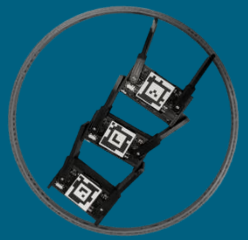
Robot learning on the edge

Online learning in hardware



Thomas A. Berrueta

Postdoctoral Scholar, Computing + Mathematical Sciences

ELLIIT Robot Learning Symposium (2025/11/19)

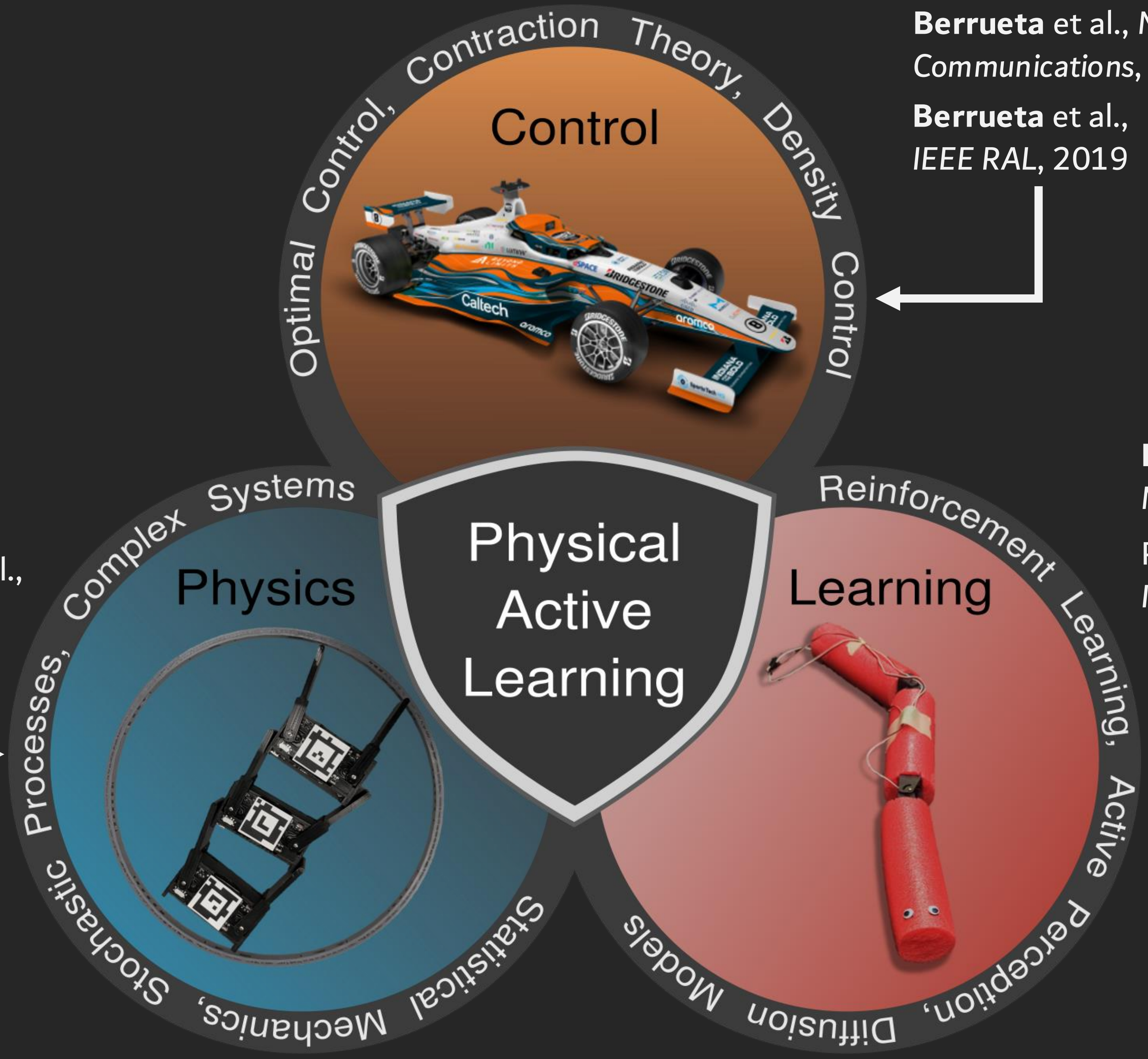
Caltech

Berrueta et al., *Nature Communications*, 2022

Berrueta et al., *IEEE RAL*, 2019

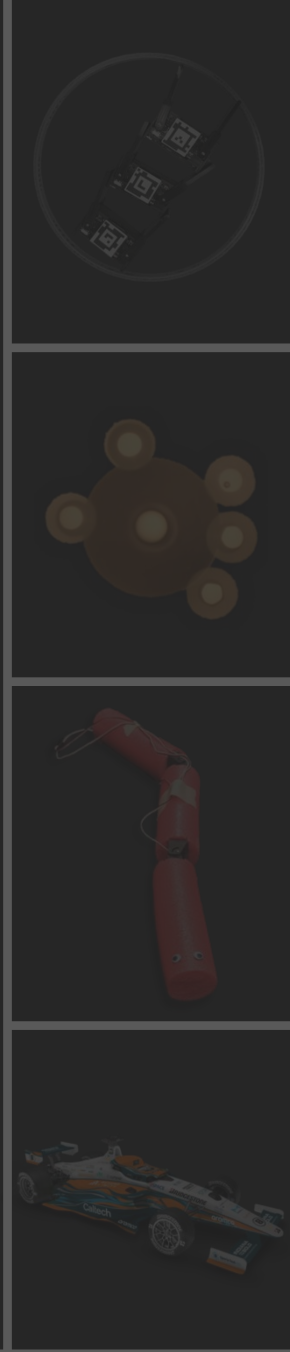
Savoie, **Berrueta** et al., *Science Robotics*, 2019

Chvykov, **Berrueta** et al., *Science*, 2021



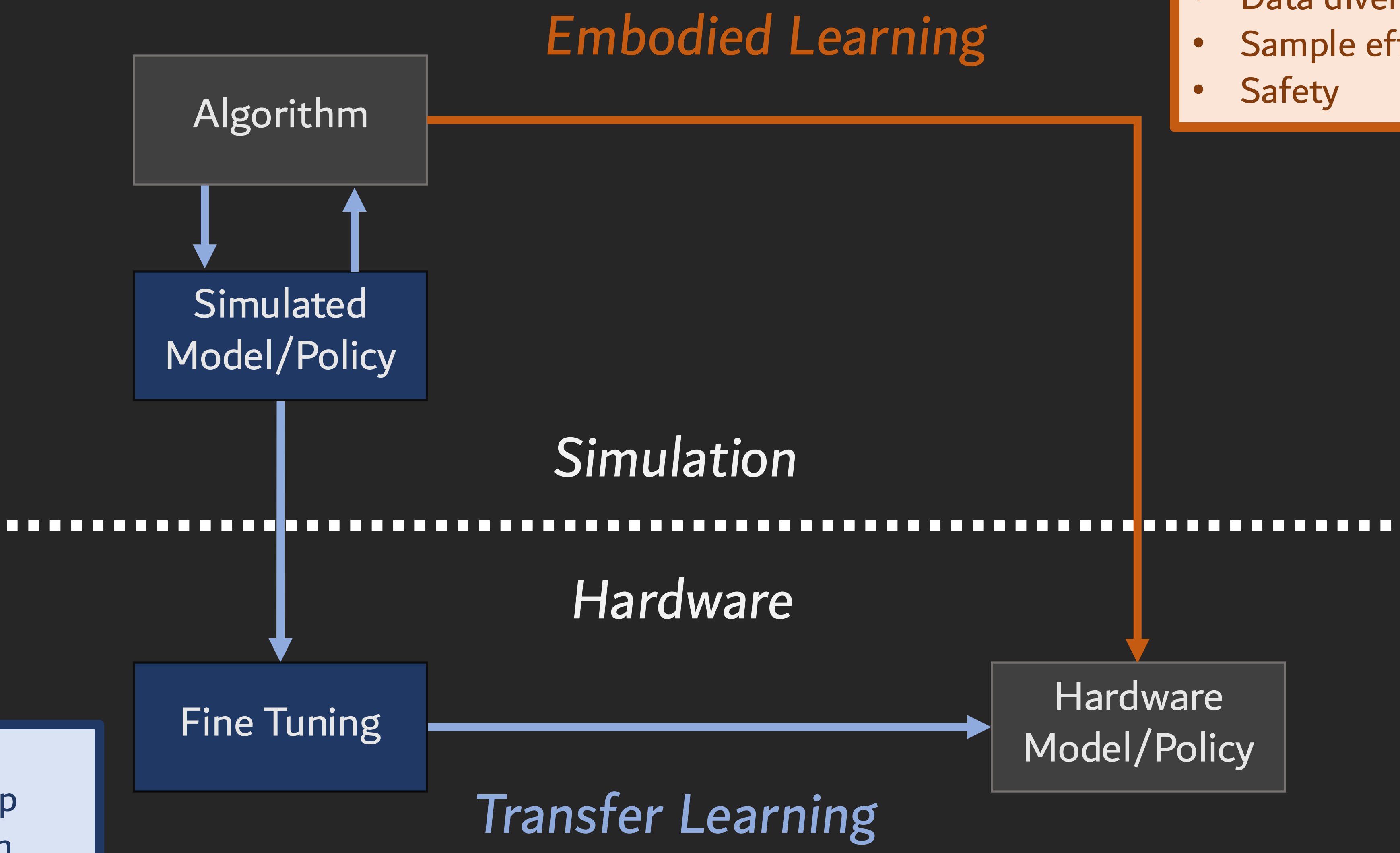
Berrueta et al., *Nature Machine Intelligence*, 2024

Pinosky, **Berrueta** et al., *IEEE CASE*, 2025

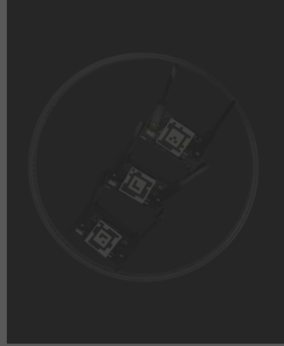


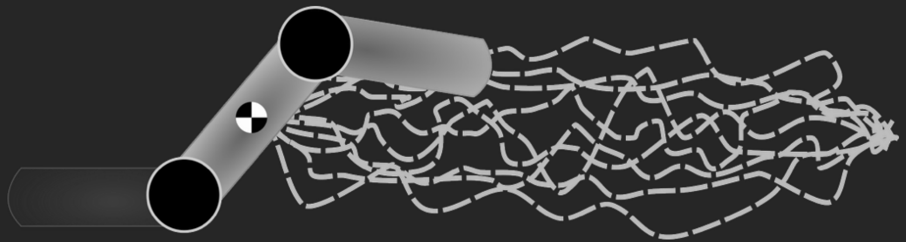

CAST
LAP 1 89.556
LAP 2 100.547
LAP 3 110.962
LAP 4 111.436
LAP 5 121.915
LAP 6 122.360
LAP 7 133.129
LAP 8 133.514
LAP 9 144.130
CAST Racer
THROTTLE
BRAKE
SPEED 146.4 mph
GEAR 5
RPM 5704

