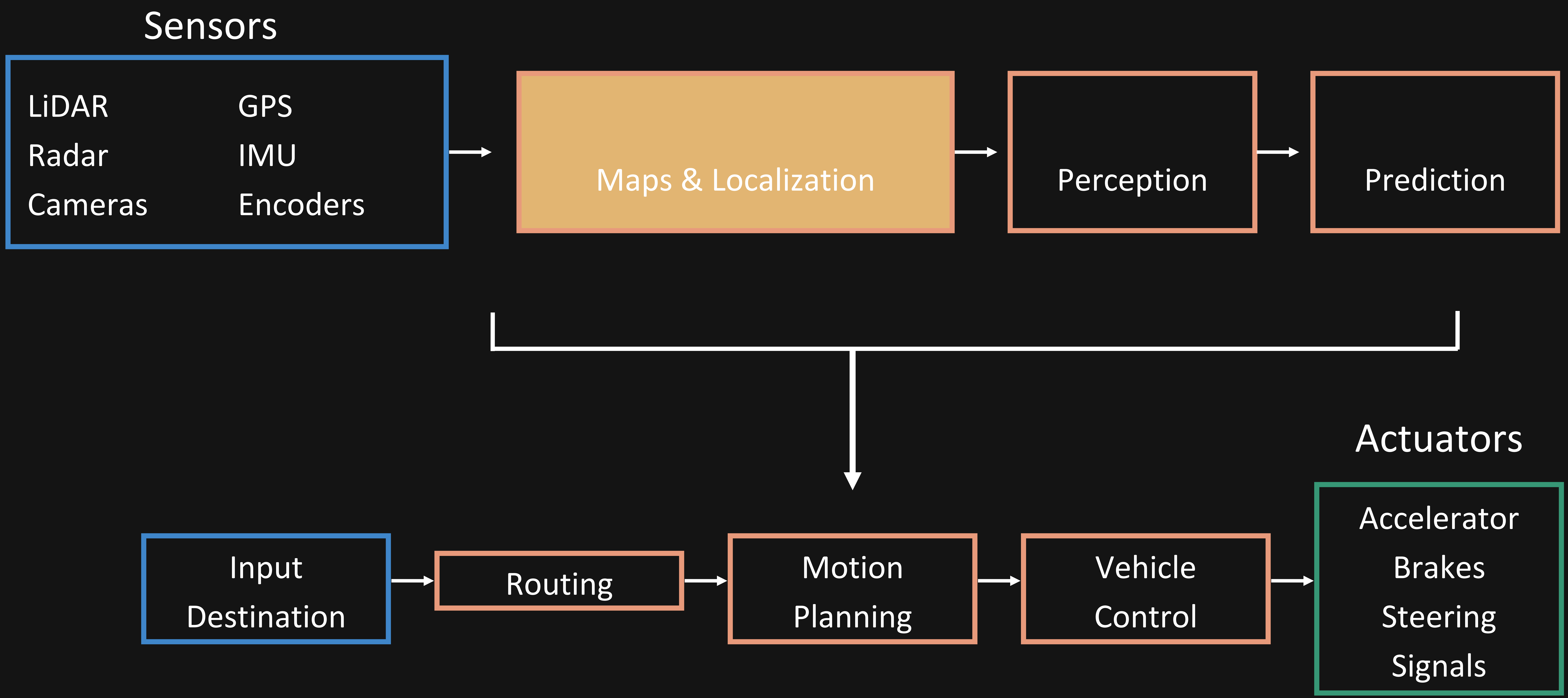
Custom designed compute and storage

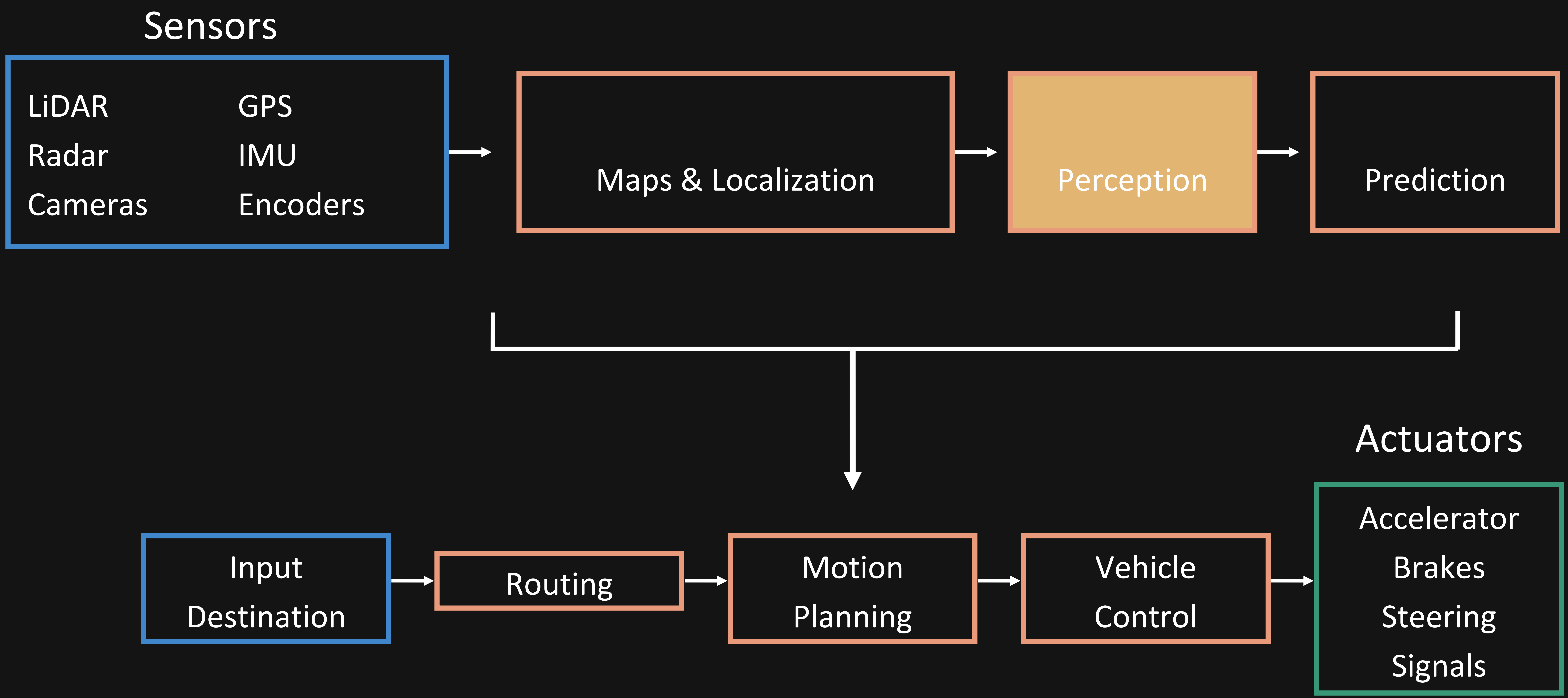
Allow for real-time processing of data while a fully integrated cooling solution keeps components running optimally

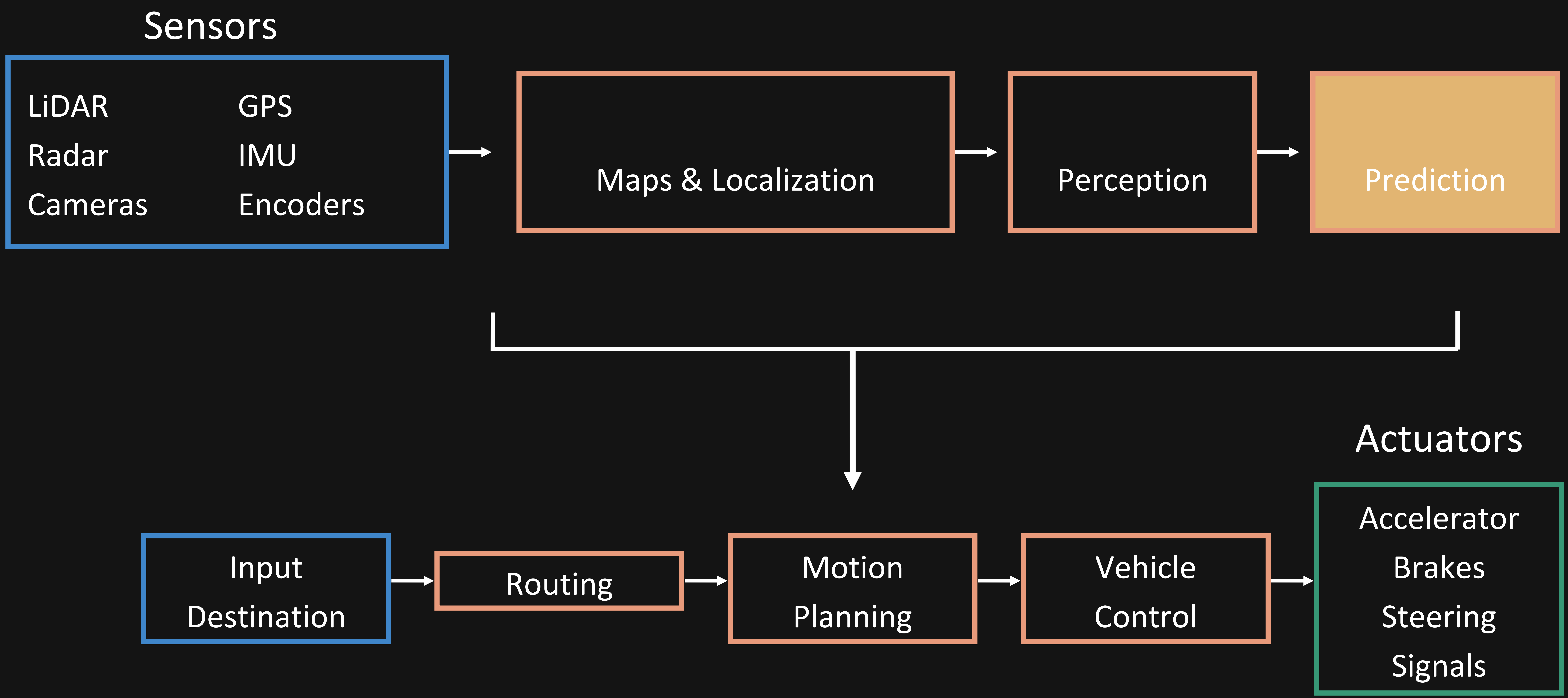
Autonomy Software

System Architecture

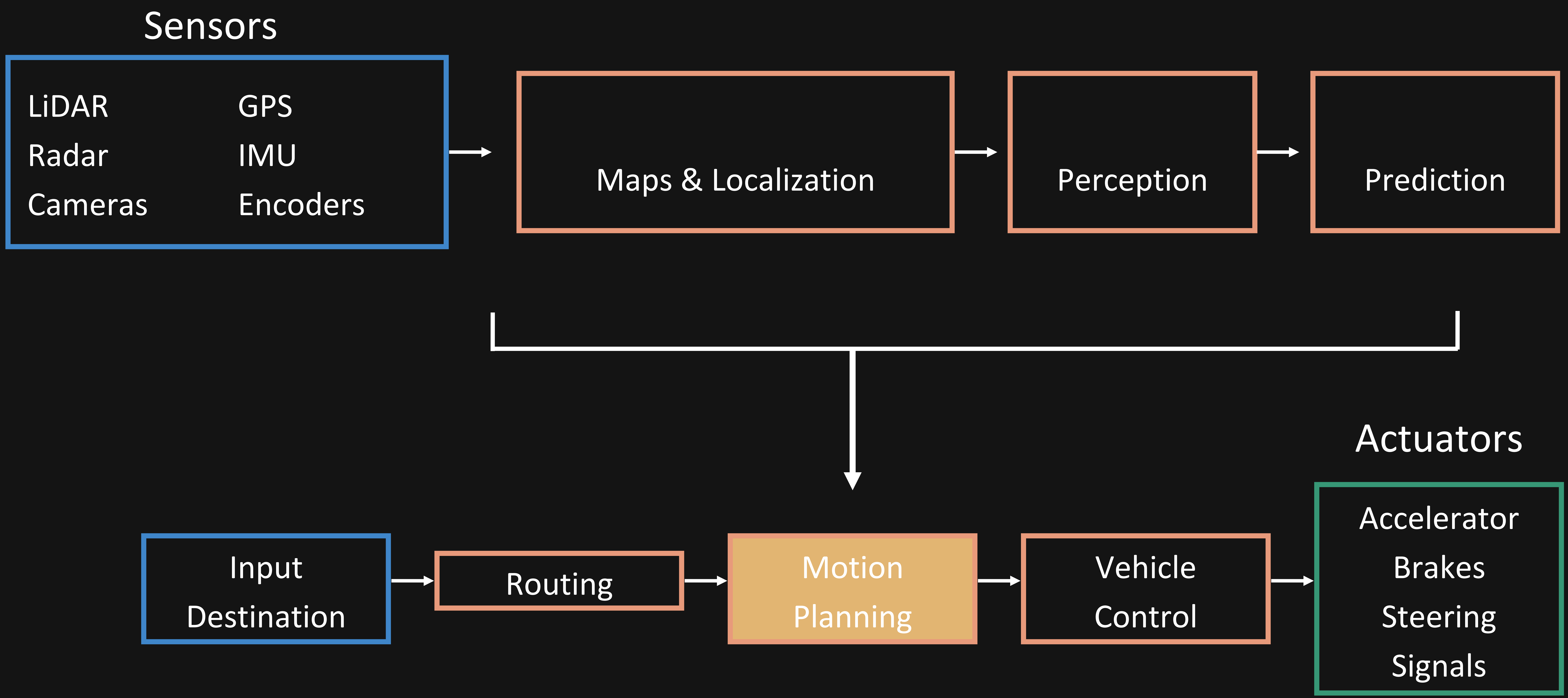

Sensors
LiDAR
Radar
Cameras
GPS
IMU
Encoders
Maps & Localization
Perception
Prediction
Actuators
Input
Destination
Routing
Motion
Planning
Vehicle
Control
Accelerator
Brakes
Steering
Signals

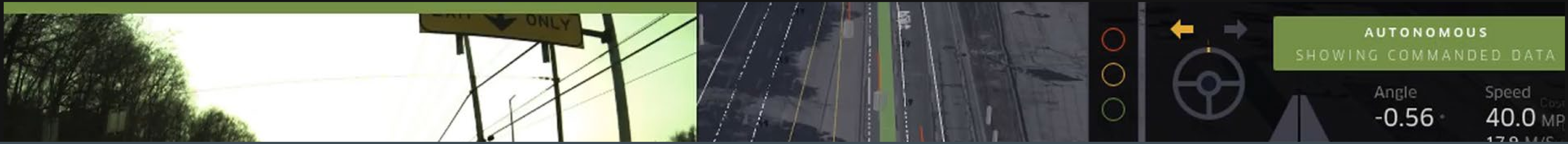

Sensors
LiDAR
Radar
Cameras
GPS
IMU
Encoders
Maps & Localization
Perception
Prediction
Actuators
Input
Destination
Routing
Motion
Planning
Vehicle
Control
Accelerator
Brakes
Steering
Signals


Sensors
LiDAR
GPS
Radar
IMU
Cameras
Encoders
Maps & Localization
Perception
Prediction
Actuators
Input
Destination
Routing
Motion
Planning
Vehicle
Control
Accelerator
Brakes
Steering
Signals

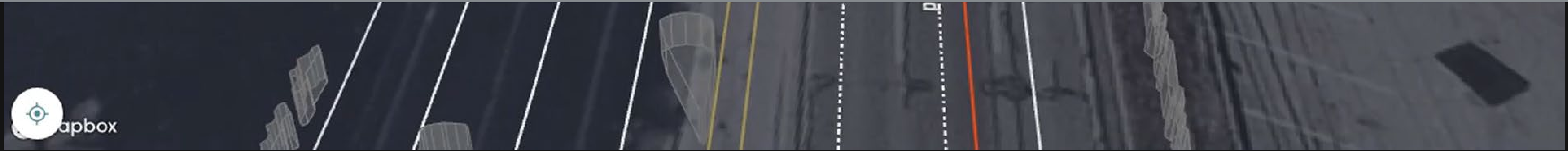

Sensors
LiDAR
Radar
Cameras
GPS
IMU
Encoders
Maps & Localization
Perception
Prediction
Actuators
Input
Destination
Routing
Motion
Planning
Vehicle
Control
Accelerator
Brakes
Steering
Signals

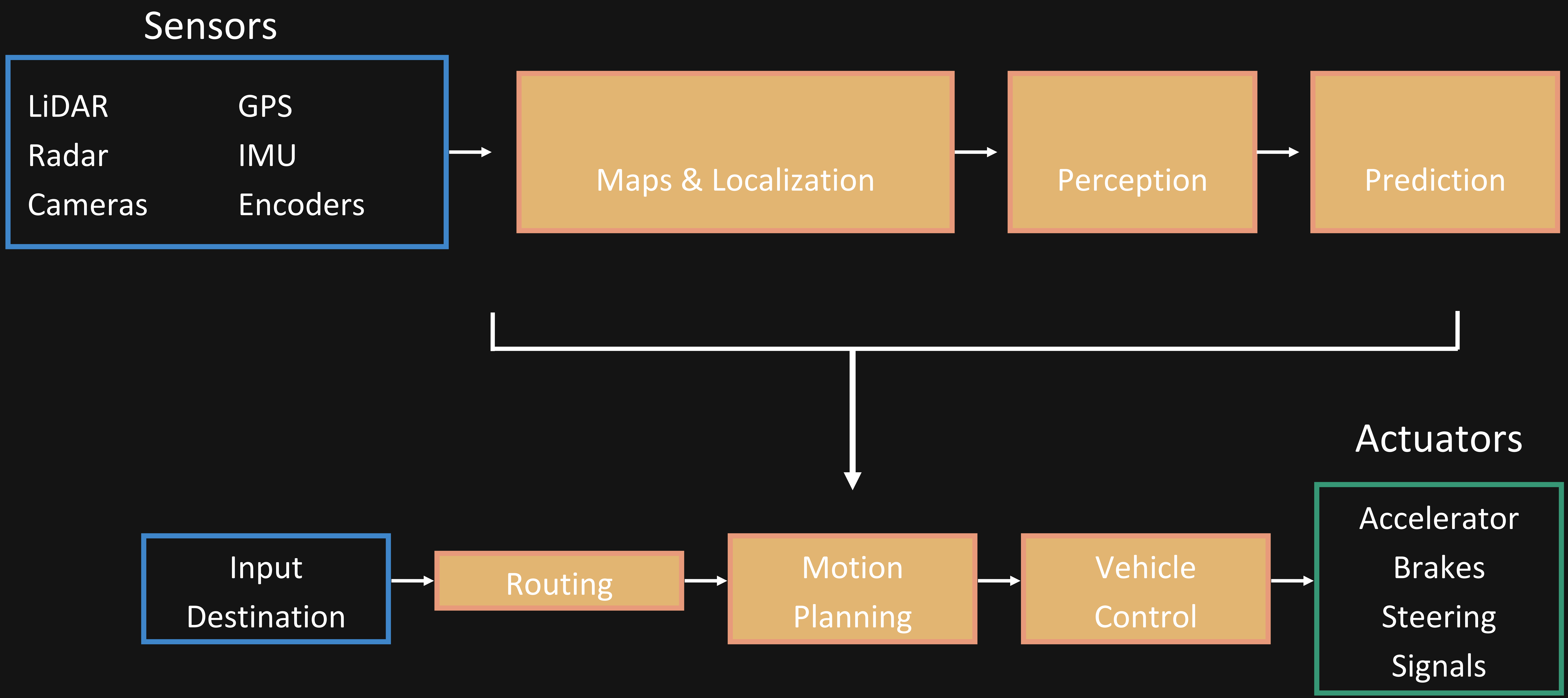

AUTONOMOUS
Angle
0.00 °
Speed
1.07 MPH
0.48 M/S
SOCR LIMIT
- MPH
Tempe Bicycle
(Rural & University)
E University Dr
S Rural Rd
mapbox