Challenges:
- Data diversity
- Sample efficiency
- Safety



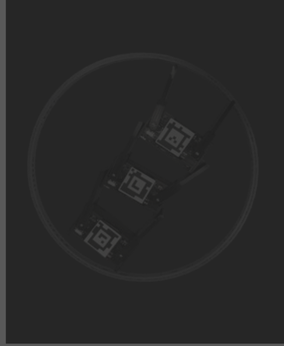
Challenges:
- Reality gap
- Perception
- Hardware

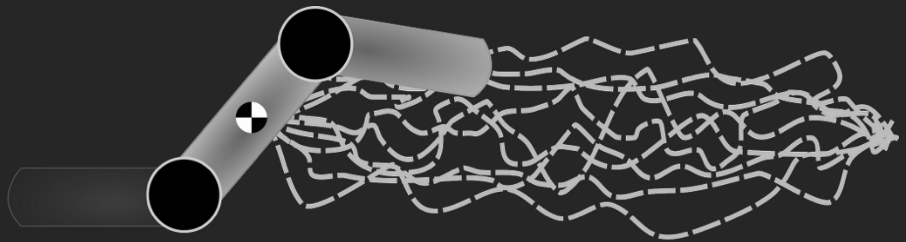


Path decorrelation for efficient online learning

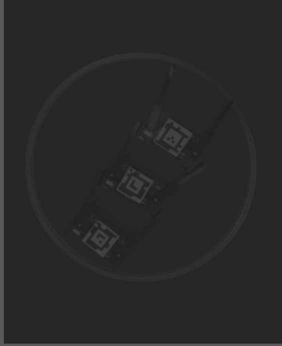
Operational safety through modular design



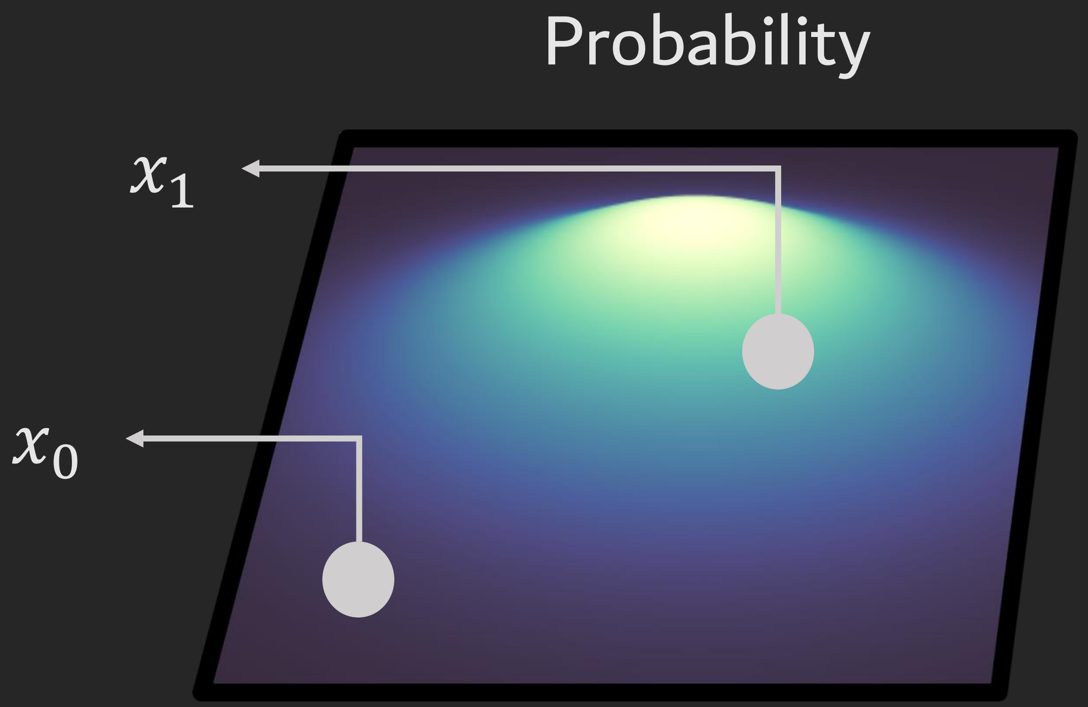


Path decorrelation for efficient online learning

Operational safety through modular design

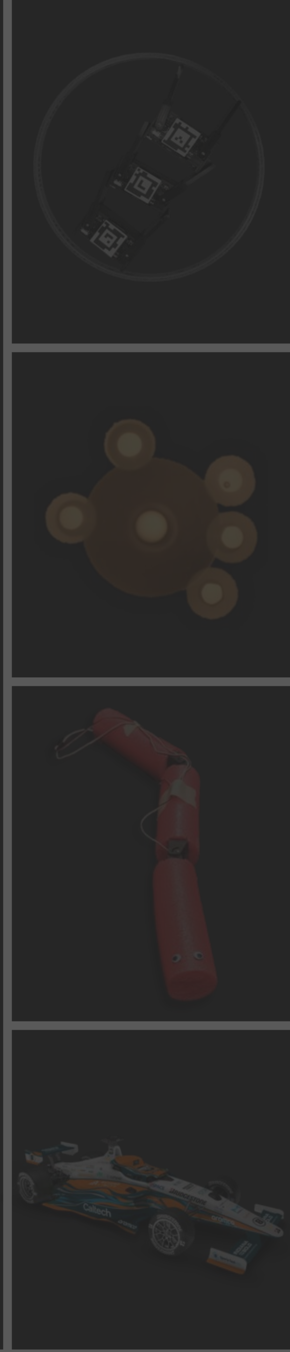


Physics gets in the way of data diversity

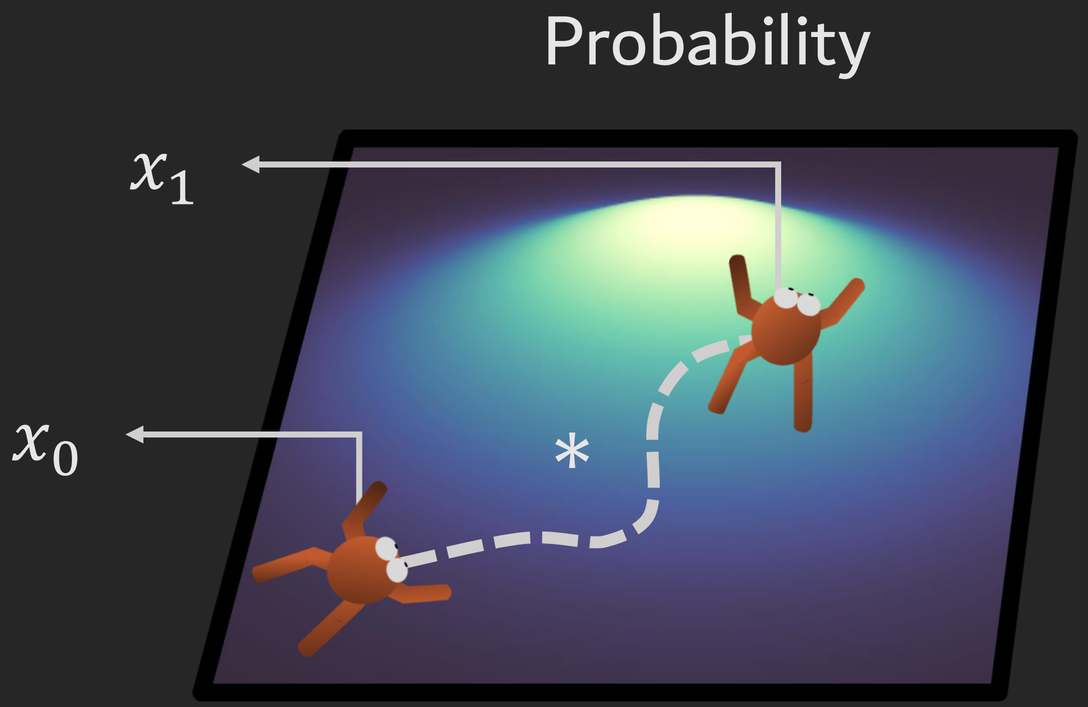


$$p(x_0, x_1) = p(x_1)p(x_0)$$

$$S[p(x_0, x_1)] = S[p(x_1)] + S[p(x_0)]$$



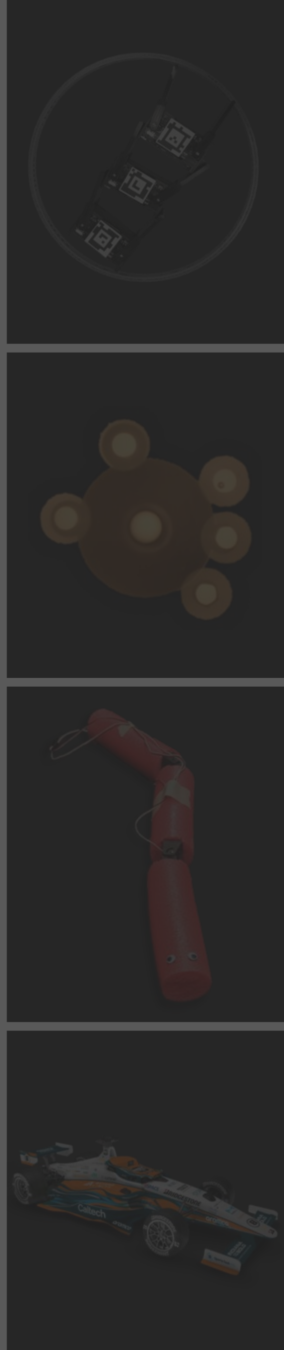
Physics gets in the way of data diversity

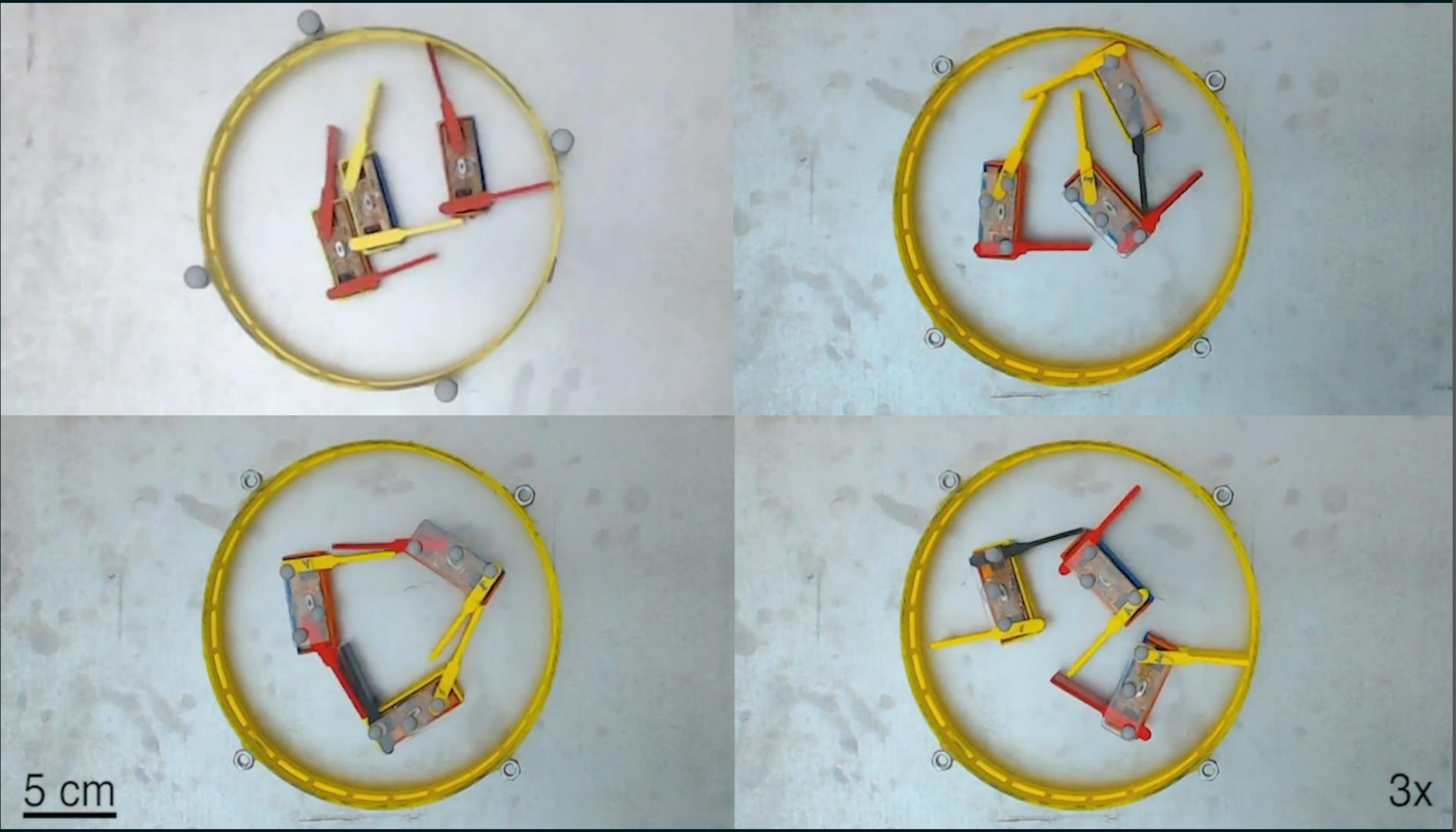


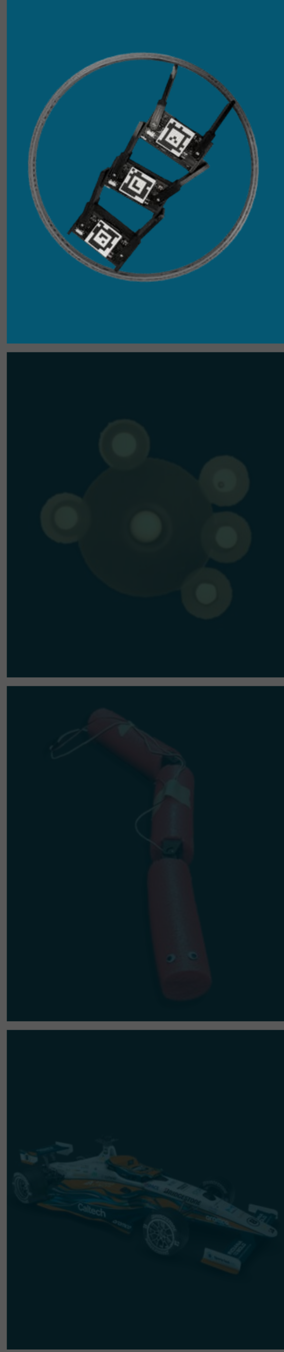
* Transitions along path distribution (i.e., dynamics)

$$p(x_0, x_1) = p(x_1|x_0)p(x_0)$$

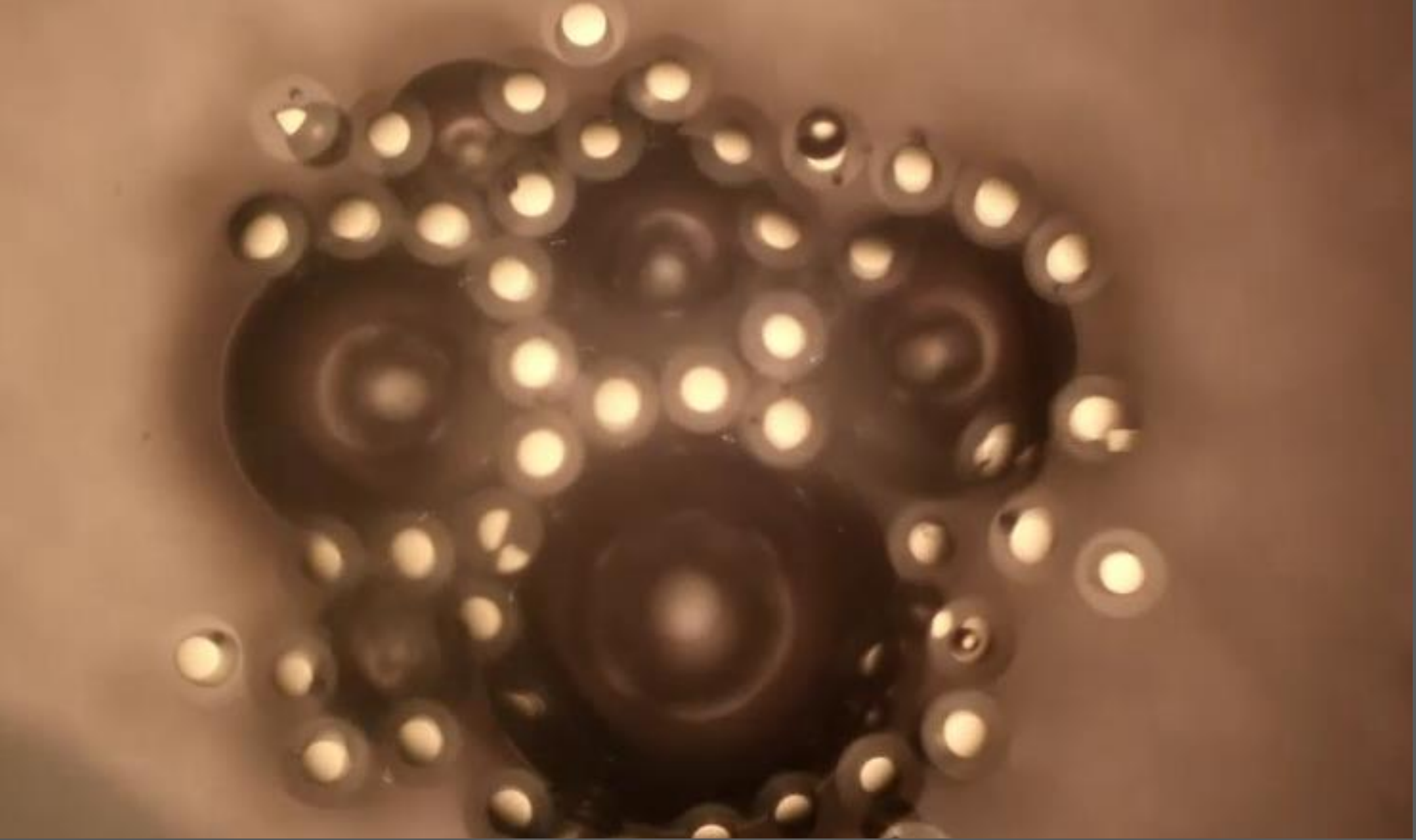
$$S[p(x_0, x_1)] \leq S[p(x_1)] + S[p(x_0)]$$

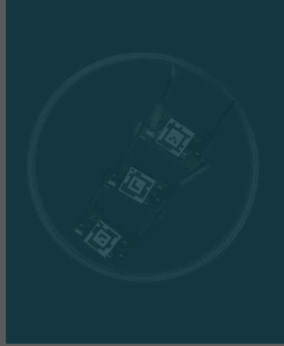


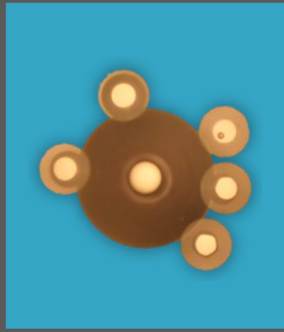




Chvykov, **Berrueta**, et al., “Low rattling: A predictive framework for self-organization in active collectives,” *Science*, 371 (6524), 90-95 (2021)