Sensors
LiDAR
Radar
Cameras
GPS
IMU
Encoders
Maps & Localization
Perception
Prediction
Actuators
Input
Destination
Routing
Motion
Planning
Vehicle
Control
Accelerator
Brakes
Steering
Signals

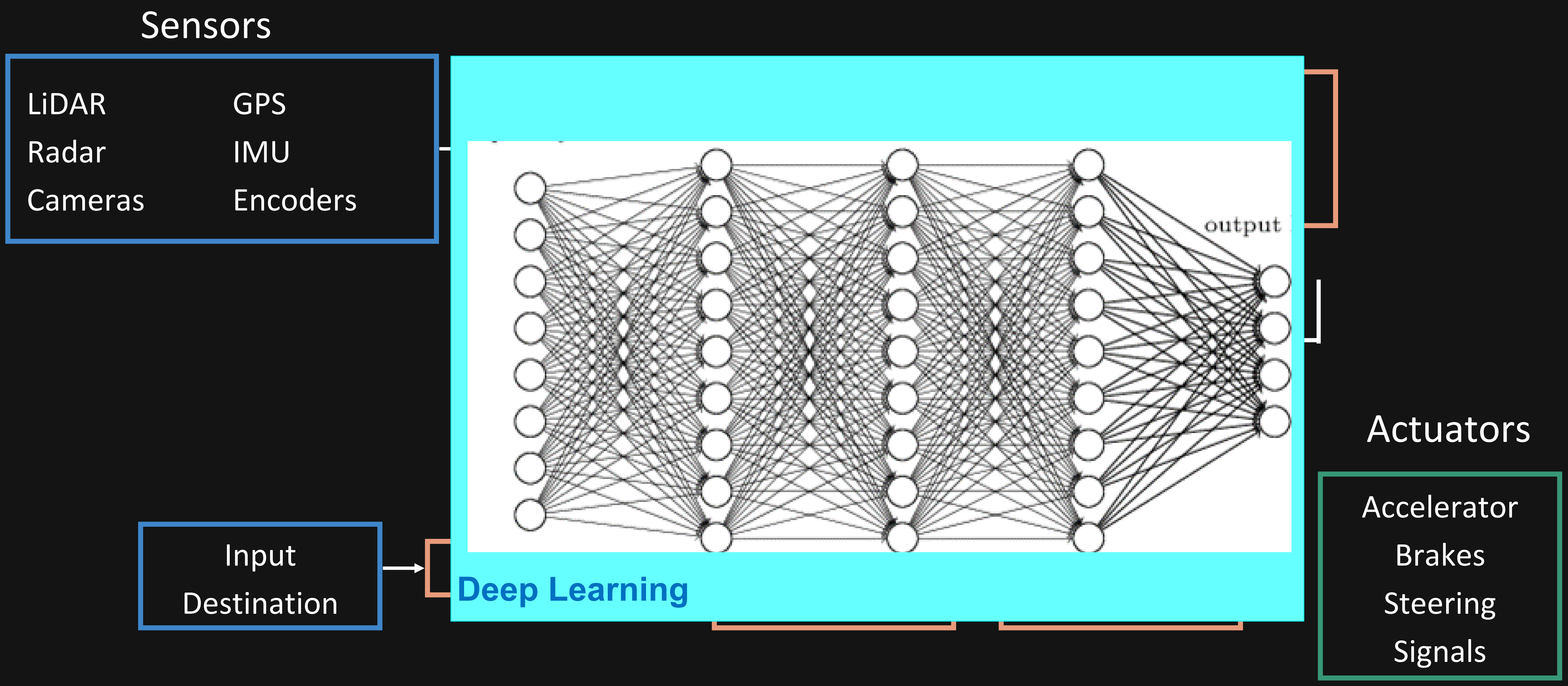


Planning and Model Predictive Control




Sensors
LiDAR
Radar
Cameras
GPS
IMU
Encoders
Maps & Localization
Perception
Prediction
Input
Destination
Routing
Motion
Planning
Vehicle
Control
Actuators
Accelerator
Brakes
Steering
Signals

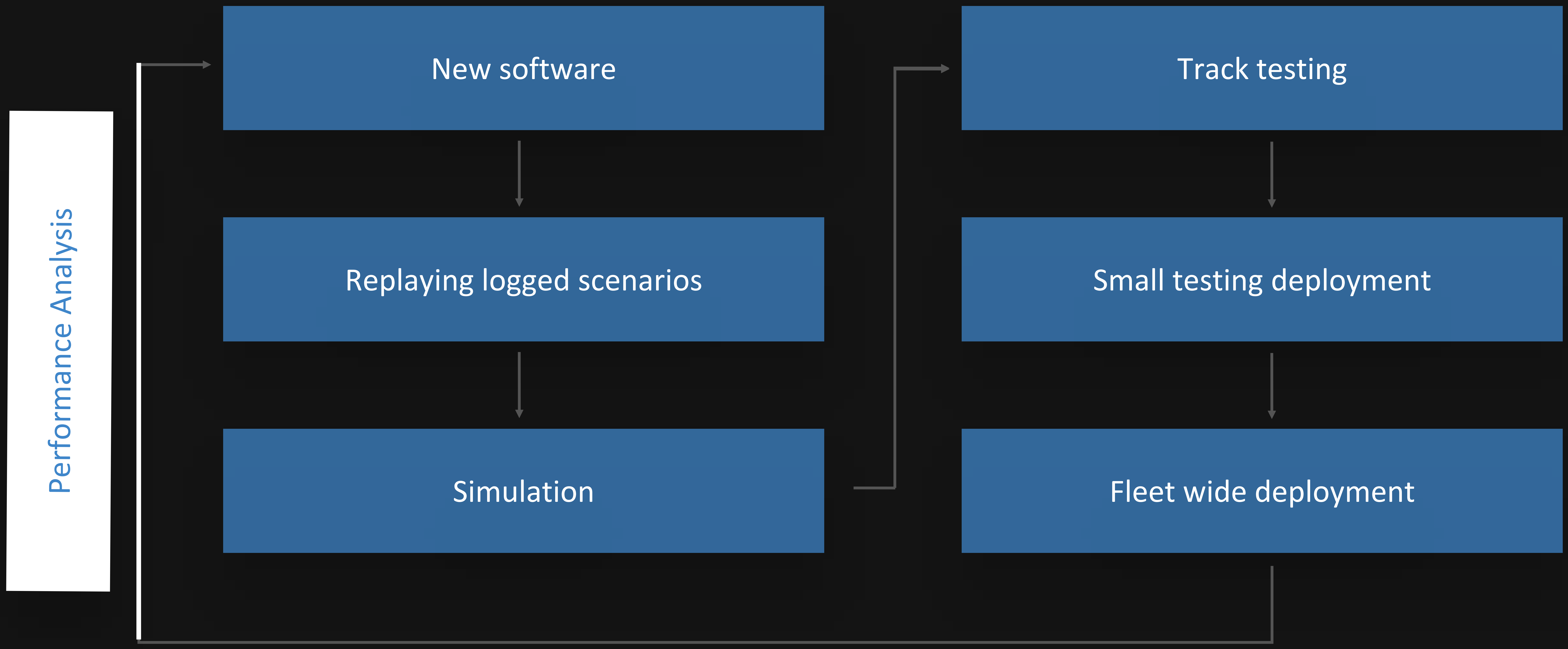
Imitation Learning



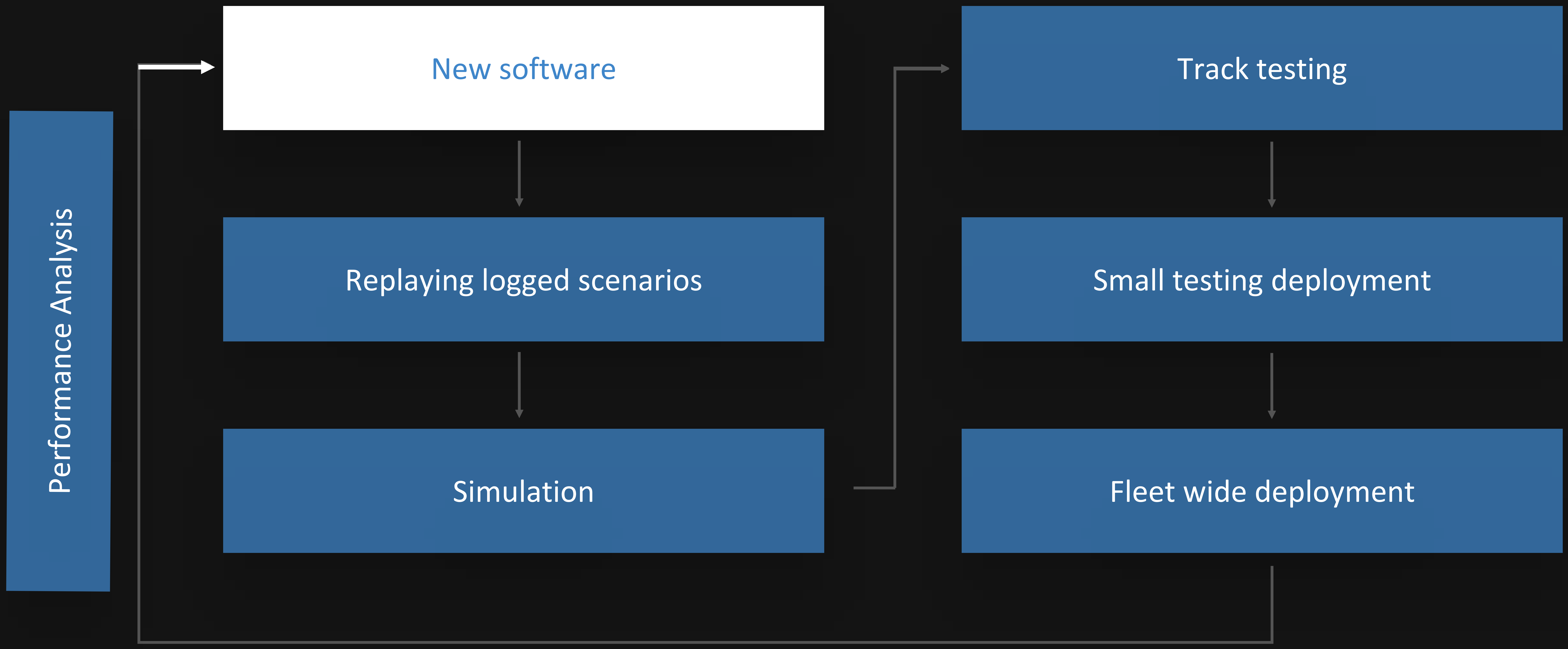

Current diagnostics
ENGINE
START
STOP

Autonomy Software

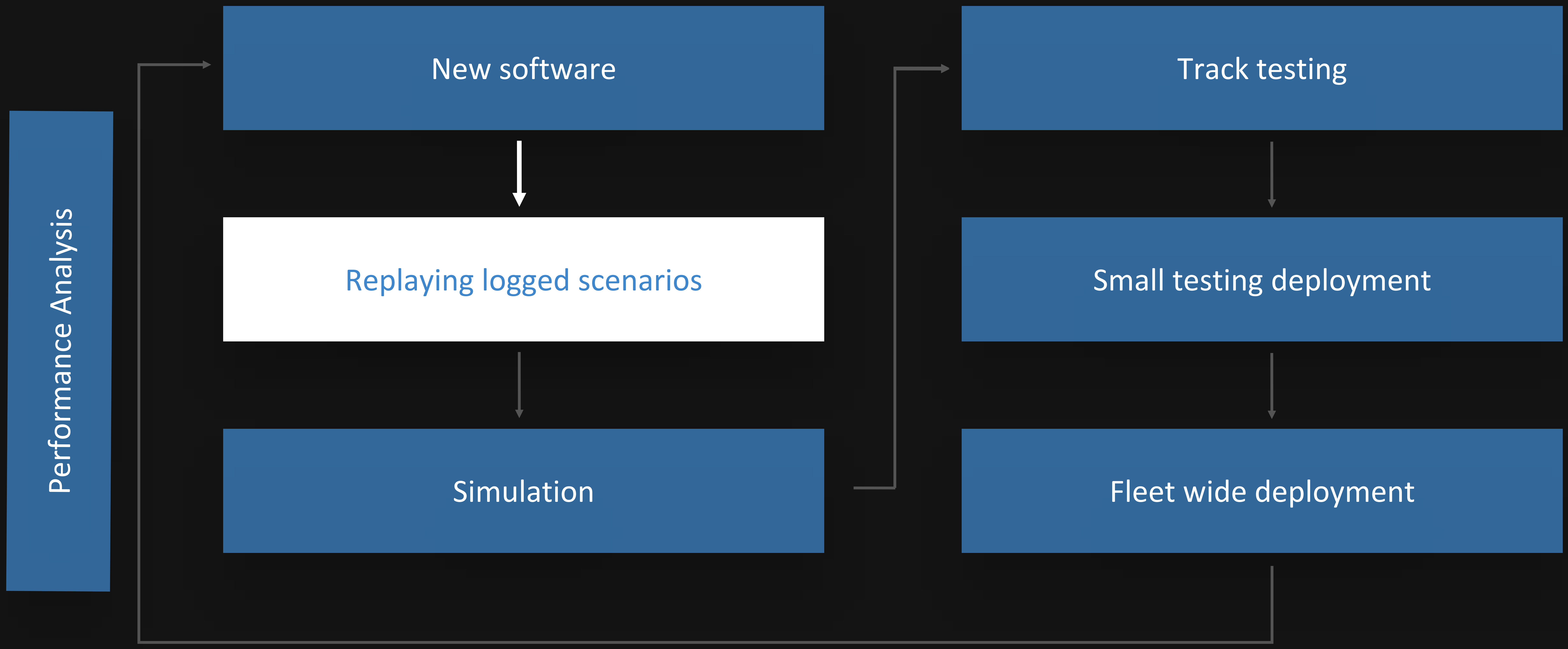
Development and Testing Process

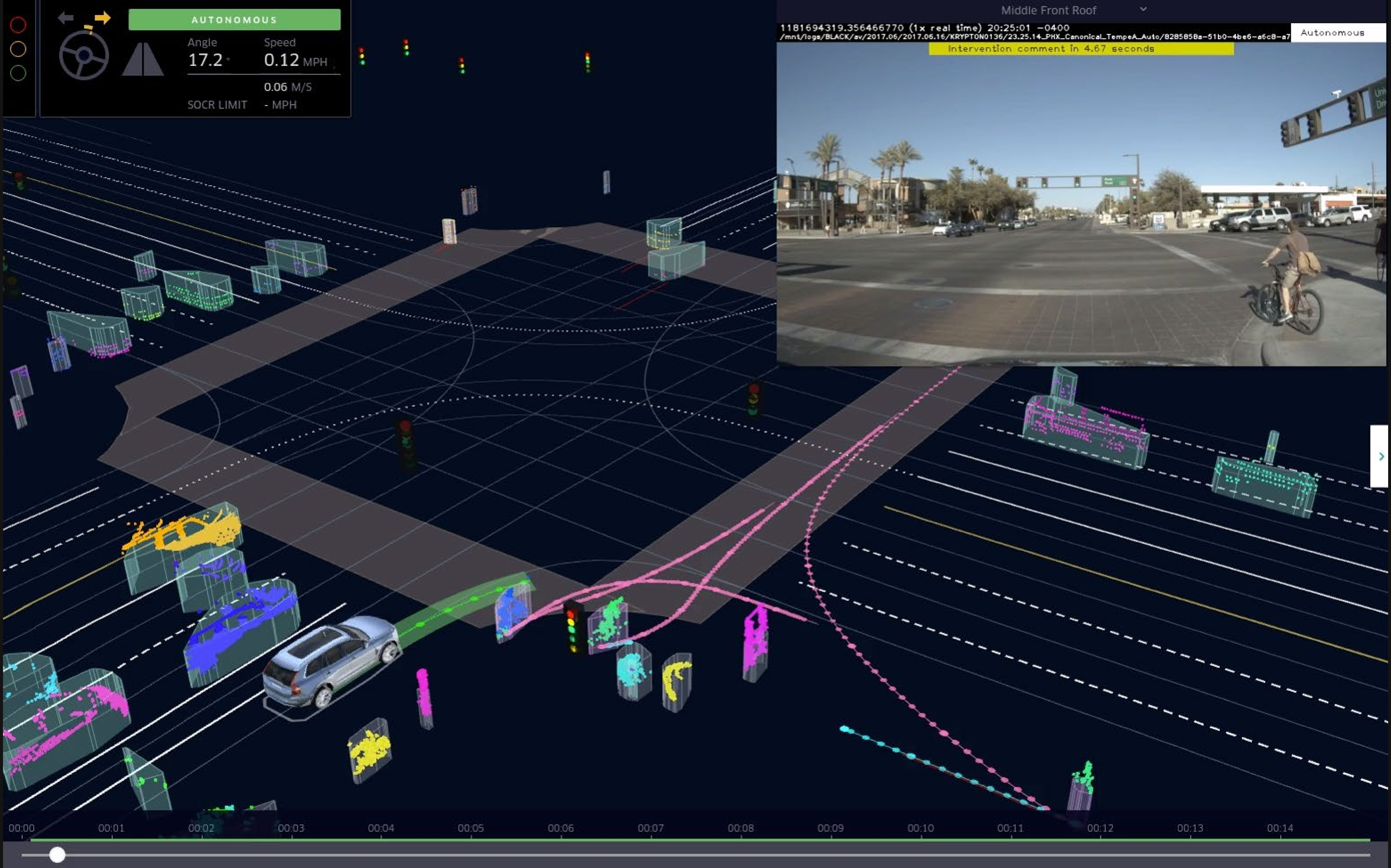

Performance Analysis
New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment