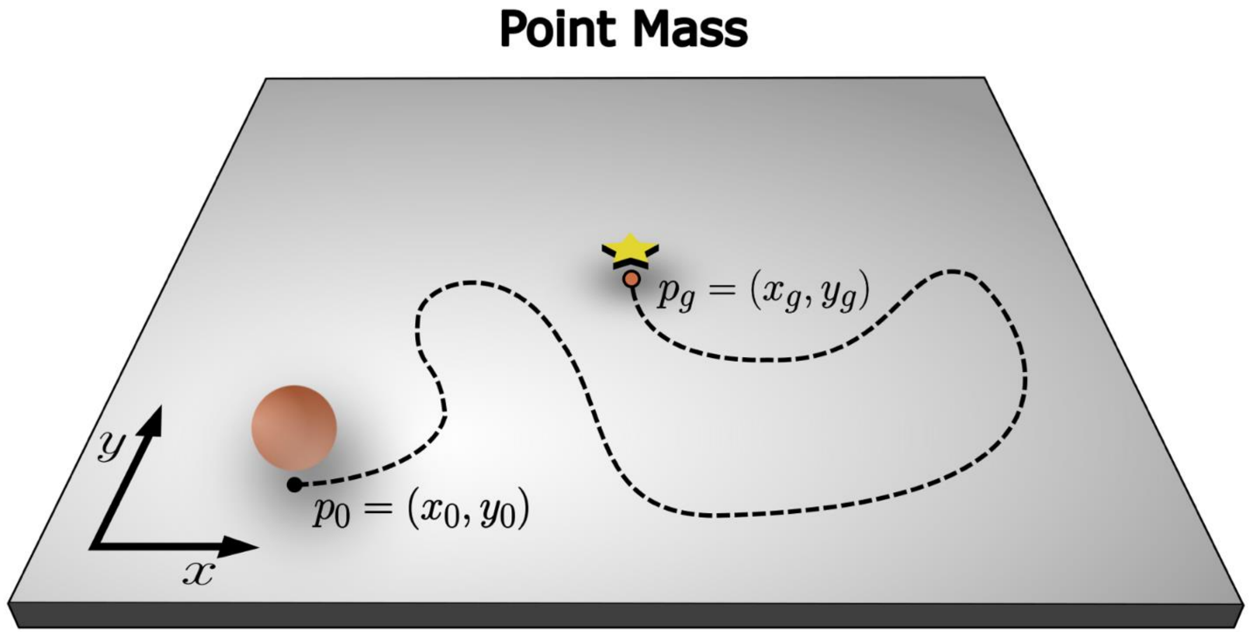






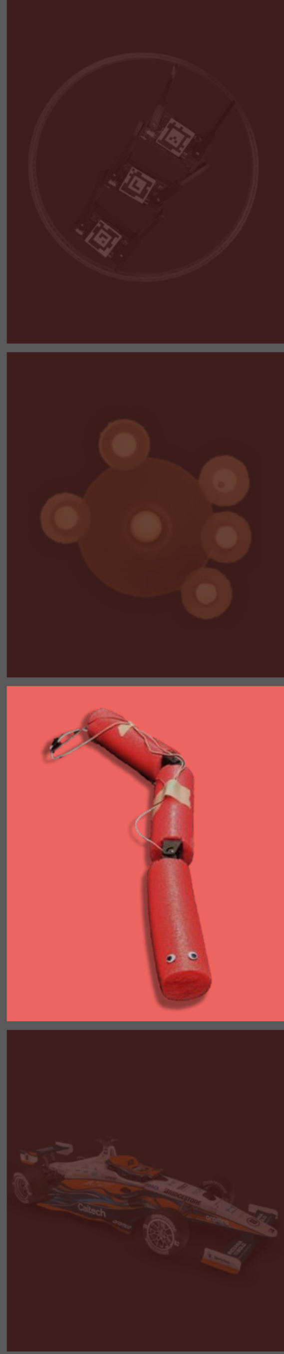
Berrueta, et al., “Emergent microrobotic oscillators via asymmetry-induced order,” *Nature Communications*, 13 (1), 1-11 (2022)

Decorrelating sample paths

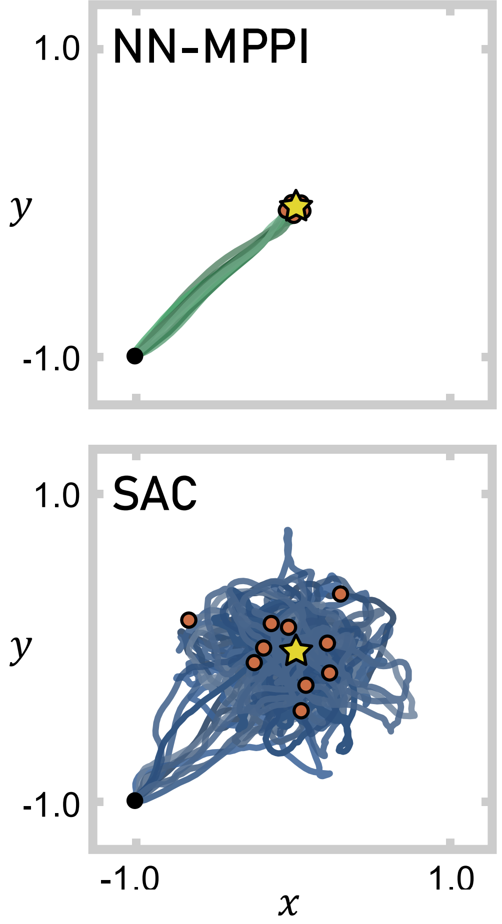


Dynamics: $\vec{x}_{t+1} = A\vec{x}_t + B\vec{u}_t$

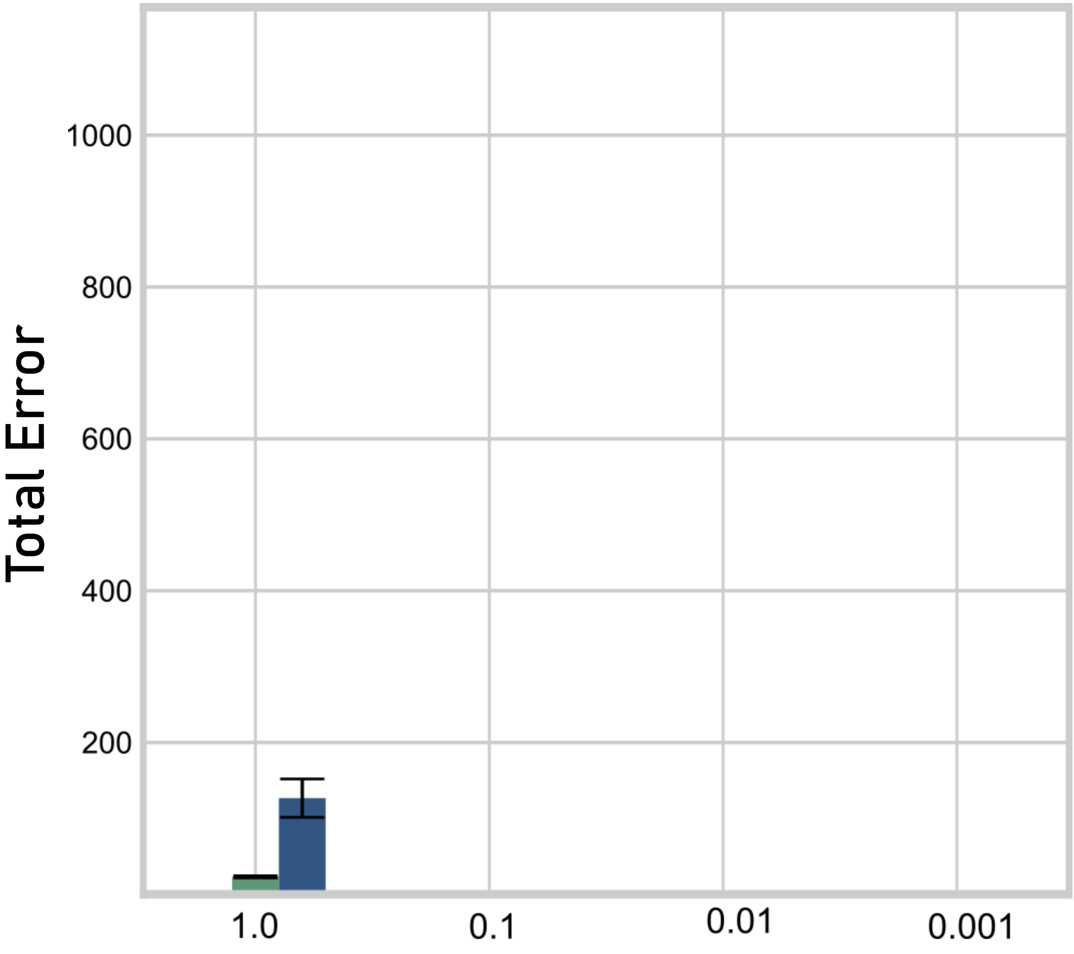
$$A = \begin{bmatrix} 1 & 0 & \beta & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}$$