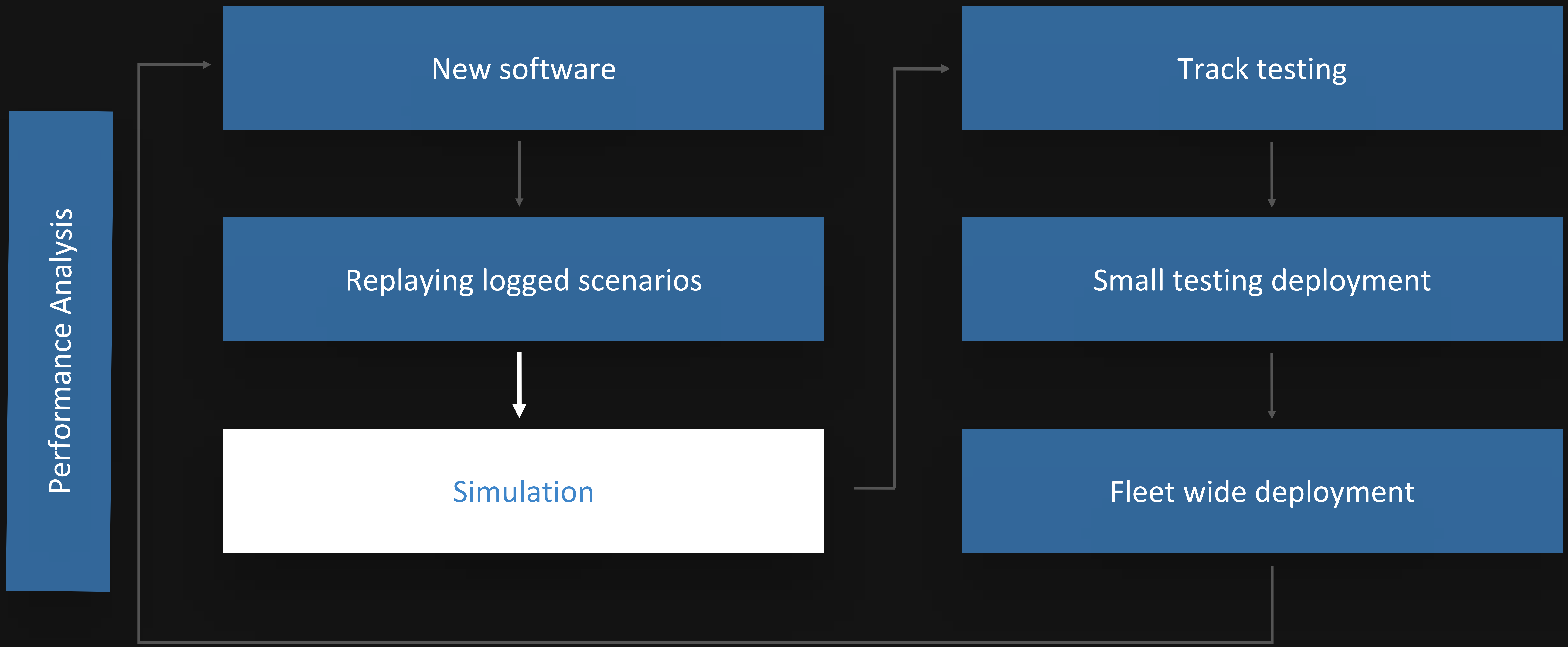
SEXTANT
Sidebar
Table
Legend
Date Range
Thu Feb 8th
Fri Mar 9th
Daily Time (EST)
0:00
23:00
Blueprint
Platform
Release
Map Version
Events
Washington's Landing
LOWER LAWRENCEVILLE
SPRING HILL
EAST ALLEGHENY
STRIP DISTRICT
POLISH HILL
BEDFORD DWELLINGS
HILL DISTRICT
CRAWFORD-ROBERTS
TERRACE VILLAGE
WEST OAKLAND
Pittsburgh
Riverfront Landing
University of Pittsburgh
Penn Ave
Liberty Ave
Smallman St
River Ave
Bedford Ave
Webster Ave
Centre Ave
Bigelow Blvd
31st St Bridge
Troy Hill Rd
Fifth Ave
Grant St

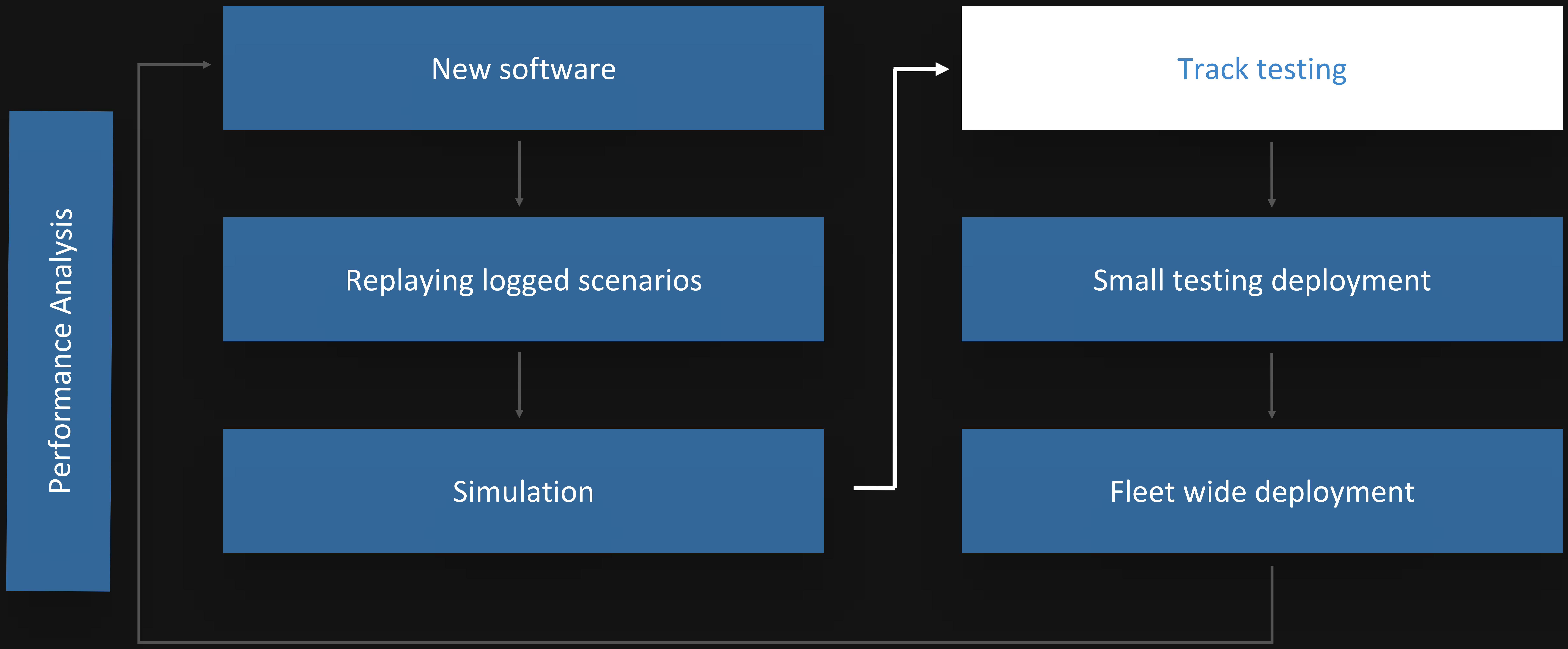


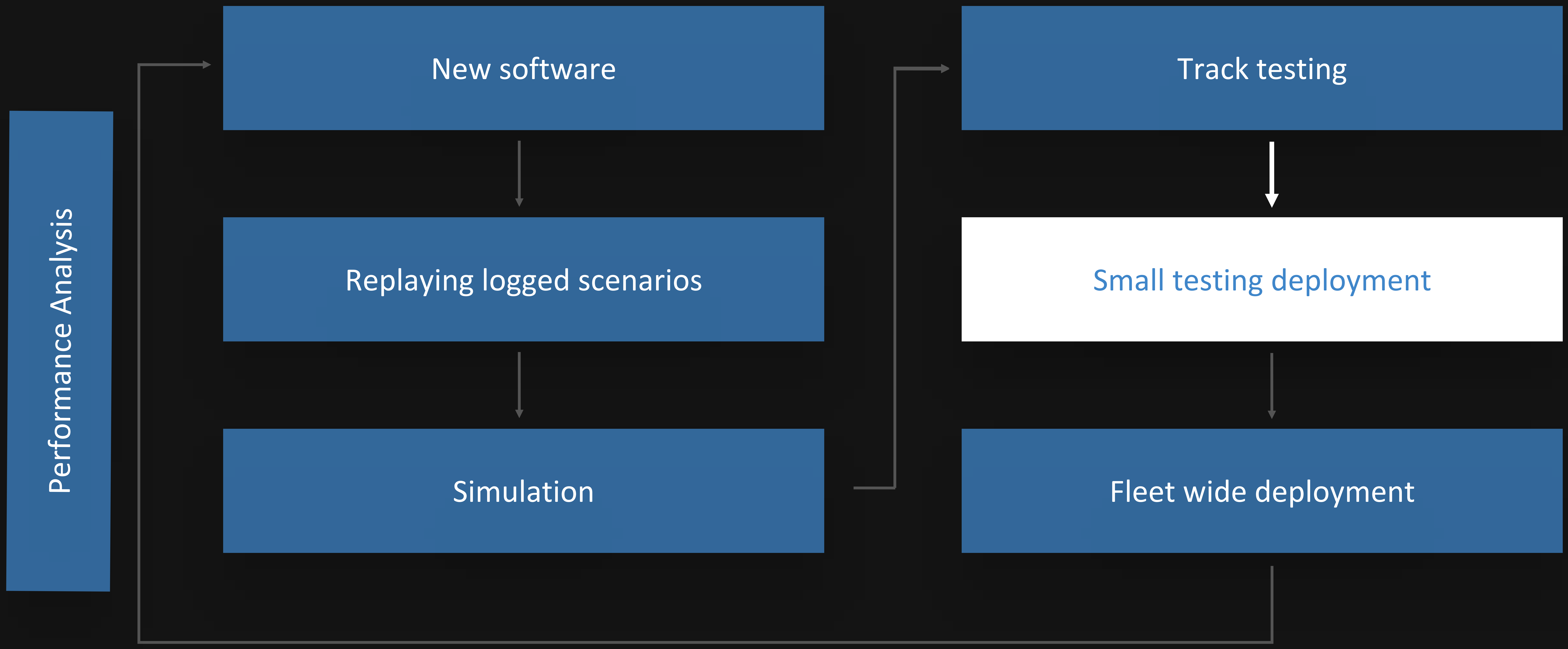
New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment
Performance Analysis

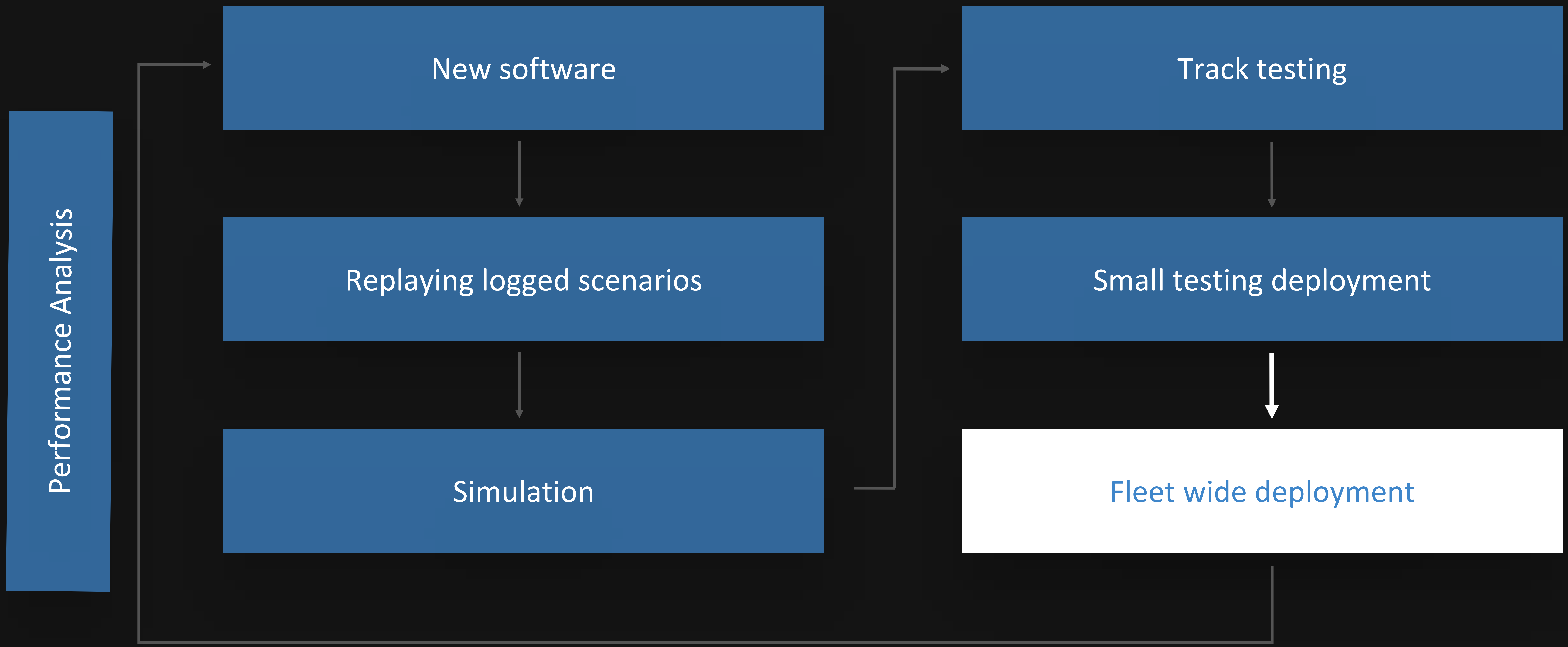

Performance Analysis
New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment


AUTONOMOUS
Angle
17.2 °
Speed
0.12 MPH
0.06 M/S
SOCR LIMIT
- MPH
Middle Front Roof
1181694319.356466770 (1x real time) 20:25:01 -0400
/mnt/logs/BLACK/av/2017.06/2017.06.16/KRYPTON0136/23.25.14_PHX_Canonical_TempeA_Auto/8285858a-51b0-4be6-a6c8-a7
Autonomous
Intervention comment in 4.67 seconds
00:00
00:01
00:02
00:03
00:04
00:05
00:06
00:07
00:08
00:09
00:10
00:11
00:12
00:13
00:14


Performance Analysis
New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment


New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment
Performance Analysis

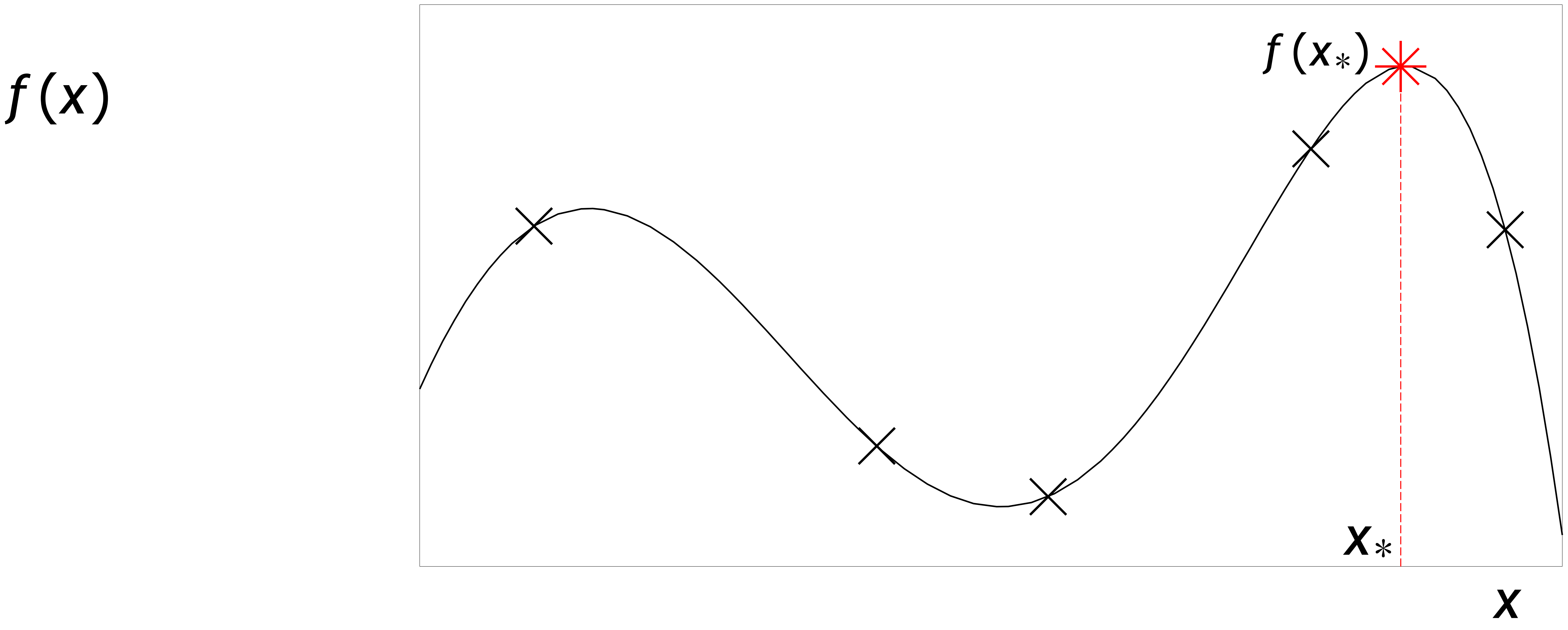

Performance Analysis
New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment


Performance Analysis
New software
Replaying logged scenarios
Simulation
Track testing
Small testing deployment
Fleet wide deployment


1204386618.111423015 (3x real time) 10:50:00 -0500
/mnt/logs/ROSE/av/2018.03/2018.03.06/KRYPTON0083/15.38.31_T205148/TransmissionTimeIndex00000.hlog
Autonomous
5465
ZERO EMISSIONS VEHICLE
STOP BACK 5 FEET

Active Optimization

f is an unknown expensive black-box function.
Let $\mathbf{x}_* = \mathrm{argmax}_x f(x)$.
Goal: approximately optimize f with as few experiments as possible

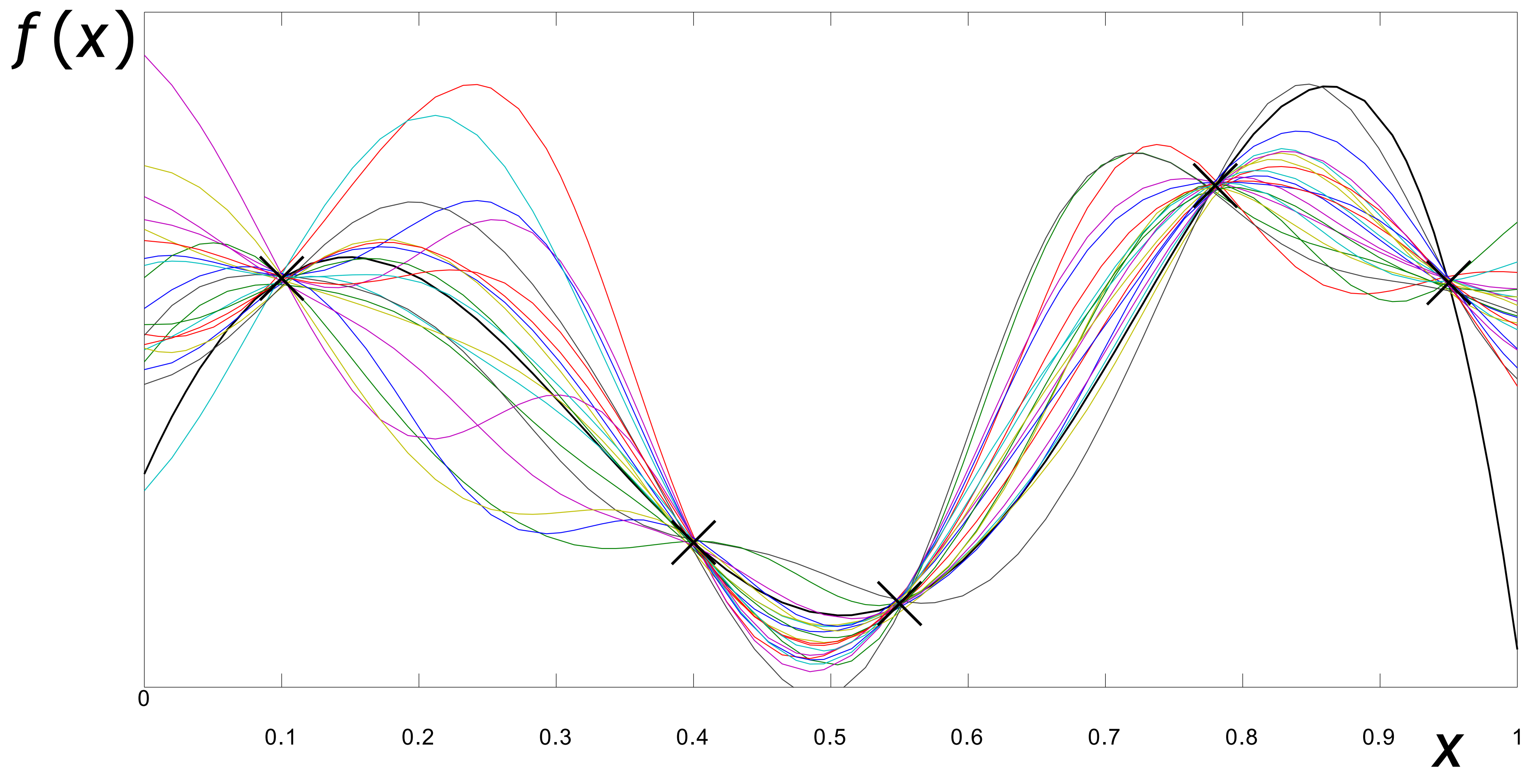


Optimizing expensive to evaluate functions

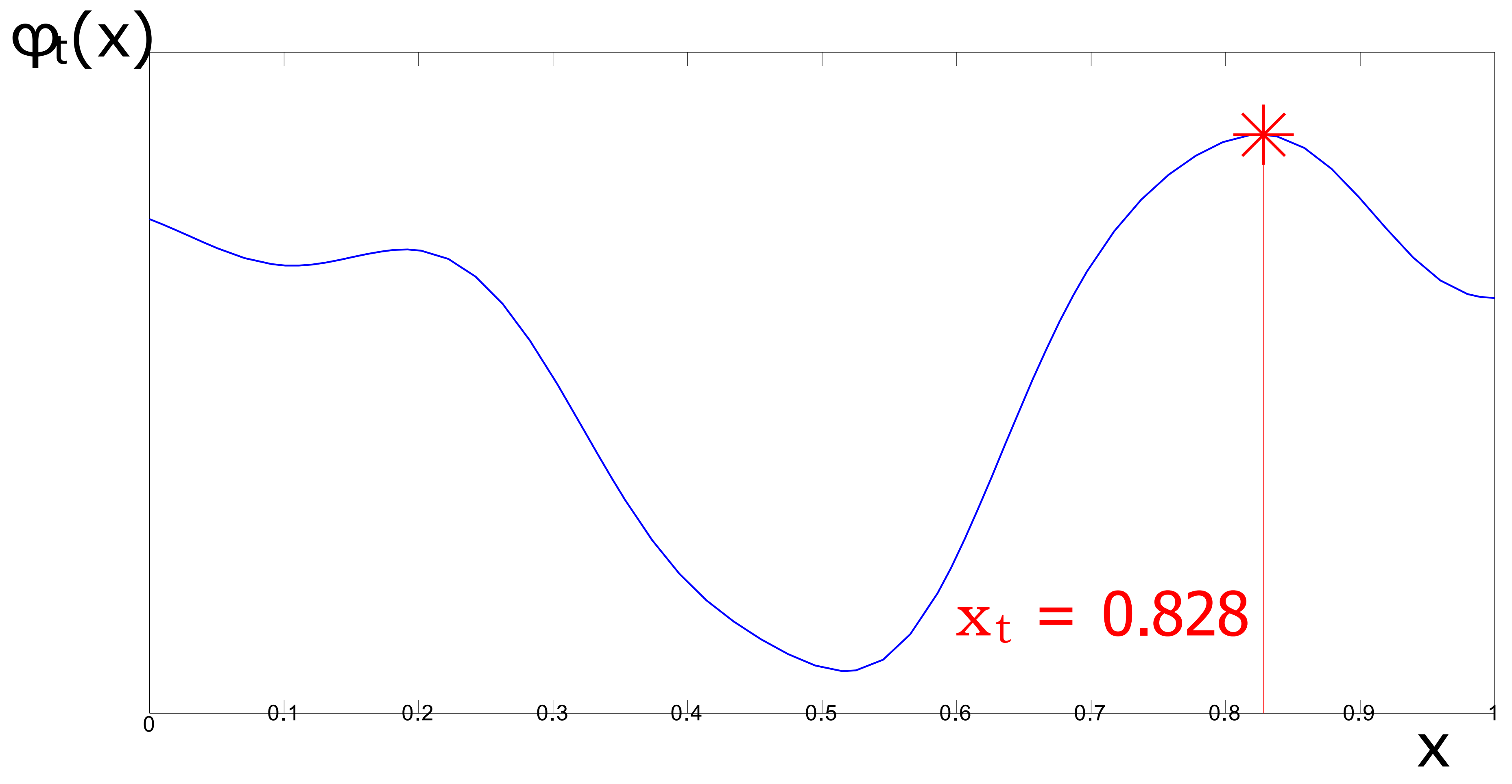
- Tuning the hyperparameters of supervised learning algorithms
  - e.g. deep networks
- Systems requiring physical experiments (online/onboard optimization)
- Algorithms that are tested via expensive simulations
  - compute stack performance
  - planner/controller parameters
  - scientific model fitting

Active Optimization Algorithm

Model f as a sample from a Gaussian Process.

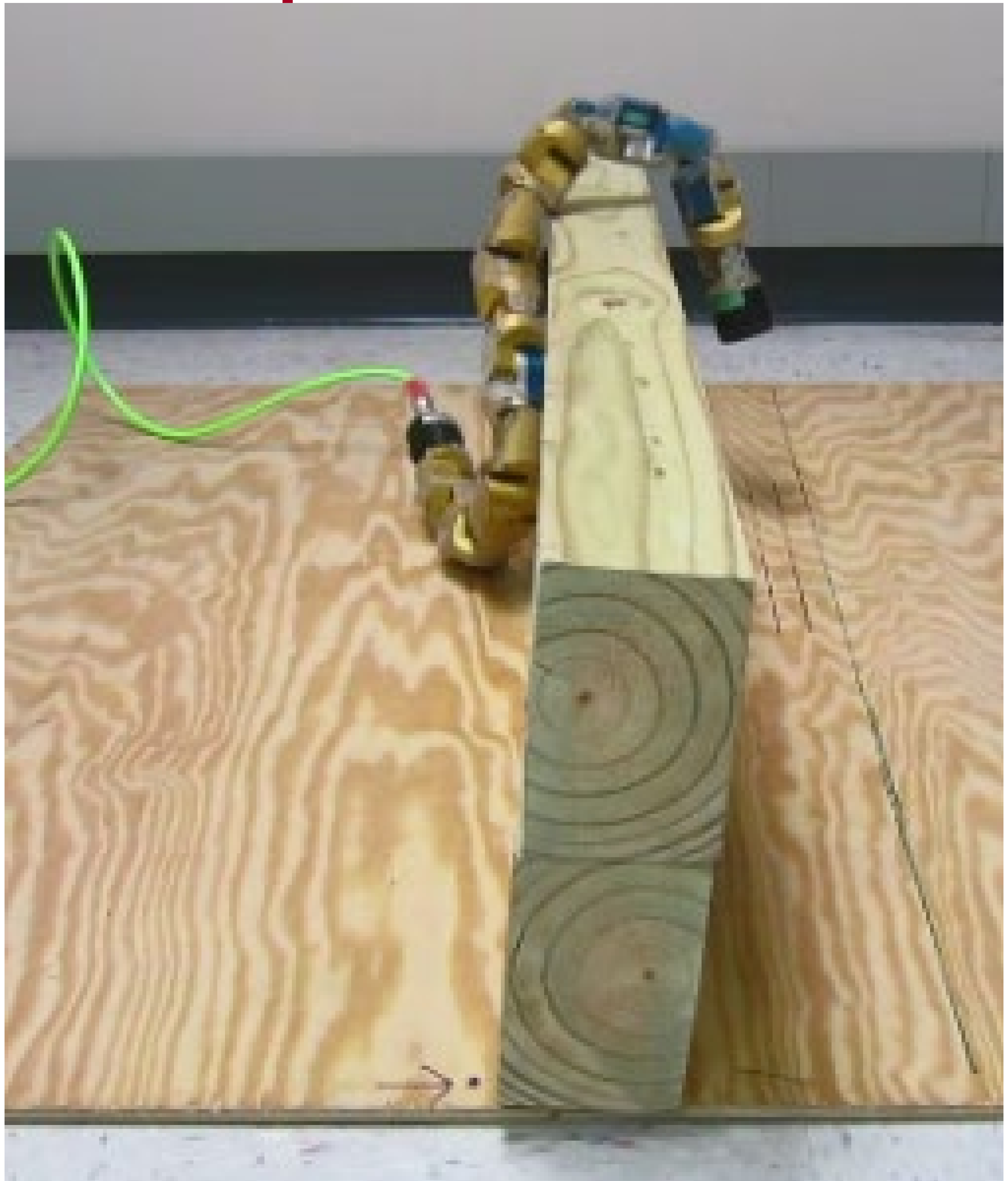


Maximise acquisition function ϕ_t: $\mathbf{x}_t = \mathrm{argmax}_x \, \phi_t(x)$.



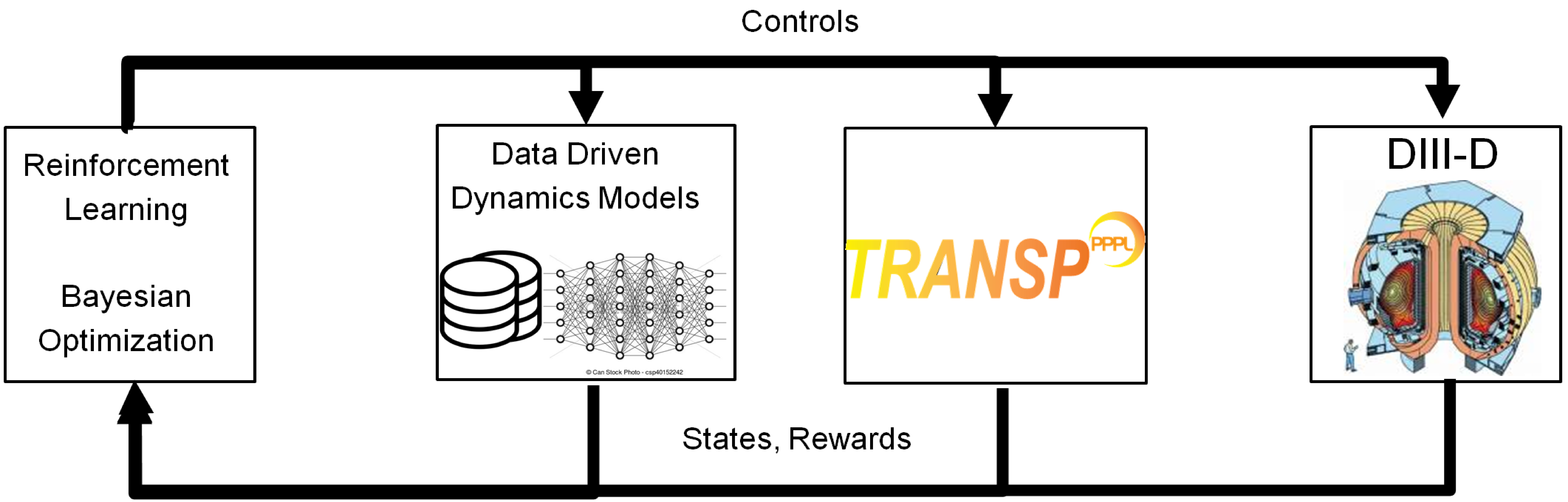
1. Learn model from data you have (including uncertainty)
2. Search the model for the best experiment.
3. Run the chosen experiment and collect a new data point.
4. While experiment budget is not exhausted, repeat.

Active Optimization Trials



Controlling Fusion Plasmas

Nuclear Fusion and Machine Learning

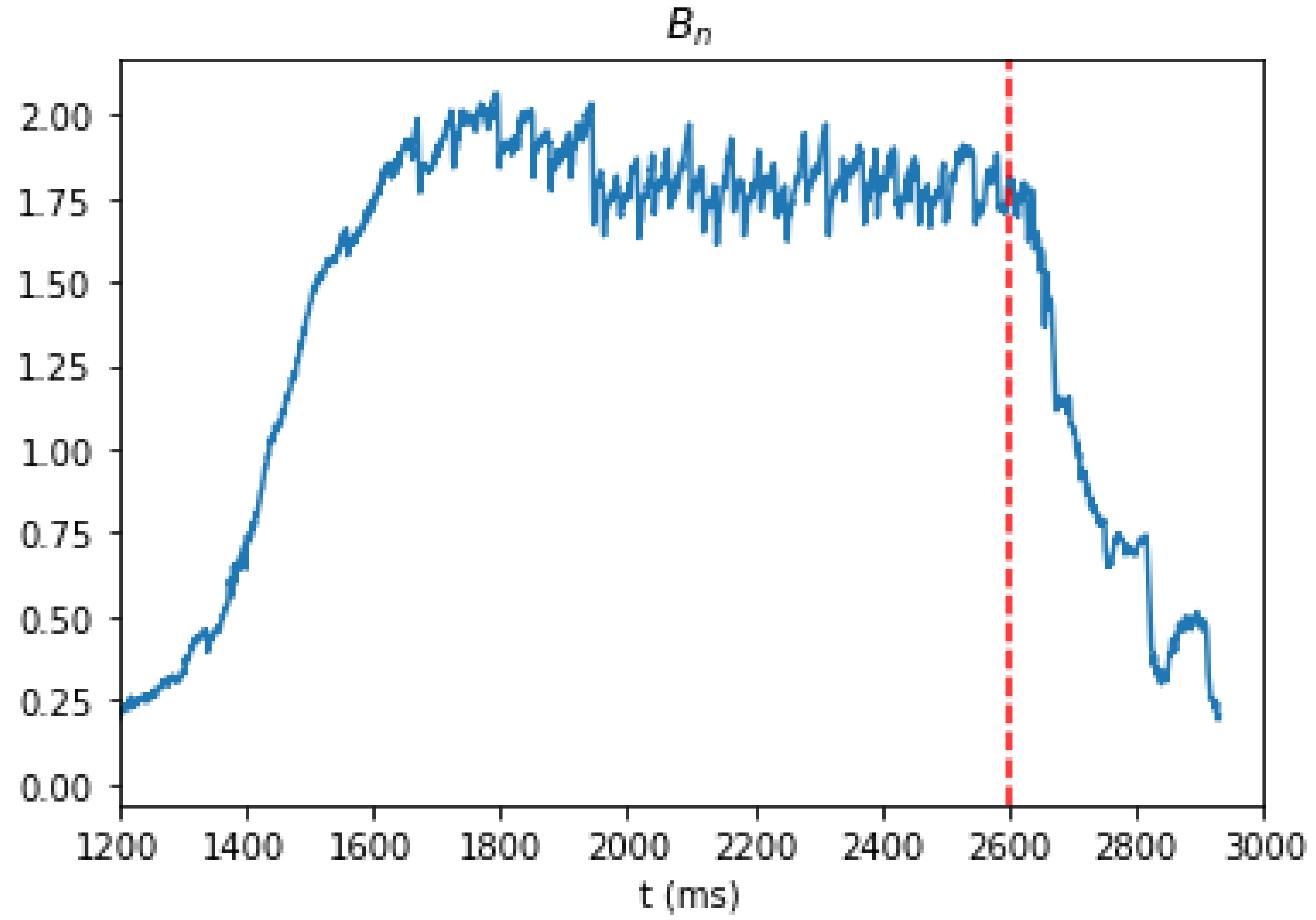


- Bayesian optimization for response to instabilities
- Improved models combining data and first principles
- Beta_N and Rotation tracking with reinforcement learning

Carnegie Mellon
Auton Lab

Bayesian Optimization with TRANSP for response to instabilities

- **Signals**
  - Use B_n to measure pressure.
  - Use δW as a proxy for stability.