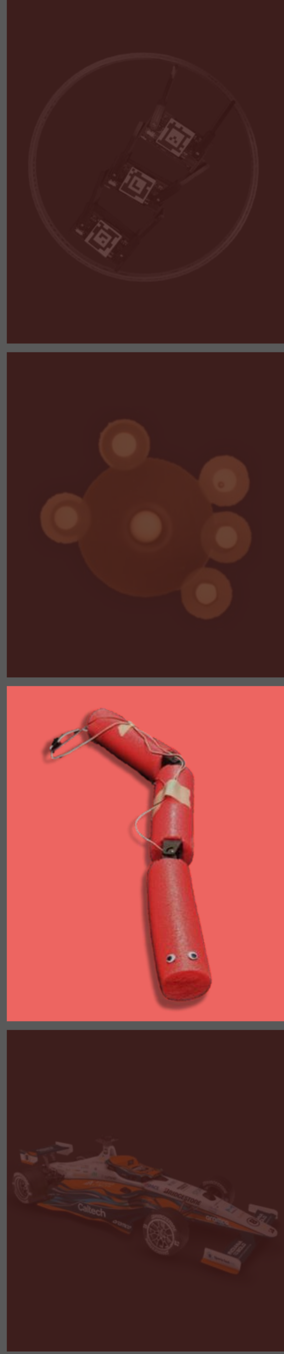
Decorrelating sample paths



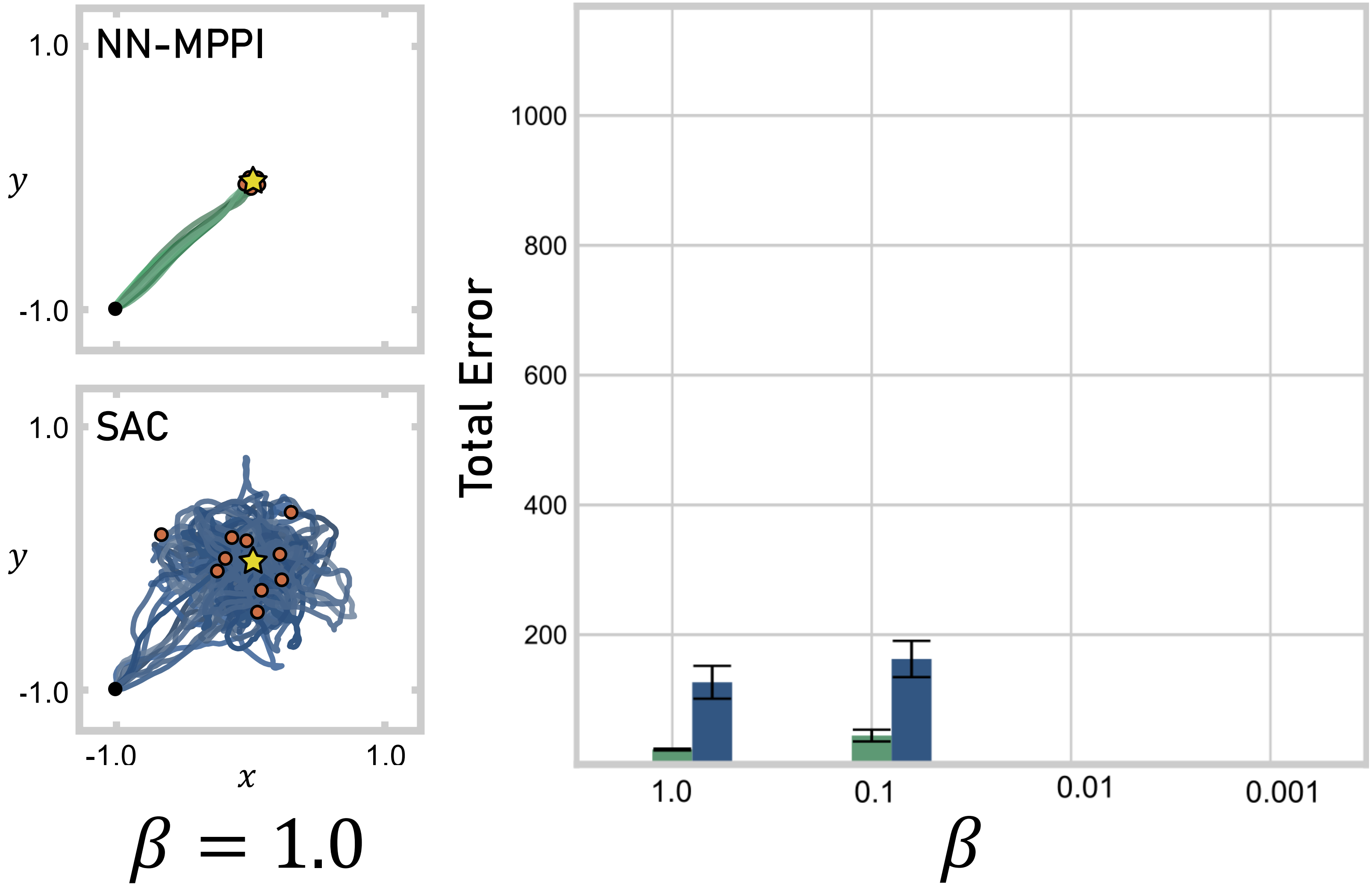
$\beta = 1.0$

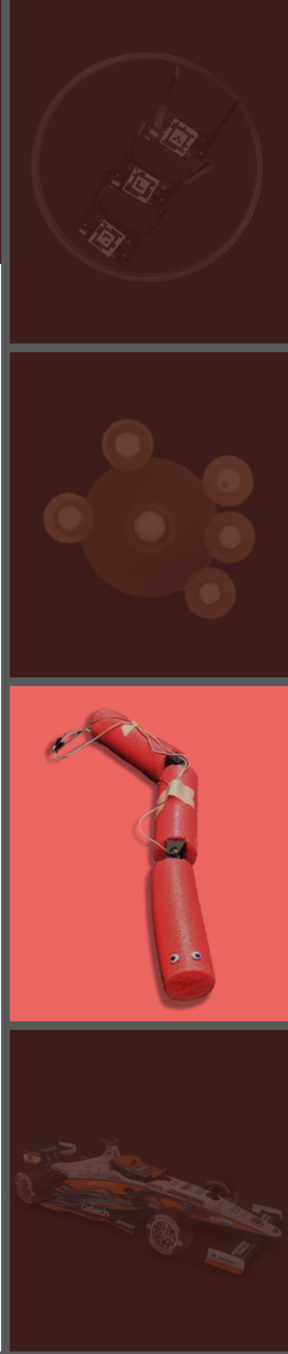


β

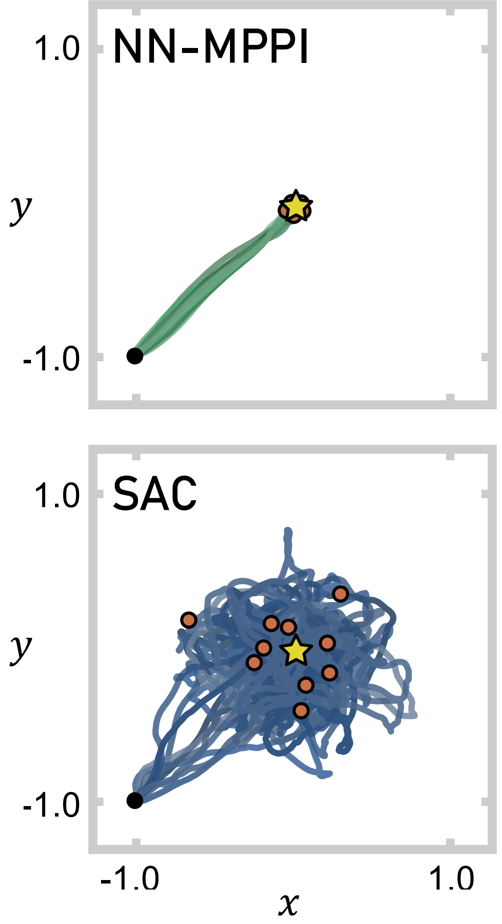


Decorrelating sample paths

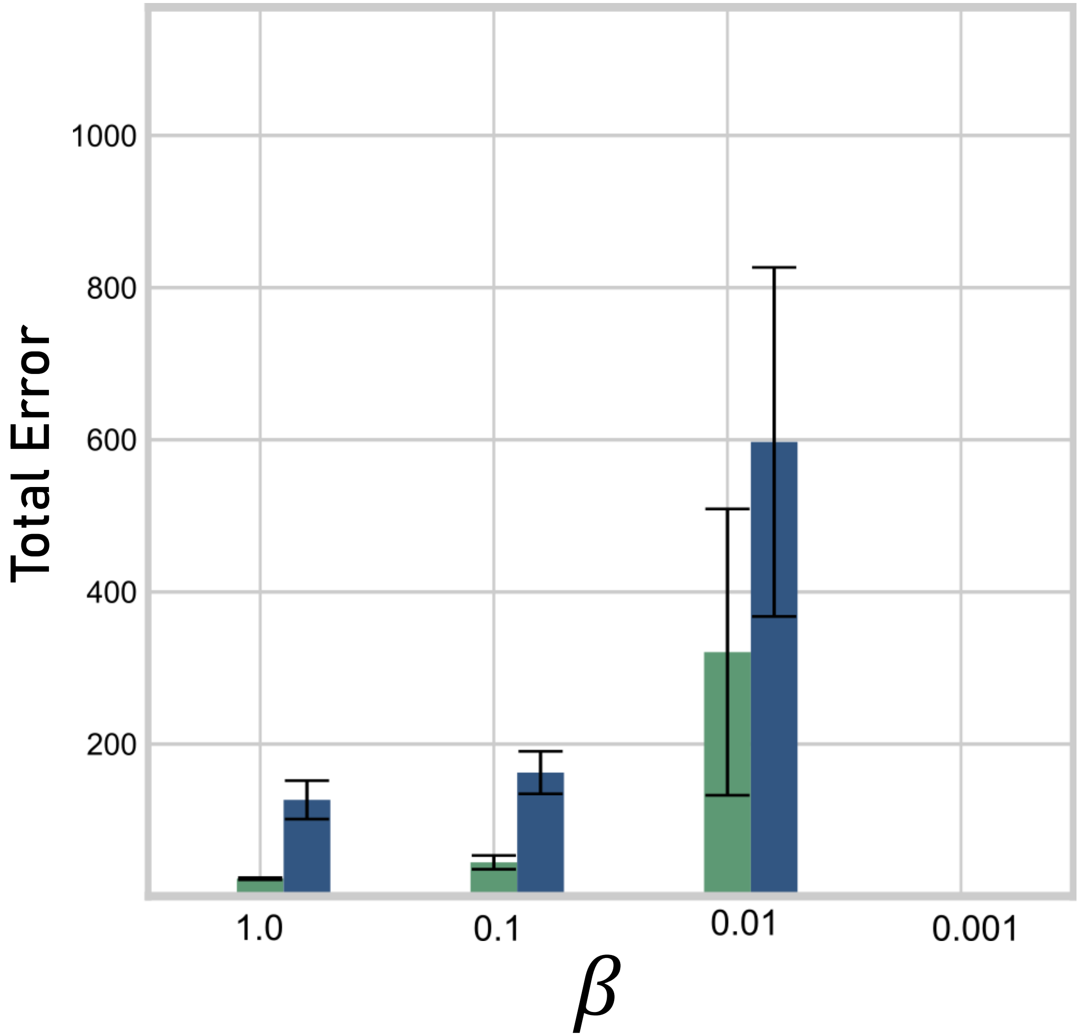


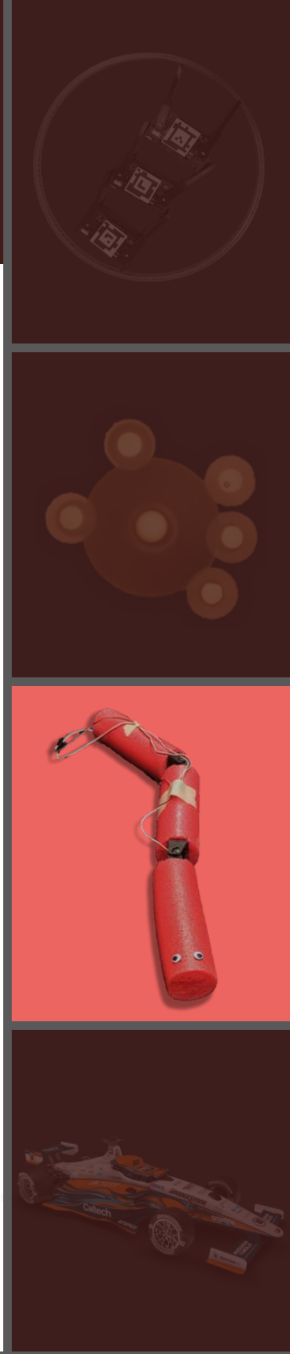


Decorrelating sample paths

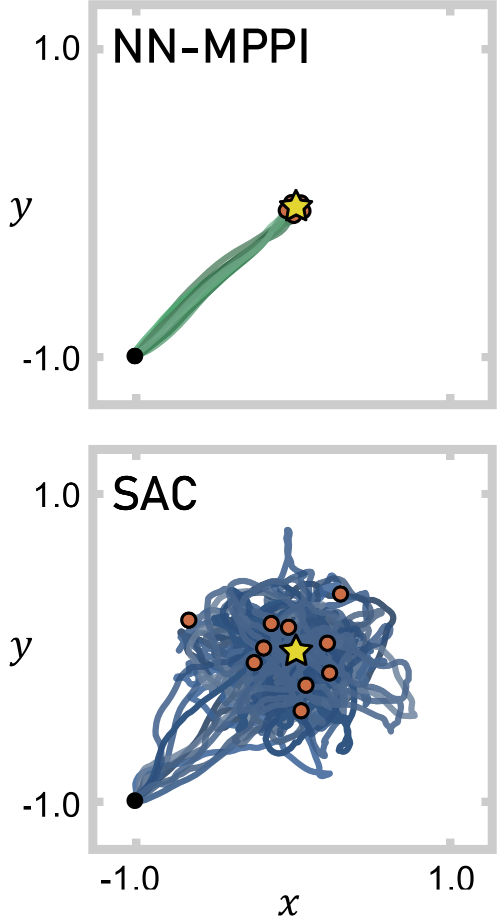


$\beta = 1.0$

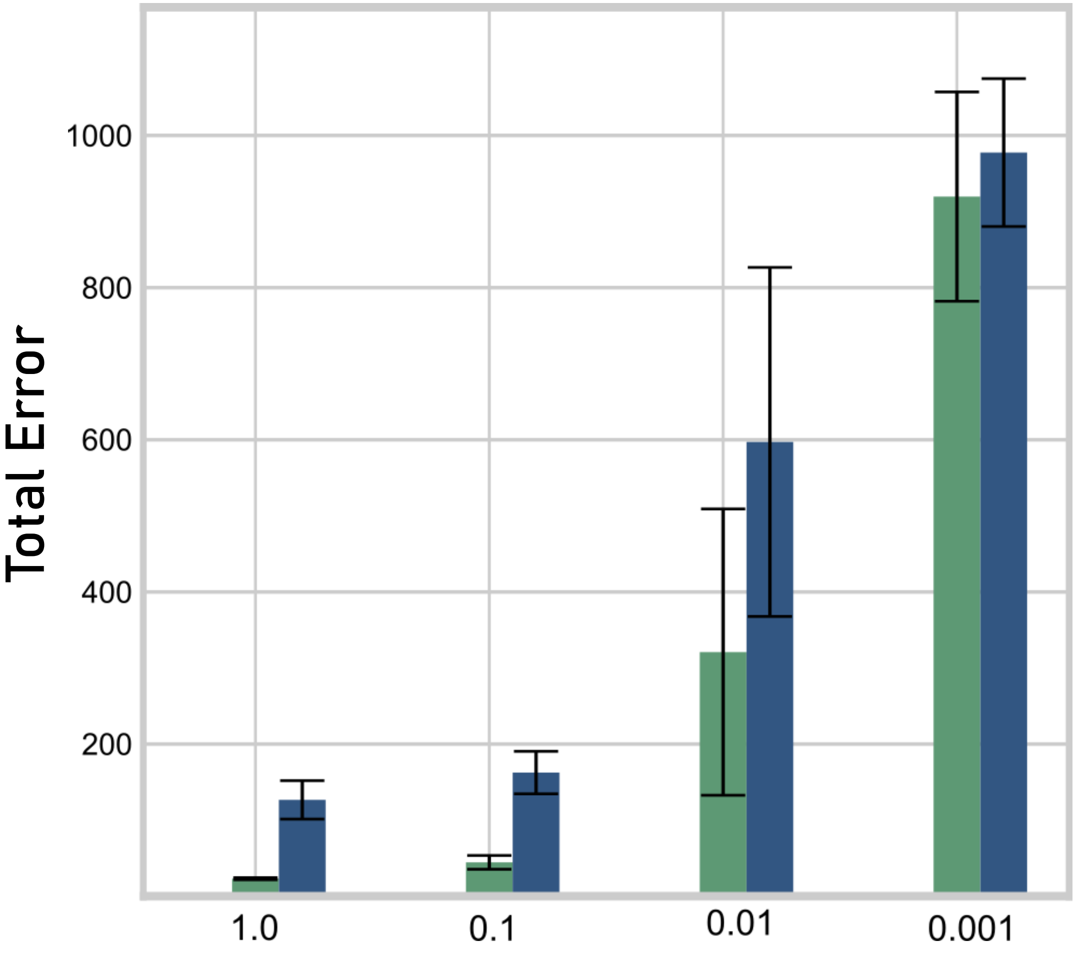




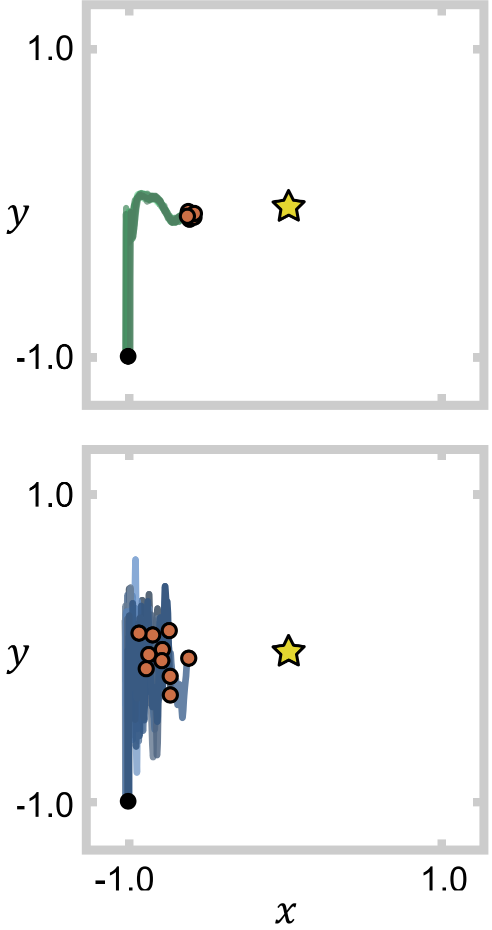
Decorrelating sample paths



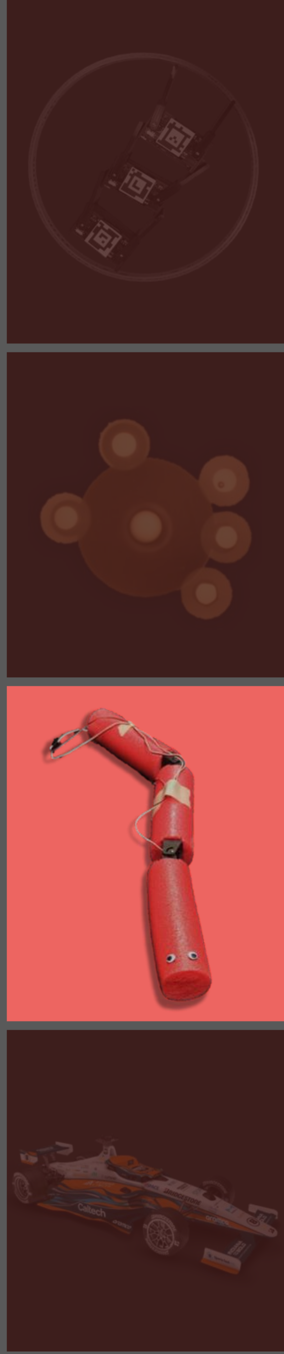
$\beta = 1.0$



β



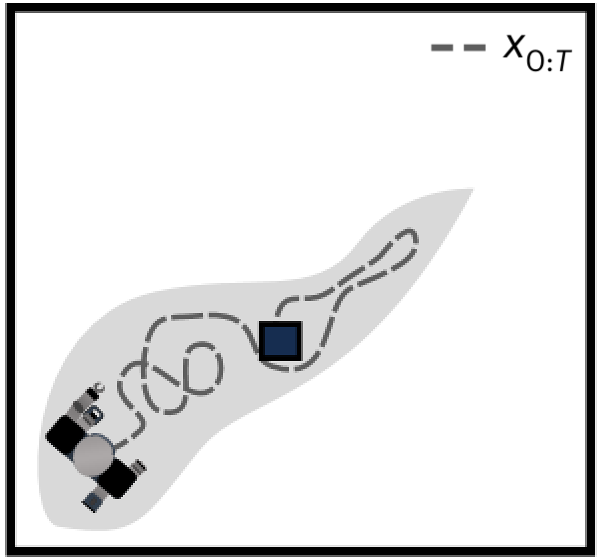
$\beta = 0.001$



Decorrelating sample paths

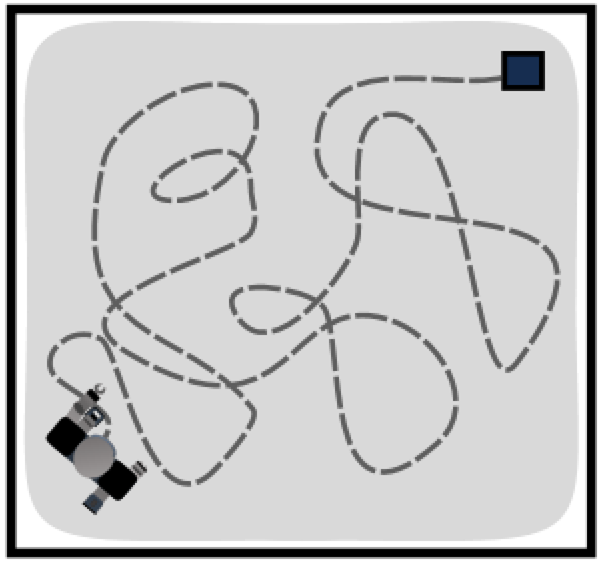
- Entropy maximization as a means of sample path decorrelation.
- However, this is intractable for general unknown dynamics.