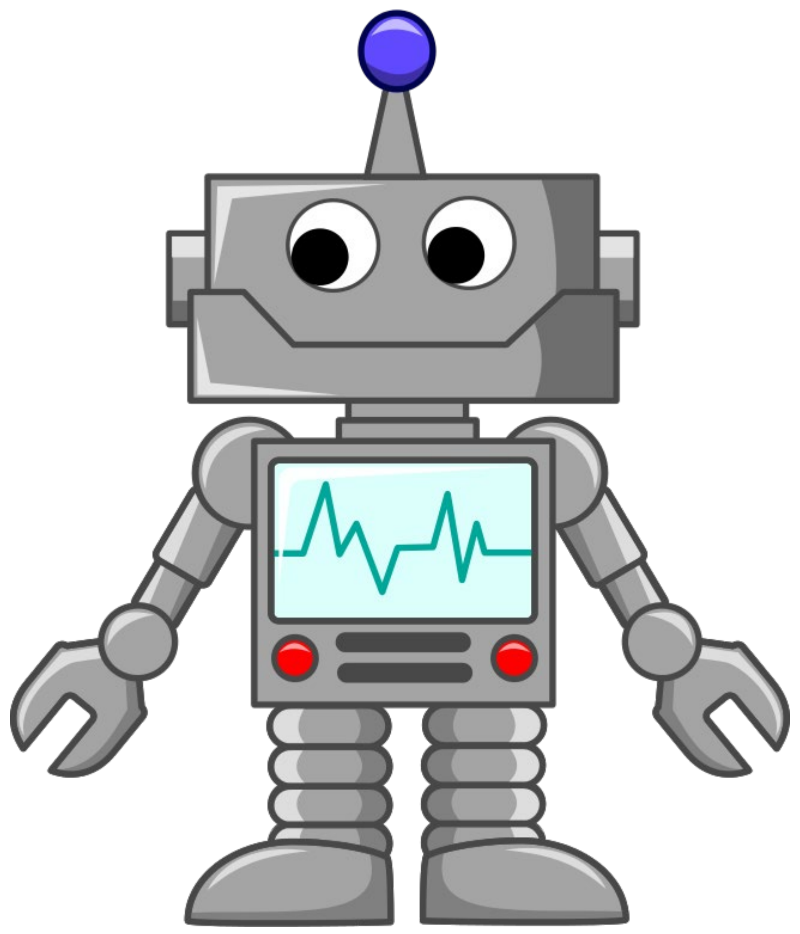


B_n, δW → Controller → Co-current Beam Power, Counter-current Beam Power

- Use TRANSP as a simulator to test effect of beam powers.
  - Start 150ms before tearing mode and run until 150ms after.

- **Score to maximize:** $B_n + C\delta W$

We can learn this controller by repeatedly querying TRANSP.



Contextual Bayesian
Optimization Algorithm

Request state of plasma
and beam powers

Receive future plasma
pressure and stability

Simulator
(TRANSP)

30 min to
simulate 150ms

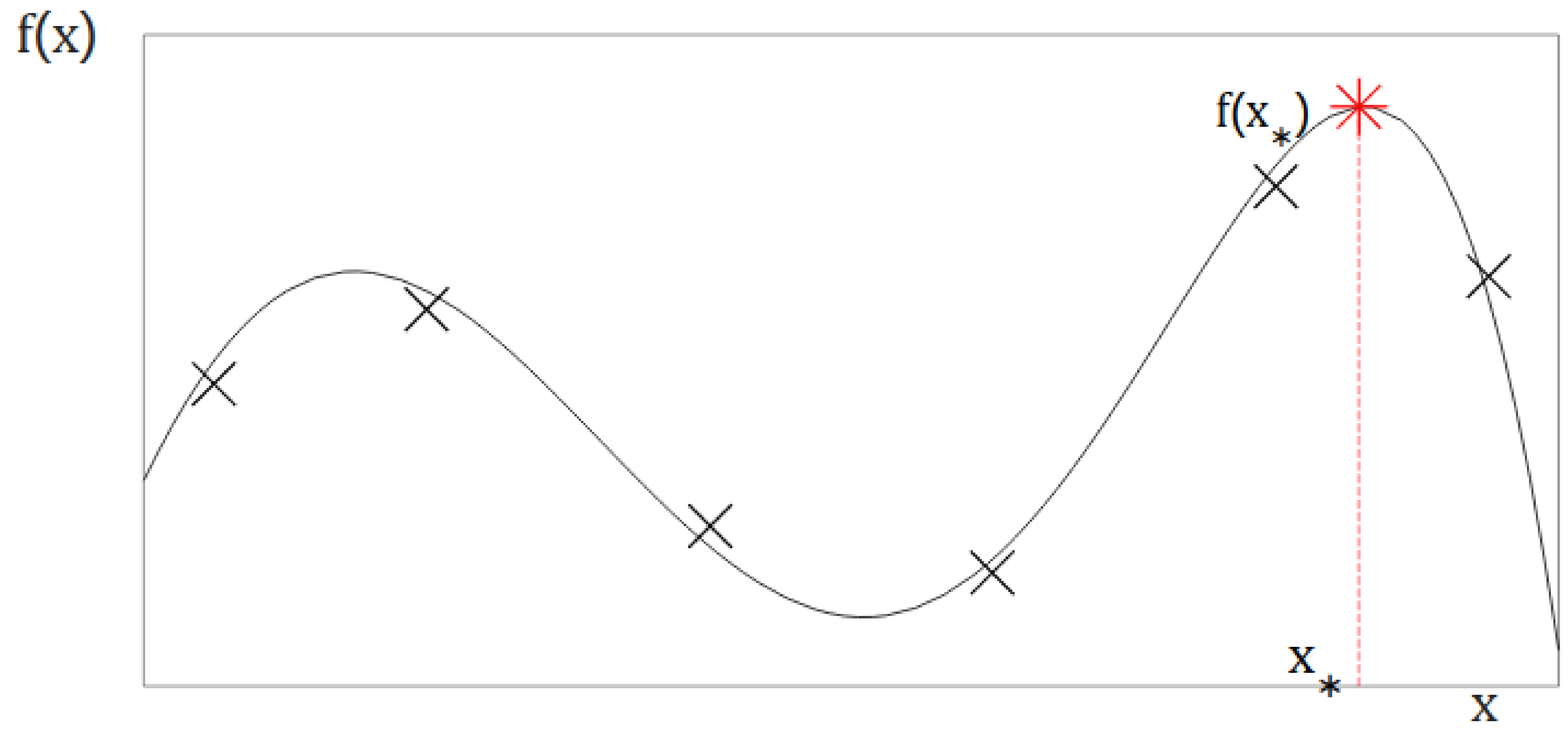
Carnegie Mellon

Auton Lab

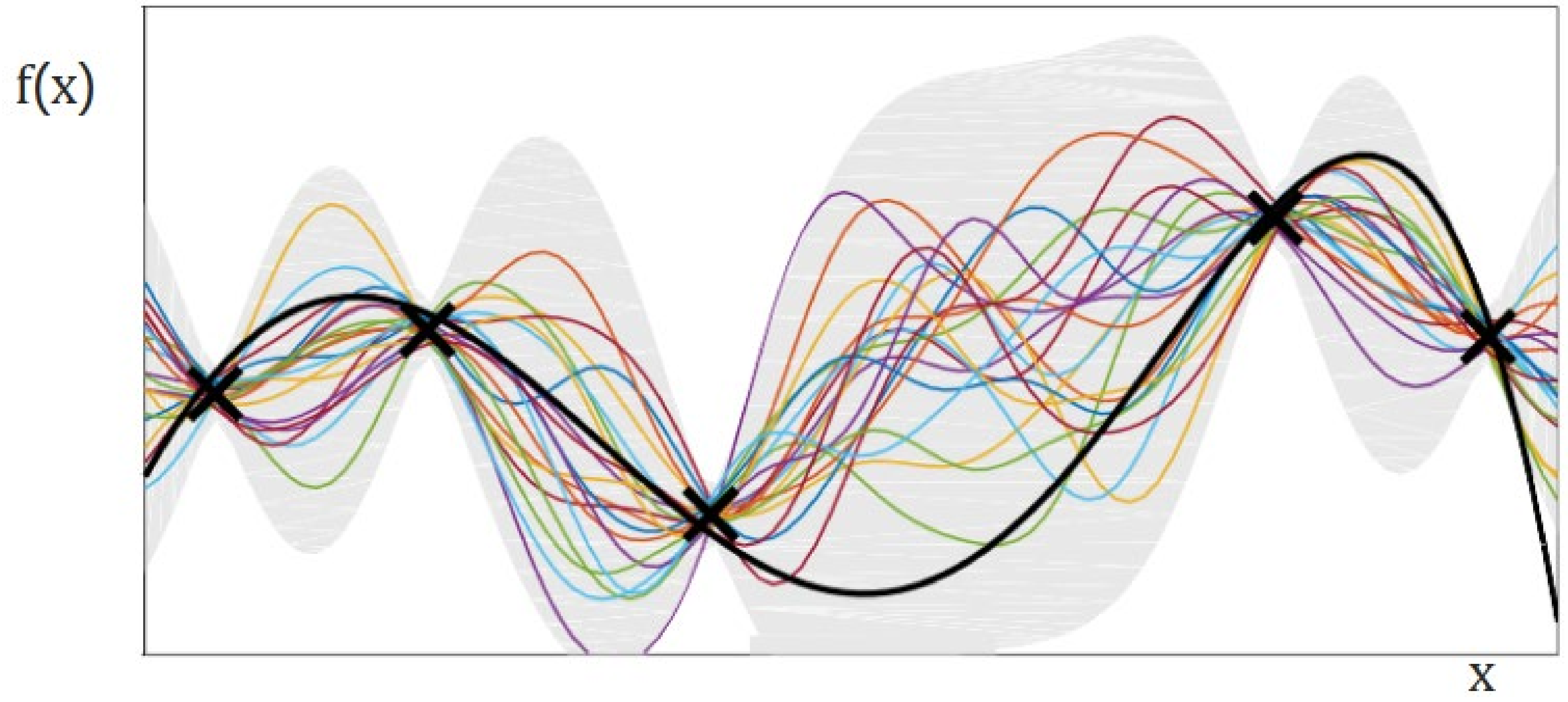
Bayesian Optimization

$f : X \rightarrow \mathbb{R}$ is an expensive black-box function, accessible only via noisy evaluations.

Let $x_* = \text{argmax}_x f(x)$.

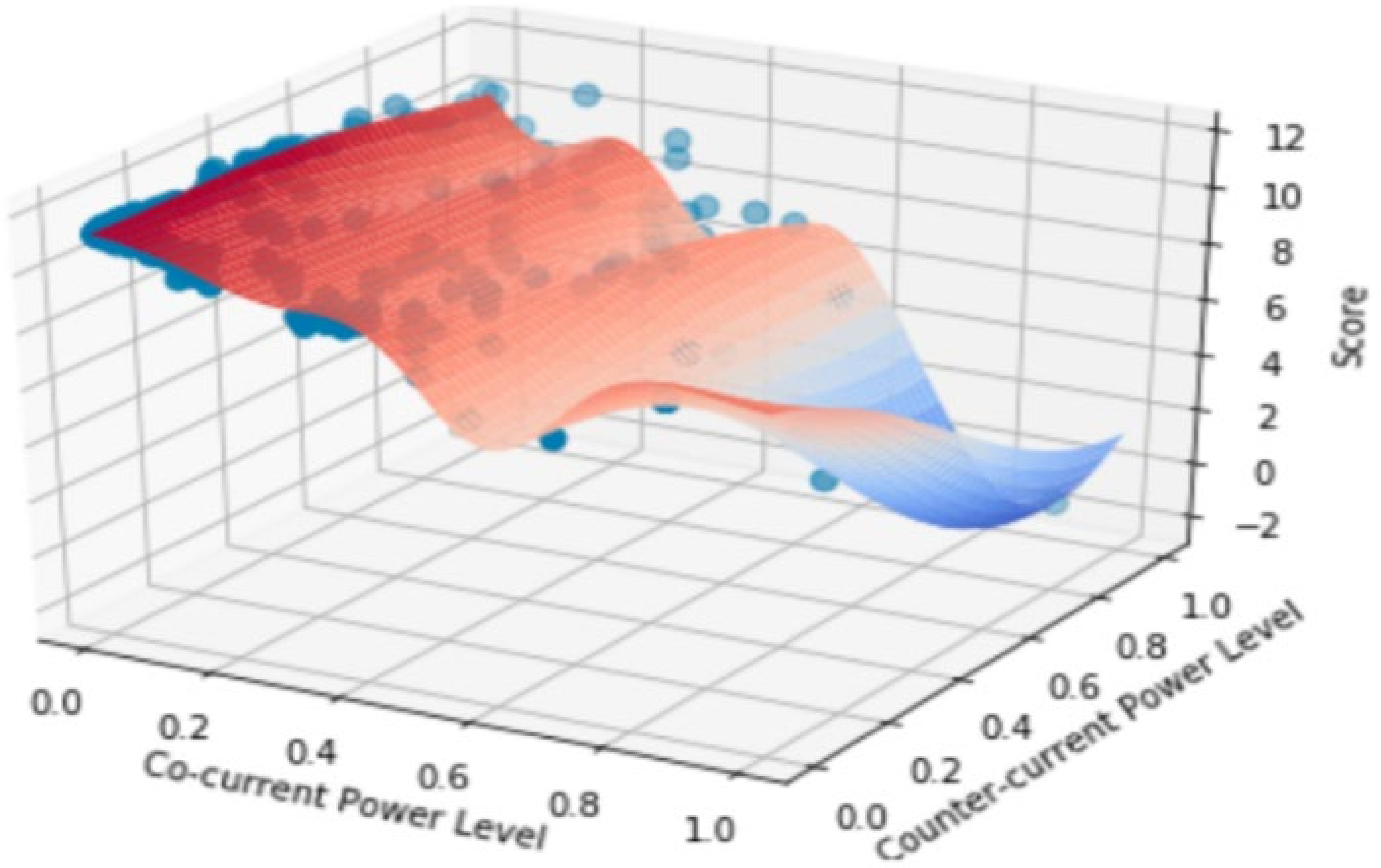


Model of the Function



- For our application...
  - The expensive function f is TRANSP and returns pressure+stability score.
  - x_* is the best possible setting for beam powers.

Offline Contextual Bayesian Optimization to Learn a Controller



Have many of these optimization landscapes, one for each state of plasma.

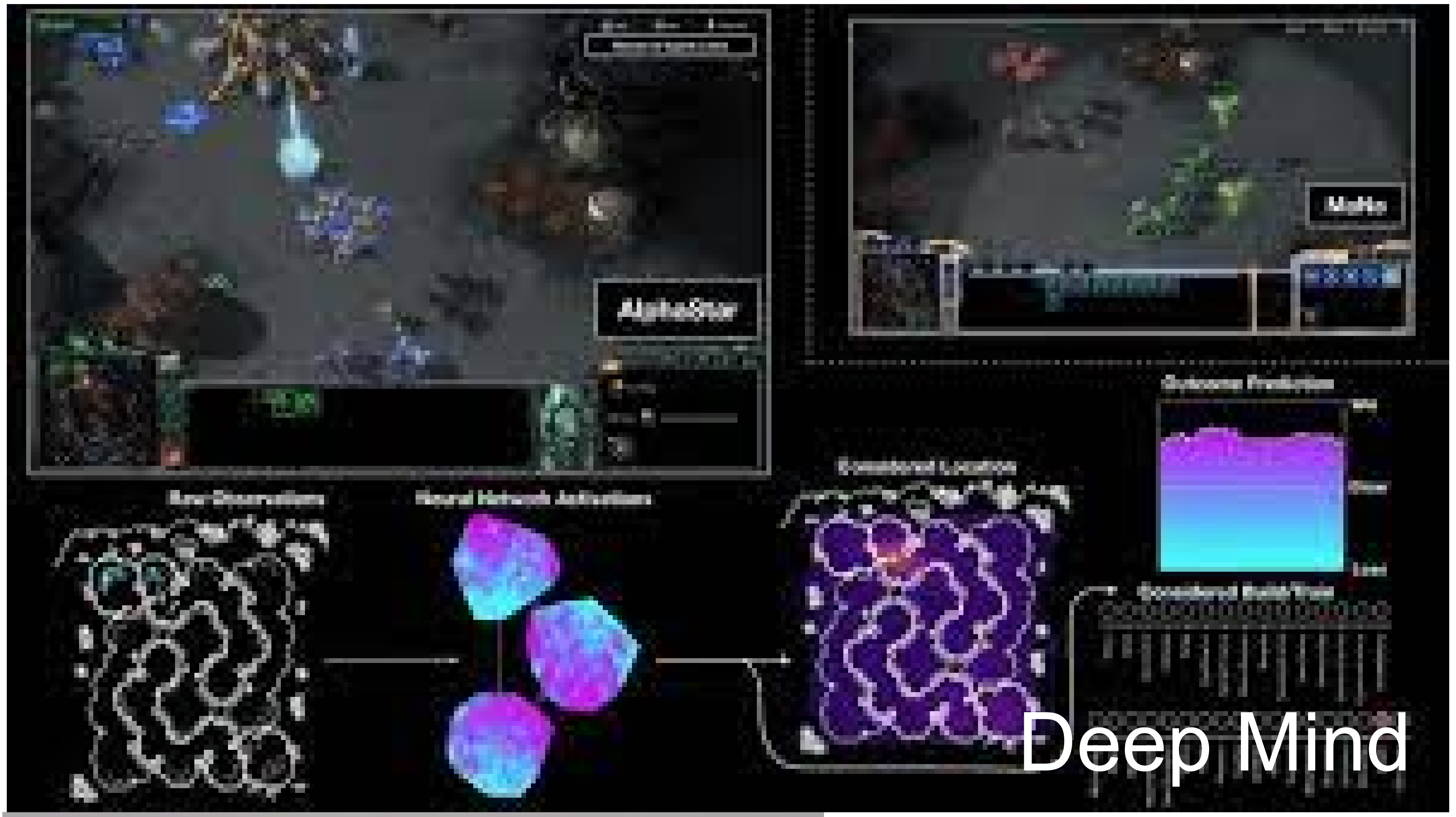
- Our algorithm efficiently picks which state of plasma to optimize for.
- This algorithm learns the best controller much faster than traditional Bayesian Optimization algorithms.
- Paper accepted at NeurIPS, a top machine learning conference