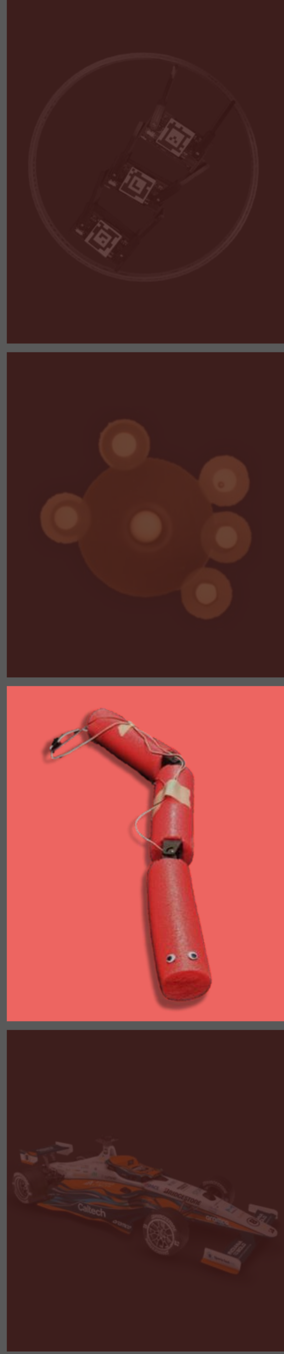
$$P[x_{0:T}] = p(x_0)\prod_{t=0}^{T} p(x_{t+1}|x_t)$$



$$\underset{P[x_{0:T}]}{\text{argmax}}\ S[P[x_{0:T}]]$$

$$P_{\max}[x_{0:T}]$$





Decorrelating sample paths

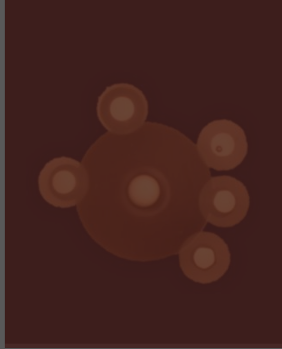
$$\underset{P[x_{0:T}]}{\operatorname{argmax}} - \int P[x_{0:T}] \log P[x_{0:T}] \mathcal{D}x_{0:T} \; - \lambda_0 \left(\int P[x_{0:T}] \mathcal{D}x_{0:T} - 1 \right)$$

$$- \int Tr \Big(\Lambda(x^*)^\intercal \big(\langle \Delta x^2_{0:T} \rangle_{x^*} - \mathbf{C}[x^*] \big) \Big) dx^*$$

where:

$$\langle \Delta x^2_{0:T} \rangle_{x^*} = \int P[x_{0:T}] \left[\sum_{i=0}^{T} (x_{i+1} - x_i)^\intercal (x_{i+1} - x_i) \delta(x_i - x^*) \right] \mathcal{D}x_{0:T}$$

$$\mathbf{C}[x^*] = \sum_{\tau = t_i}^{t_i + \Delta t} K_{XX}(t_i, \tau), \quad (w/ x_{t_i} = x^*)$$



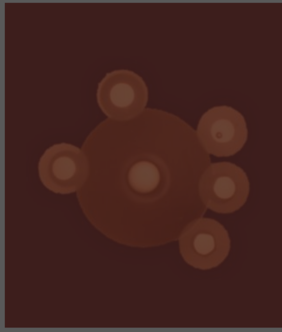
Decorrelating sample paths

- For systems with continuous sample paths, we can prove that:

$$\max_{P} S[P[x_{0:T}]] \leq \sum_{t=0}^{T} \boxed{\frac{1}{2} \log \det \mathbf{C}[x_t]} \propto \text{log-Volume of locally reachable states}$$

which is a concave and easily computable quantity.

- Optimizing this expression leads to diffusive exploration, because its optimum describes the sample paths of a class of diffusion processes.

Berrueta, et al., "Maximum diffusion reinforcement learning," *Nature Machine Intelligence*, 6 (5), 504-514 (2024)

Maximum diffusion reinforcement learning

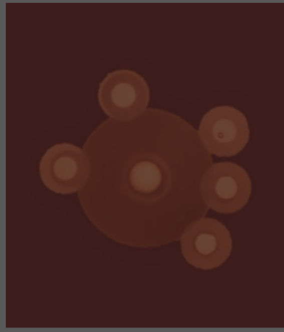
- To illustrate these results, we developed an RL pipeline based on our derivations:

$$\underset{\pi}{\operatorname{argmax}}\; E_{p,\pi}\left[\sum_{t=0}^{T} \hat{r}(x_t, u_t)\right]$$

- The rewards are augmented with a term that decorrelates sample paths:

$$\hat{r}(x_t, u_t) = r(x_t, u_t) + \frac{\alpha}{2} \log\det \mathbf{C}[x_t]$$

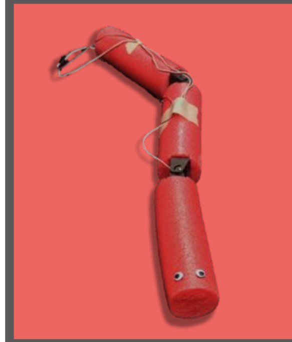
- Agents that optimize MaxDiff objectives are <u>ergodic</u> and asymptotically inherit robustness and online learning guarantees.

Berrueta, et al., "Maximum diffusion reinforcement learning," *Nature Machine Intelligence*, 6 (5), 504-514 (2024)

Maximum diffusion reinforcement learning





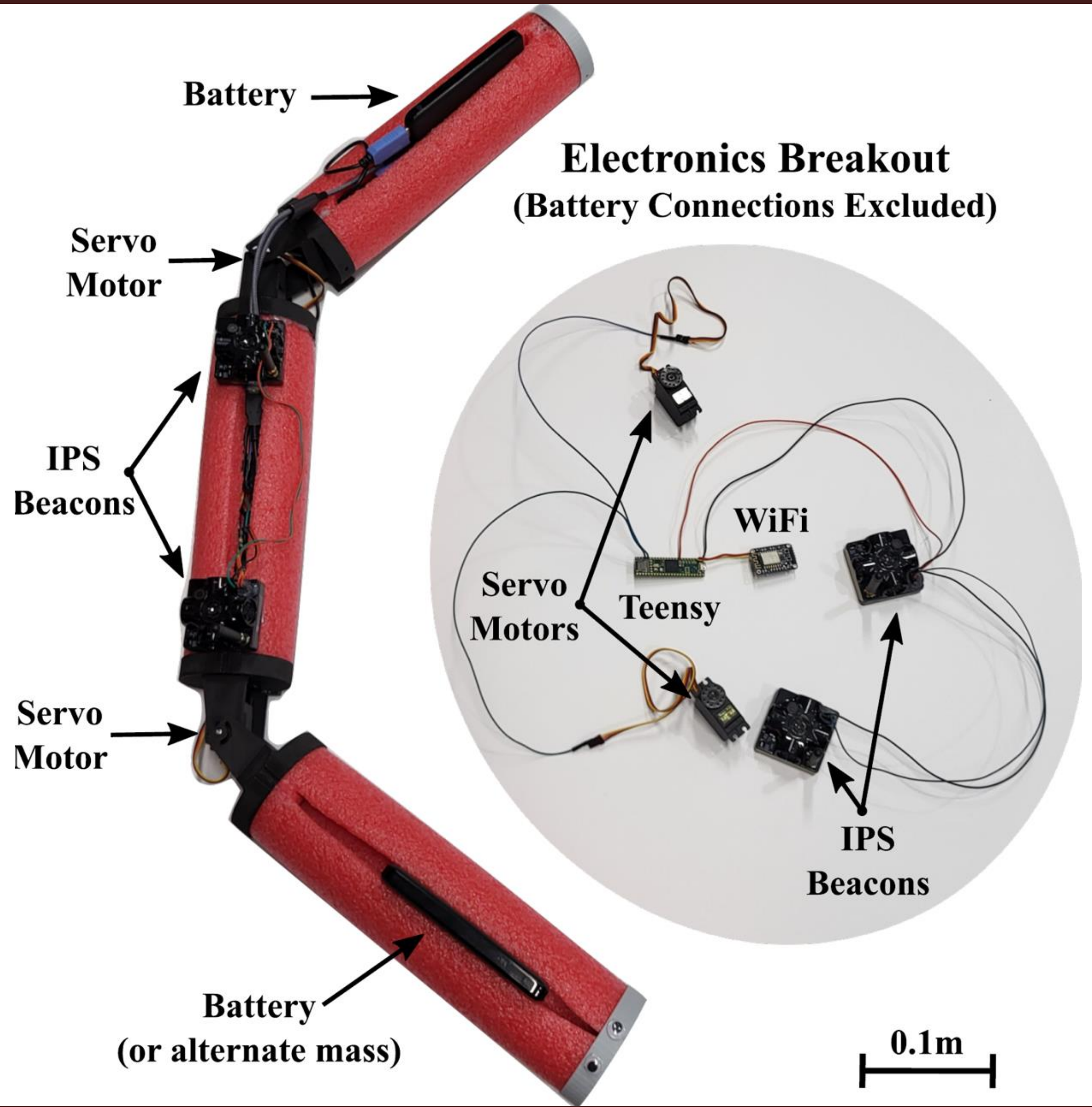

Theorem 2. *(MaxDiff RL Agents are Reliable) If there exists a PAC-MDP algorithm with policy* $\pi^{\max}$*, then the Markov chain induced by* $\pi^{\max}$ *is ergodic and* $\pi^{\max}$ *will be* ϵ*-optimal regardless of initialization.*

Theorem 3. *(MaxDiff RL Agents can Learn in Single-Shot) If there exists a PAC-MDP algorithm with policy* $\pi^{\max}$*, then the Markov chain induced by* $\pi^{\max}$ *is ergodic and any realization of* $\pi^{\max}$ *will asymptotically achieve the same* ϵ*-optimality as an ensemble.*