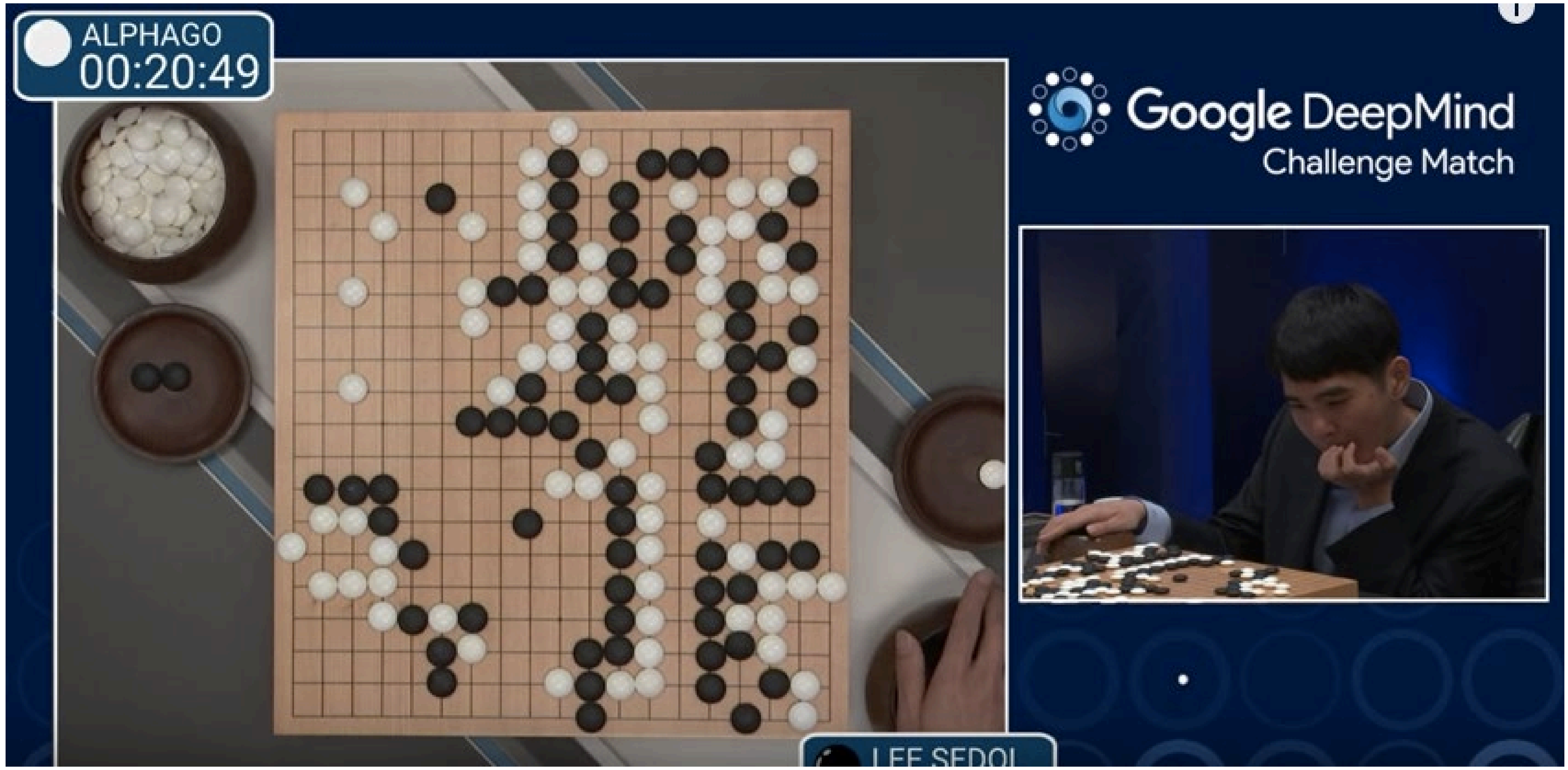
Char, I., Chung, Y., Neiswanger, W., Kandasamy, K., Nelson, A. O., Boyer, M., Koleman, E., Schneider, J., "Offline contextual bayesian optimization", *Advances in Neural Information Processing Systems*, 2019.

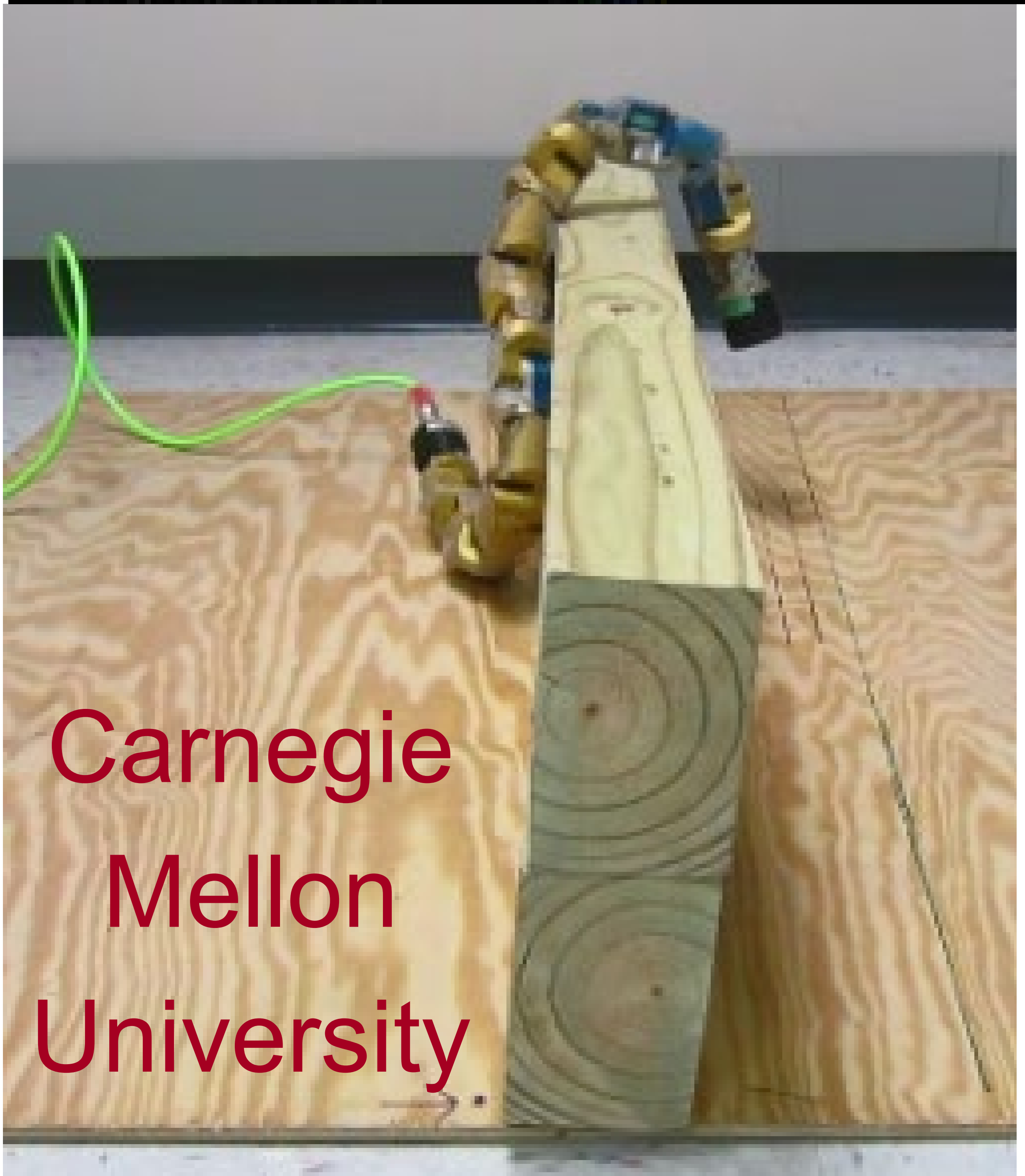
Carnegie Mellon

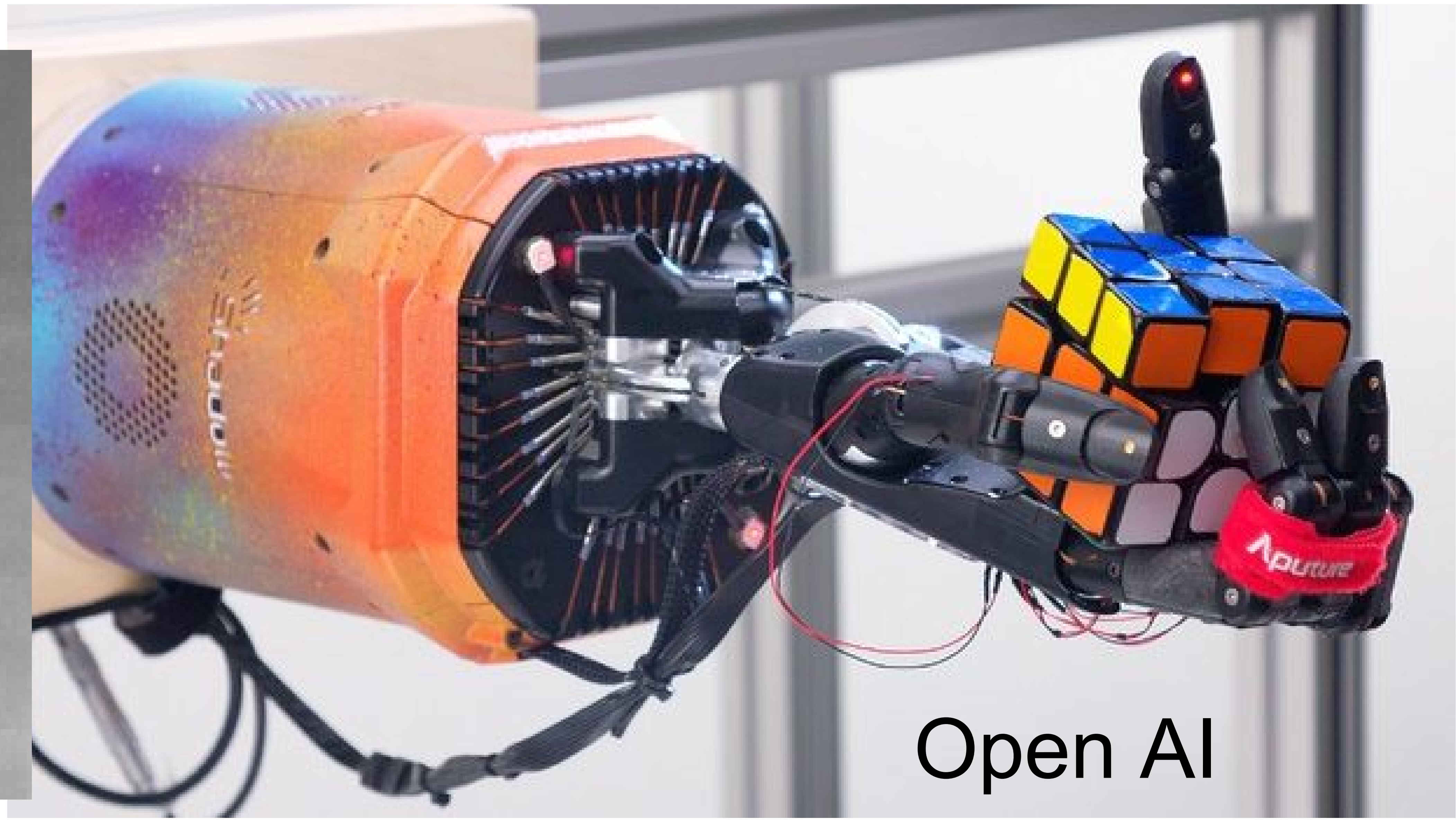
Auton Lab

Reinforcement Learning and Bayesian Optimization Successes





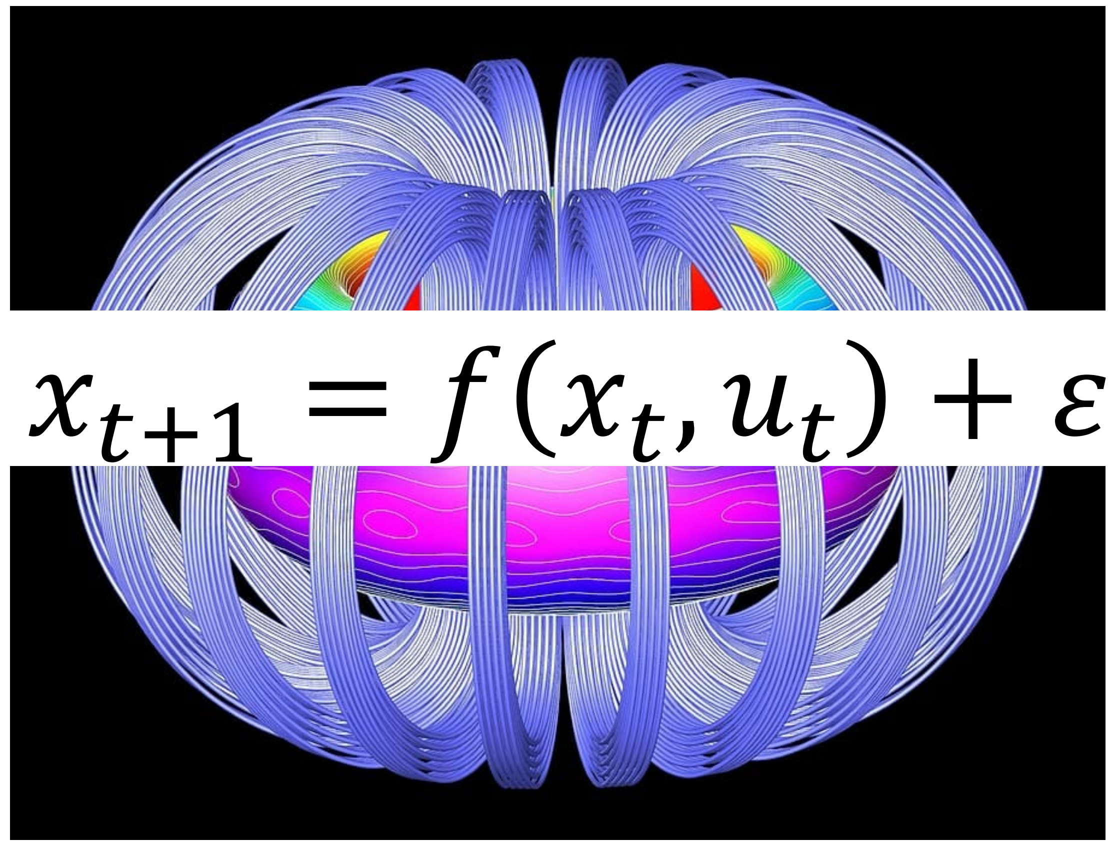




Building a Model

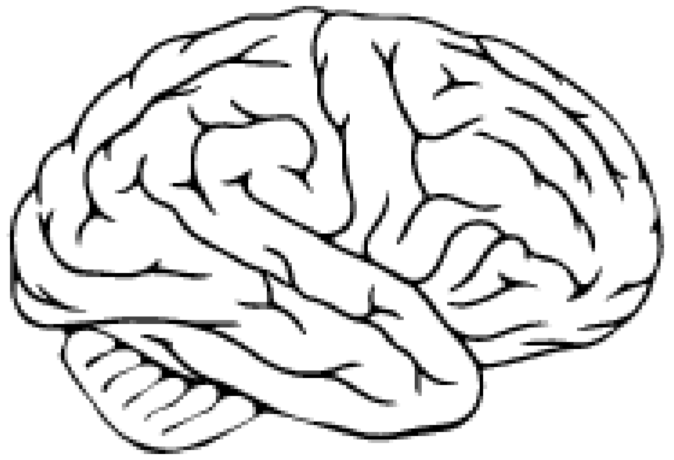
State x_t, e.g. β_N

Control u_t, e.g. total power from neutral beams



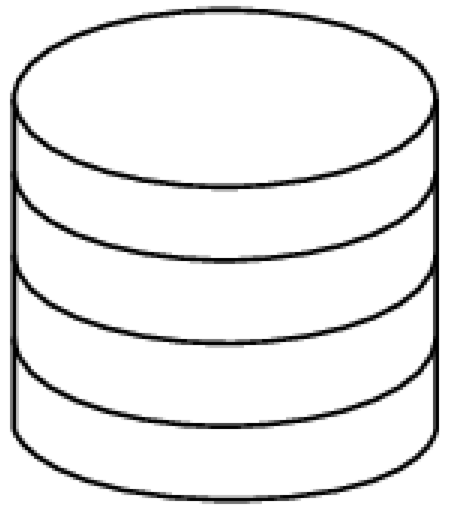
$$x_{t+1} = f(x_t, u_t) + \varepsilon$$

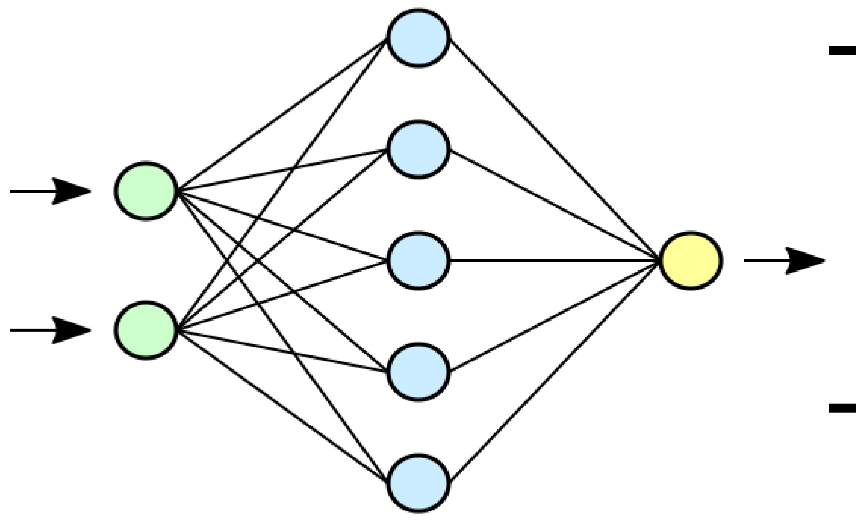
State, x_{t+1}, e.g. β_N



Scientific First Principles

- plus heuristic simplifications for tractability
- yields a simulation or equations useful for analysis and control





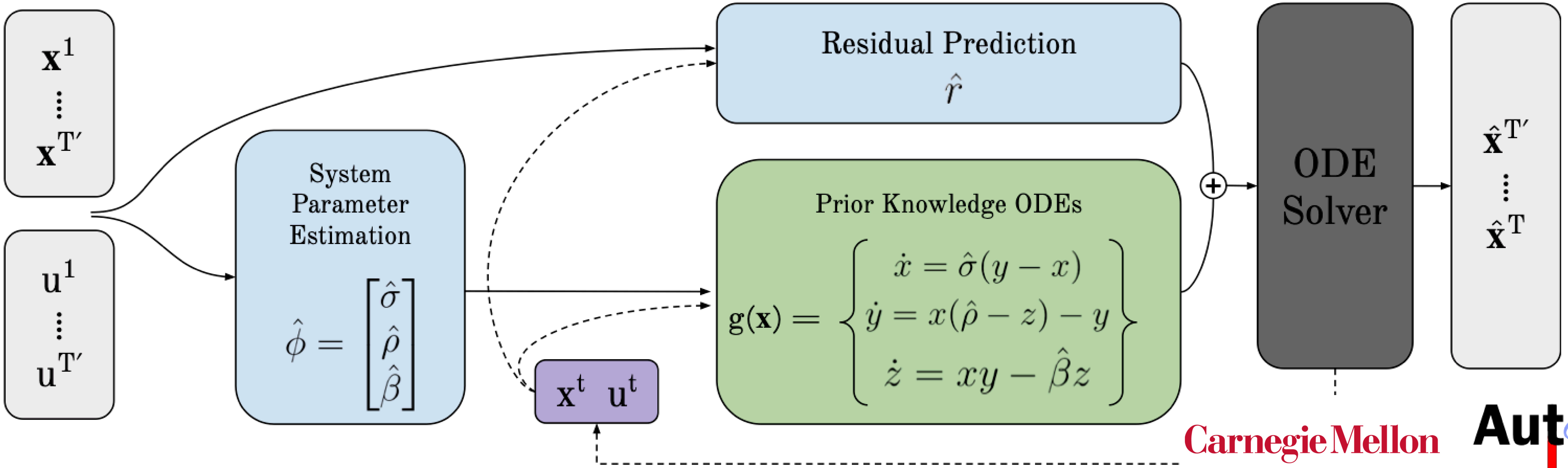
Data Driven Machine Learning

- collect data from real device or simulation
- train a model with supervised learning
- use the model like a simulator

BOTH!?

Can we combine physical knowledge with data-driven modeling?

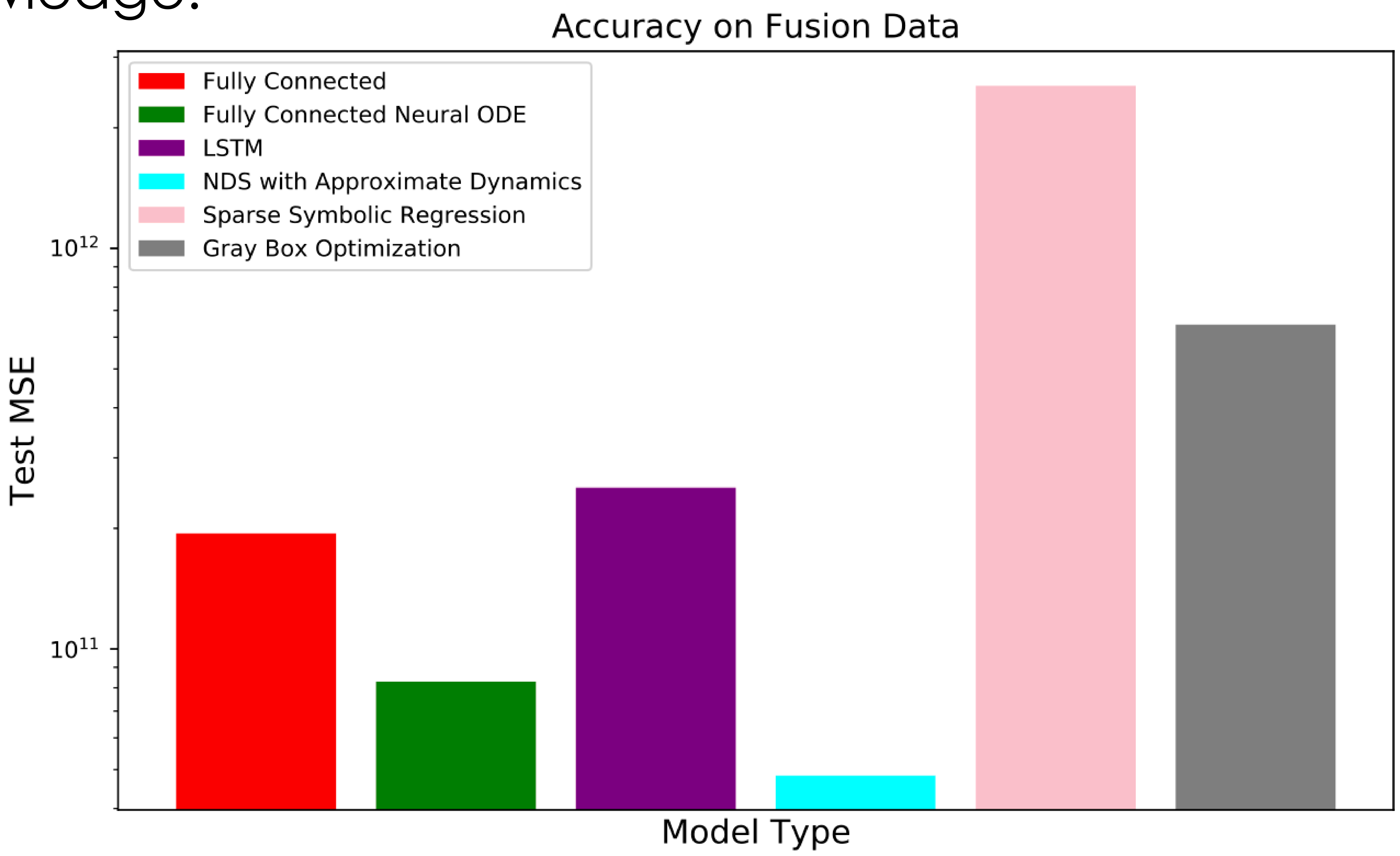
- We define a new class of model, the Neural Dynamical System, as an answer to this question.
- Leverage new methods of training ODE-based neural network models
- Use prior knowledge from physics to improve (1) accuracy and (2) sample complexity
- Can be used for model-predictive control.



Combining data and physics knowledge for modeling a tokamak

- Greatly improved overall accuracy using our neural dynamical system over baselines when we include even simple prior knowledge.
- E is stored energy, P is injected power, T is torque, and ω is rotation.
- Model is from (Boyer et al, Nuclear Fusion, May 2019)

$$\dot{E} = P - \frac{E}{\tau_e} \quad \dot{\omega} = \frac{T}{n_i m_i R_0} - \frac{\omega}{\tau_m}$$



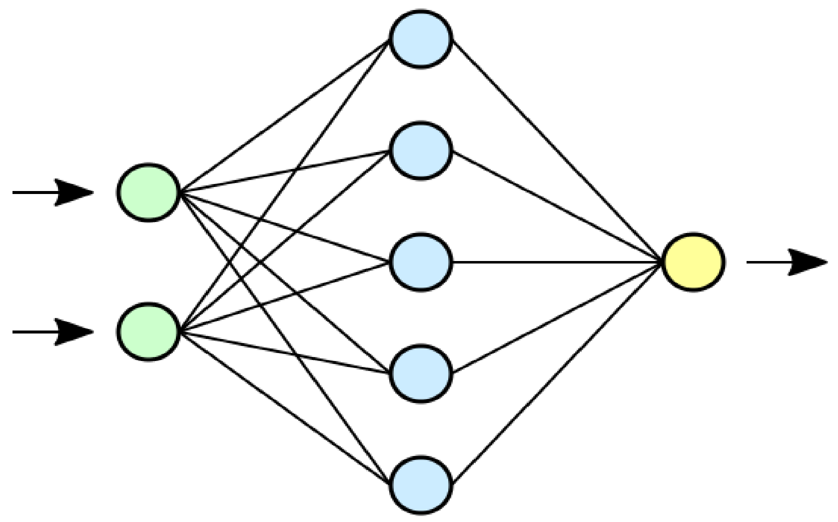
V. Mehta, I. Char, W. Neiswanger, Y. Chung, O. Nelson, D. Boyer, E. Kolemen, J. Schneider, “Neural Dynamical Systems: Balancing Structure and Flexibility in Physical Prediction“, IEEE Conference on Decision and Control (CDC), 2021

Using the Model Offline: Reinforcement Learning

(differentiable)
Control Policy

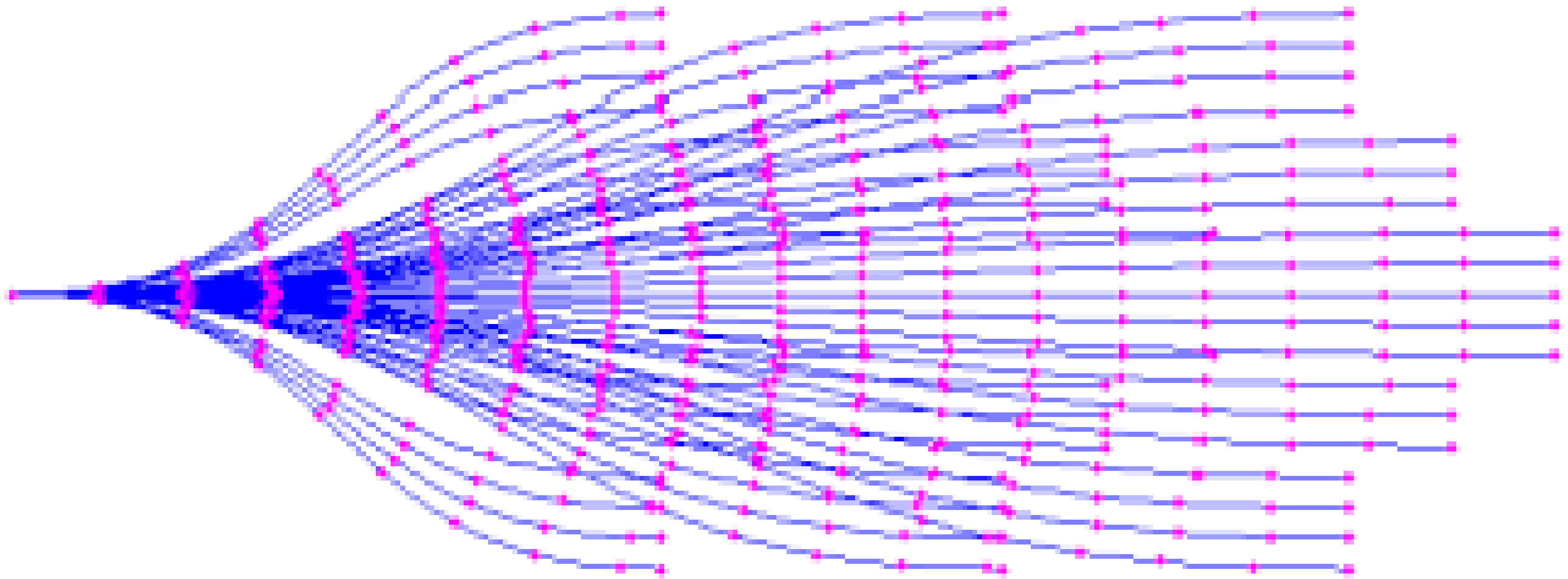
$$u_t = g(x_t; \theta)$$

(differentiable) Model



$$x_{t+1} = f(x_t, u_t; \varphi) + \varepsilon$$

Data: sequences of states and actions



$$D = [\dots, x_t, u_t, x_{t+1}, u_{t+1}, \dots]$$

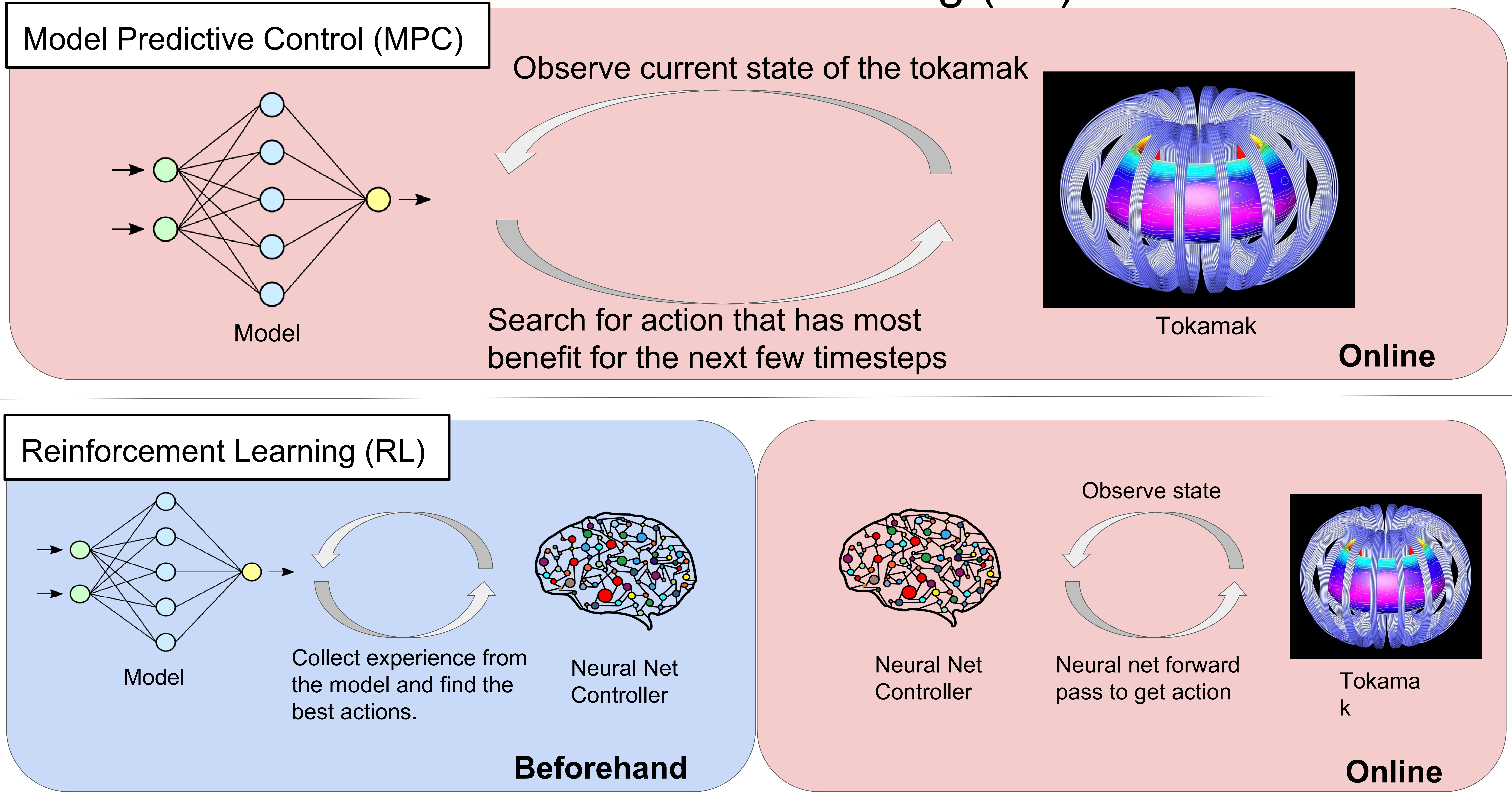
Performance Criterion

$$J(\theta) = E\left(\sum_{t=0}^{N} c(x_t, u_t)\right)$$

<u>Simplified RL</u>

1. Initialize a control policy (random, expert, imitation)
2. Generate some data (true system, model, current policy, exploration policy, external source, replay buffer)
3. Compute a policy gradient, δJ/δθ and update the policy
4. Repeat to step 2

Learning to Control From Data: Model Predictive Control (MPC) vs Reinforcement Learning (RL)

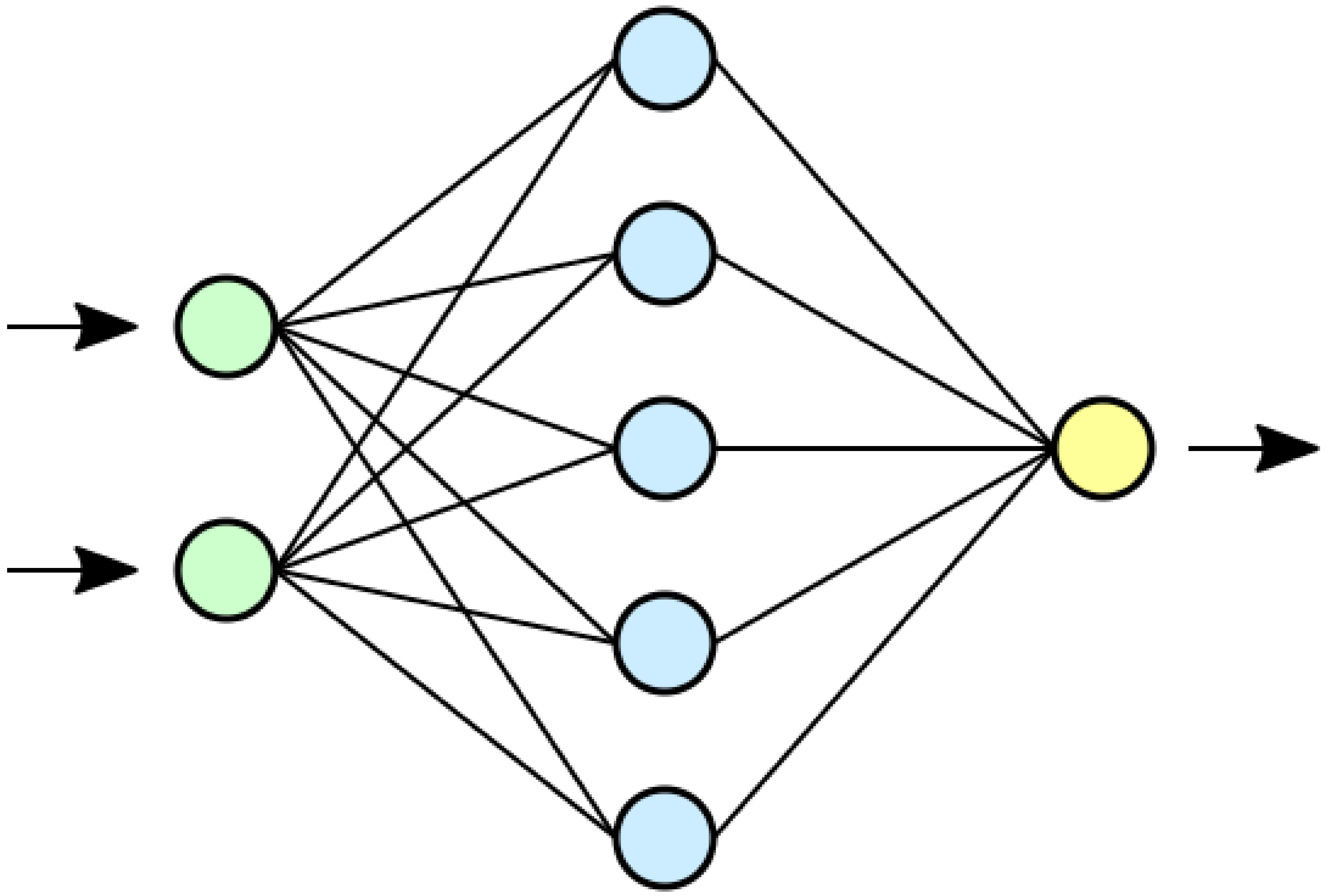


β_N and Rotation Tracking Control Loop: Dynamics Model

Inputs (Dim = 27)

- Current signal + change in last 200ms of:
  - density_estimate
  - li_EFIT01
  - volume_EFIT01
  - kappa_EFIT01
  - a_EFIT01
  - tri_top_EFIT01
  - tri_bot_EFIT01
  - rmagx_EFIT01
  - betan_EFIT01
  - injected power and torque
  - line average plasma rotation
- Current value of bt

- Change in power and torque injected for the next 200ms



Feed Forward Neural Net

Outputs (Dim = 10)

- Predict change in next 200ms for...
  - density_estimate
  - li_EFIT01
  - volume_EFIT01
  - kappa_EFIT01
  - a_EFIT01
  - tri_top_EFIT01
  - tri_bot_EFIT01
  - rmagx_EFIT01
  - betan_EFIT01
  - plasma rotation

Test Explained Variance = 0.587

- All signals are normalized using median and IQR

β_N and Rotation Tracking Control Loop: Training and Evaluation

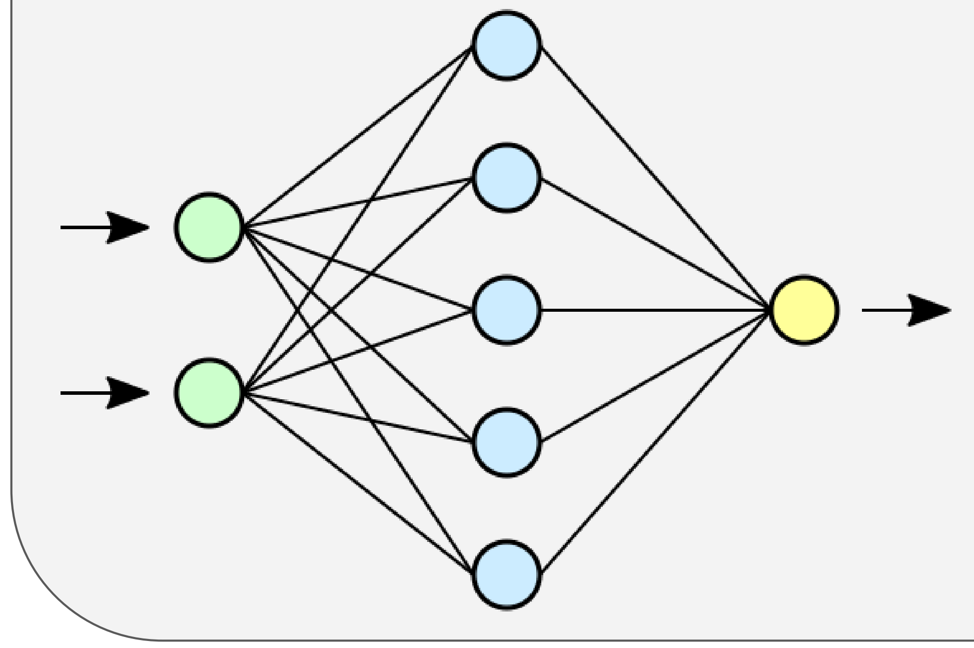
Dataset

Train (90%)

Test (10%)

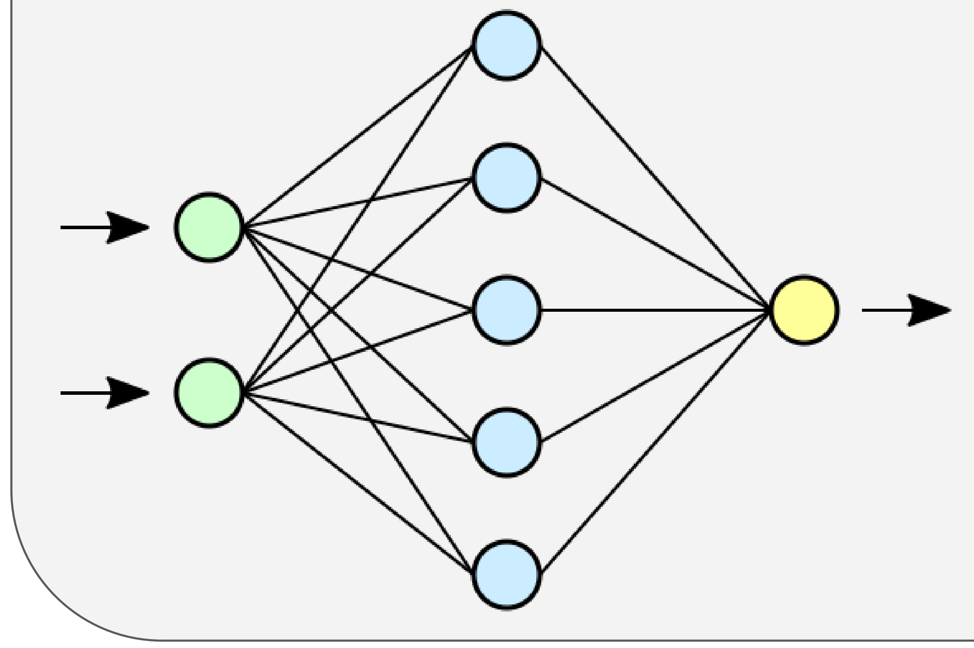
- 55,146 time steps (200ms) in dataset.
- 1,518 different shots in the dataset.
- Splits made by splitting shots randomly.

Train Environment



- Test Explained Variance = 0.581
    - (Averaged Over Output Dimension)
- Used to train controller, tune PID coefficients, and used as the model in MPC

Test Environment



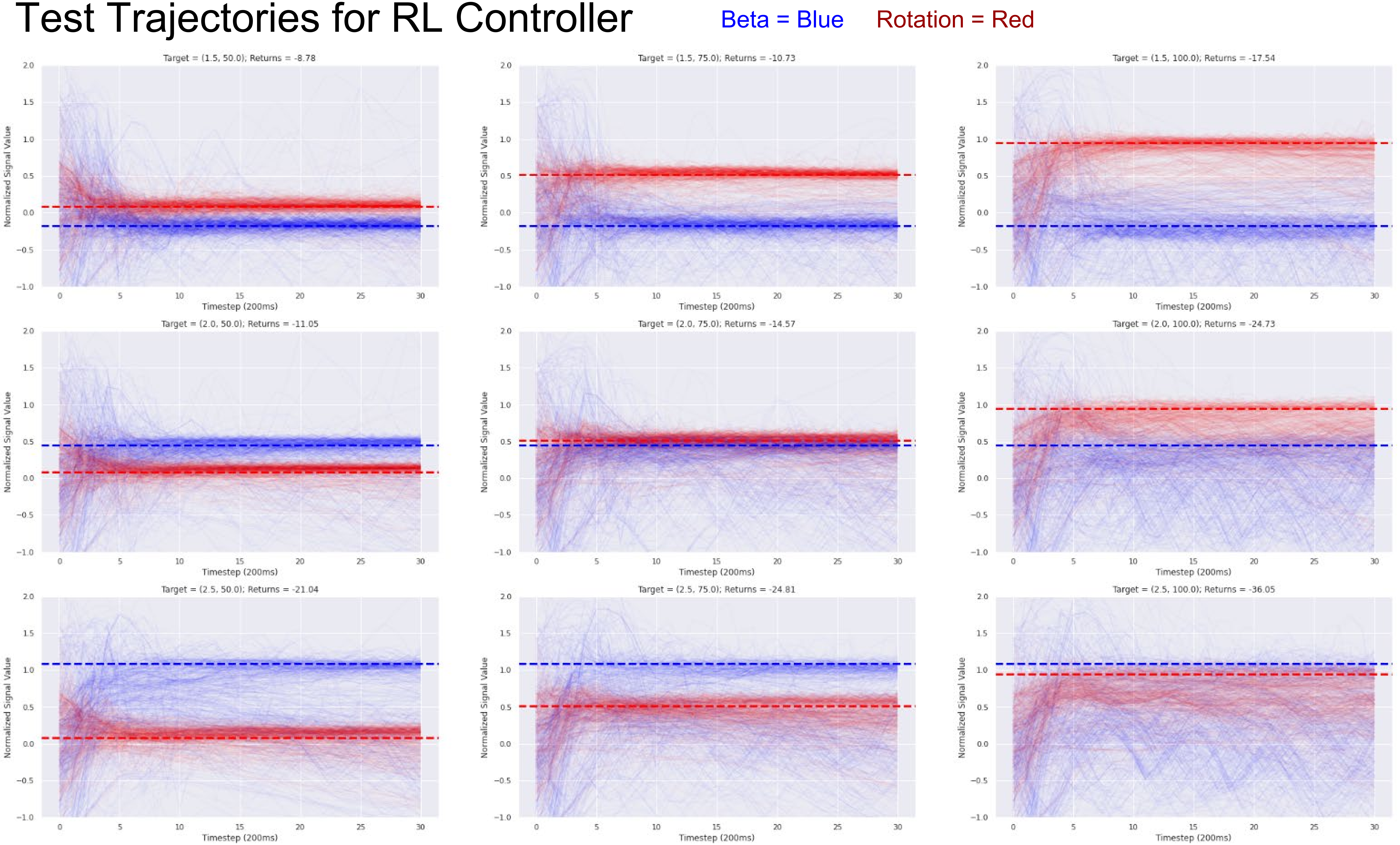
Treated as if it were the real environment. Used for evaluation only.

β_N and Rotation Tracking Control Loop: Results

Control Method	Score
Reinforcement Learning	16.95 ± 0.33
Model Predictive Control	18.45 ± 0.26
Tuned PID Controller	18.87 ± 0.09

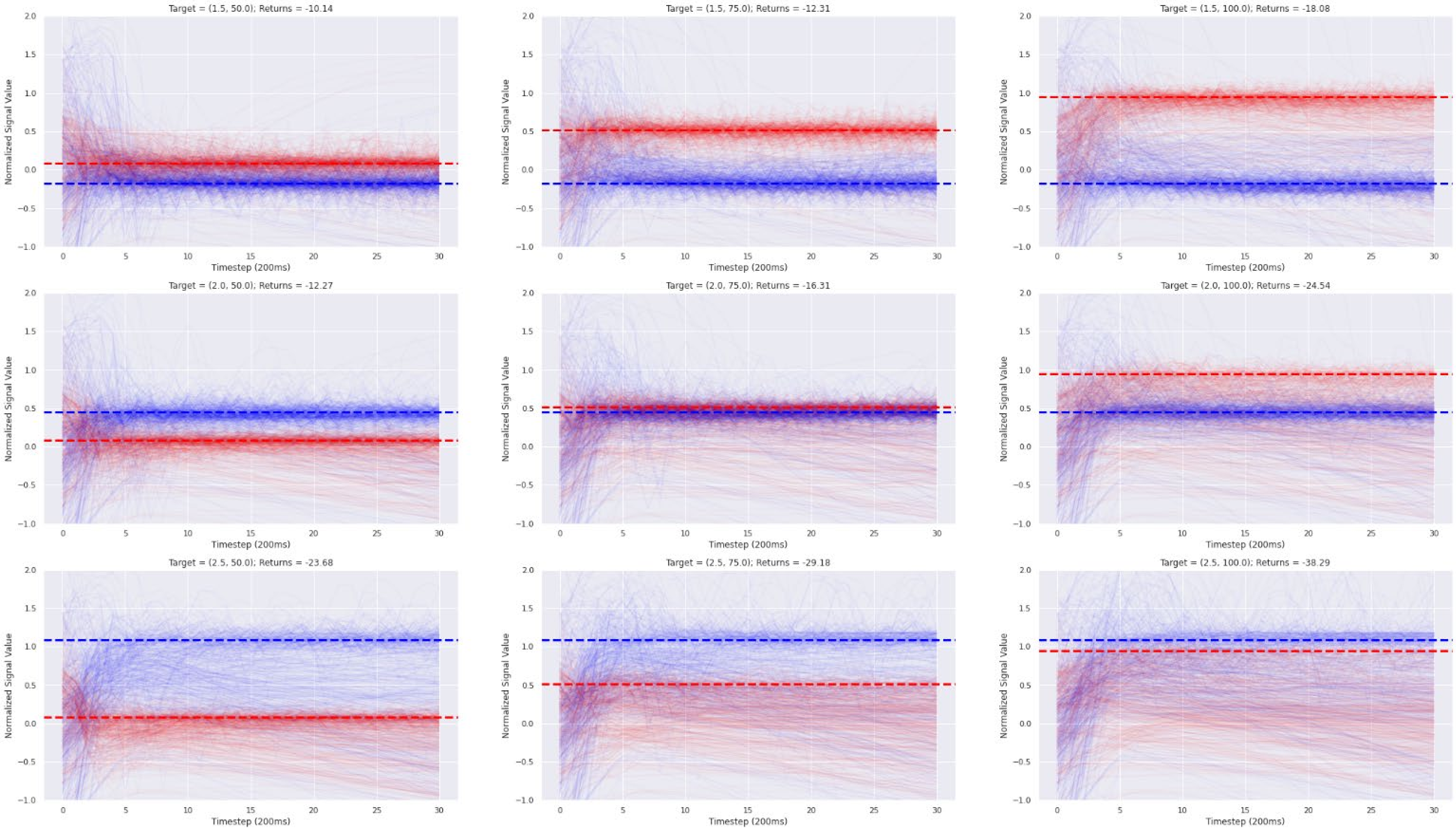
- Score is sum of normalized distance from the two targets, accumulated over the shot and average over the test set (lower is better)

Test Trajectories for RL Controller



Test Trajectories for PID Controller

Beta = Blue Rotation = Red



Summary and Next Steps

- Reinforcement Learning and Bayesian Optimization shows promise fro plasma control in simulation and learned plasma dynamics models

- Test the learned controllers on the real device

- Move from control toward scenario design

Reinforcement Learning for Fusion: Self Driving Cars to Controlled Fusion





Jeff Schneider	Ian Char	Youngseog Chung	Viraj Mehta	Willie Neiswanger
Professor	MLD PhD	MLD **PhD**	RI PhD	Stanford Postdoc

AutonLab

Carnegie Mellon THE ROBOTICS INSTITUTE

Extra Slides

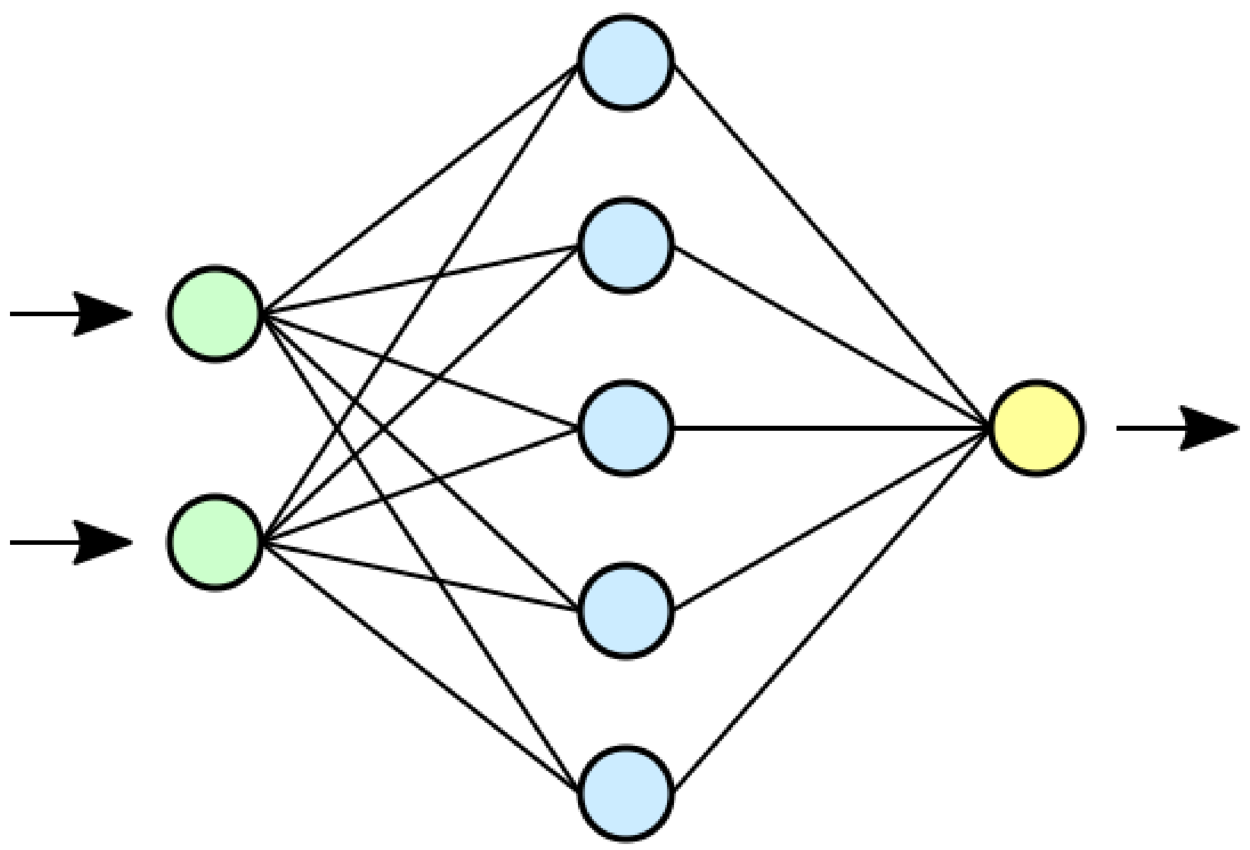
β_N +Rotation Tracking Task

- Additionally try to hit a target (line averaged) plasma rotation.

Changes to Dynamics Model

Inputs (Dim = 27)

- Same as before plus…
- Current value and change from last 200ms of..
  - Line averaged plasma rotation
  - tinj
- Change in tinj for next 200ms



Outputs (Dim = 10)

- Same as before + change in line averaged plasma rotation

Test Explained Variance = 0.587

Changes to Controller

Inputs (Dim = 10)

- Same as before but add
  - Line averaged rotation
  - tinj
  - Target rotation

Output (Dim = 2)

- Desired change in pinj and tinj for 200ms in the future.

- Change in tinj bound
  - [-1.27, 1.36]
- tinj bound
  - [0.22, 6.99]