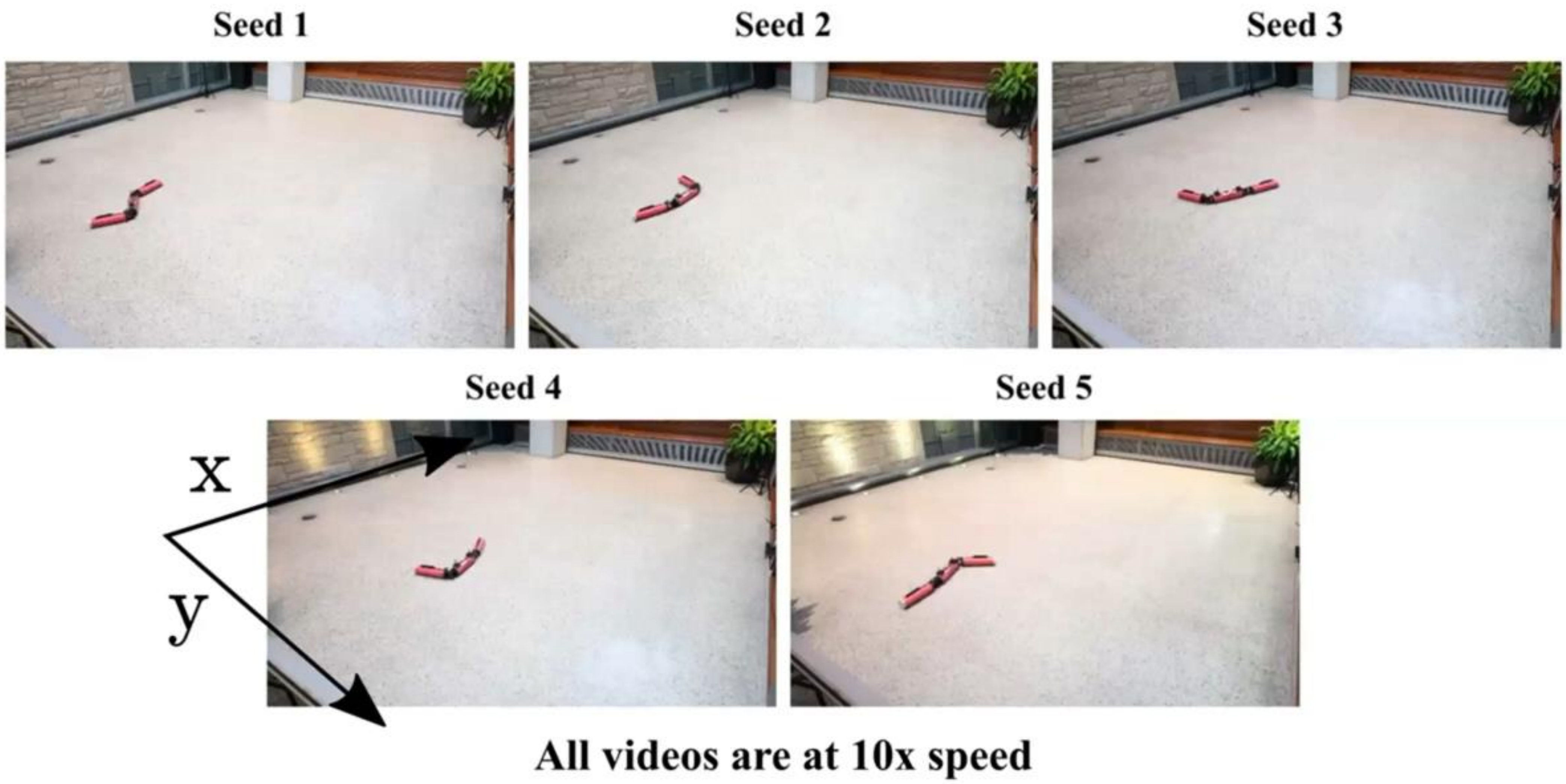
- To be PAC-MDP is ϵ-optimal $(1-\delta)$-percent of the time:

$$\Pr(\mathcal{V}_{\pi^*}(x_0) - \mathcal{V}_{\pi^{\max}}(x_0) \leq \epsilon) \geq 1 - \delta$$

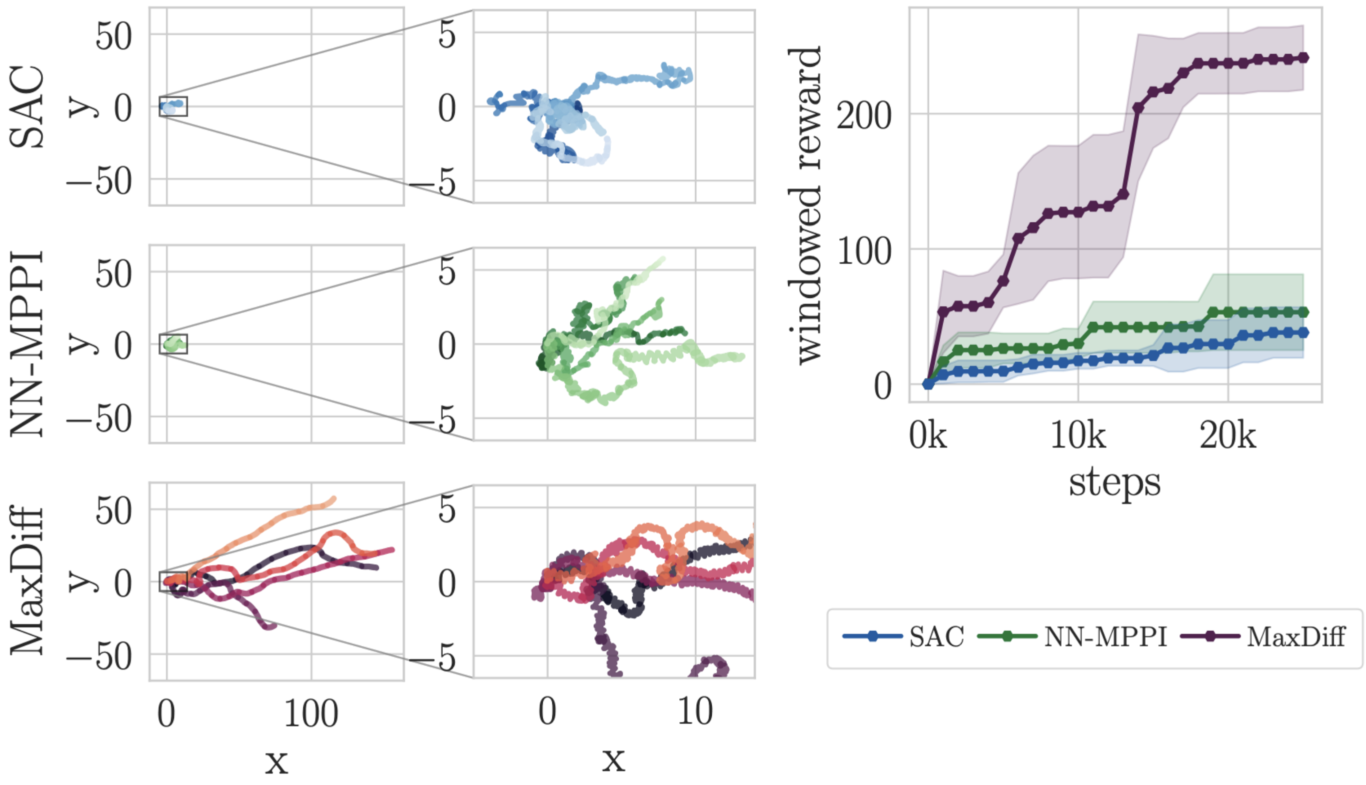
Berrueta, et al., “Maximum diffusion reinforcement learning,” *Nature Machine Intelligence*, 6 (5), 504-514 (2024)



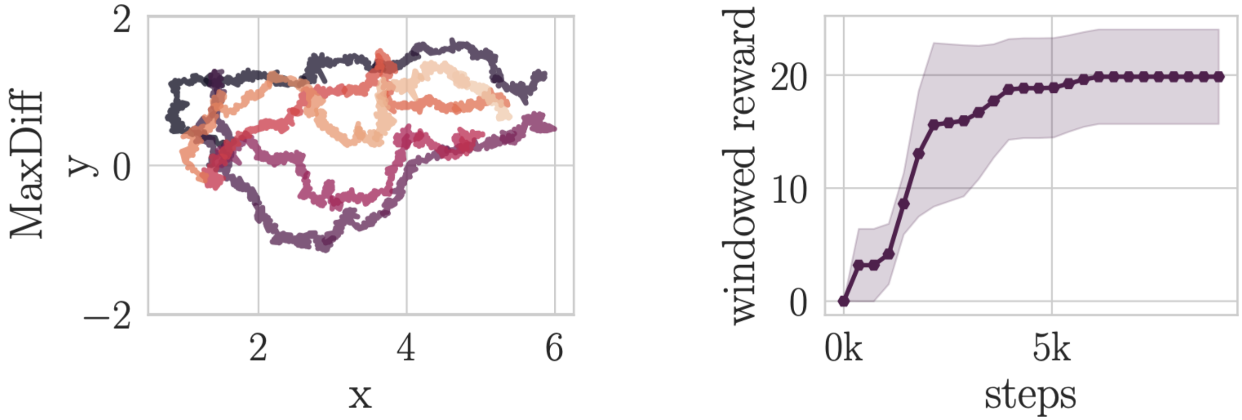
Pinosky, **Berrueta**, et al., “Physical state exploration for reinforcement learning from scratch,” *IEEE CASE*, 1715-1721 (2025)

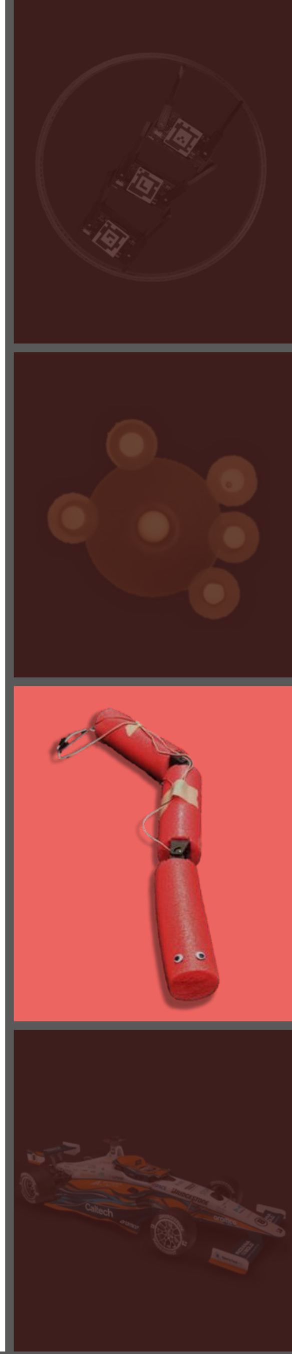

Seed 1
Seed 2
Seed 3
Seed 4
Seed 5
x
y
All videos are at 10x speed

Single-shot Simulation



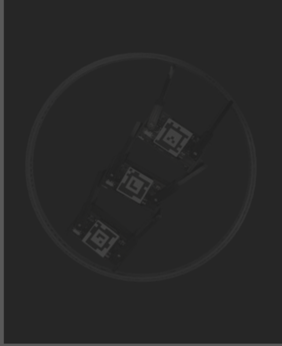
Single-shot Hardware



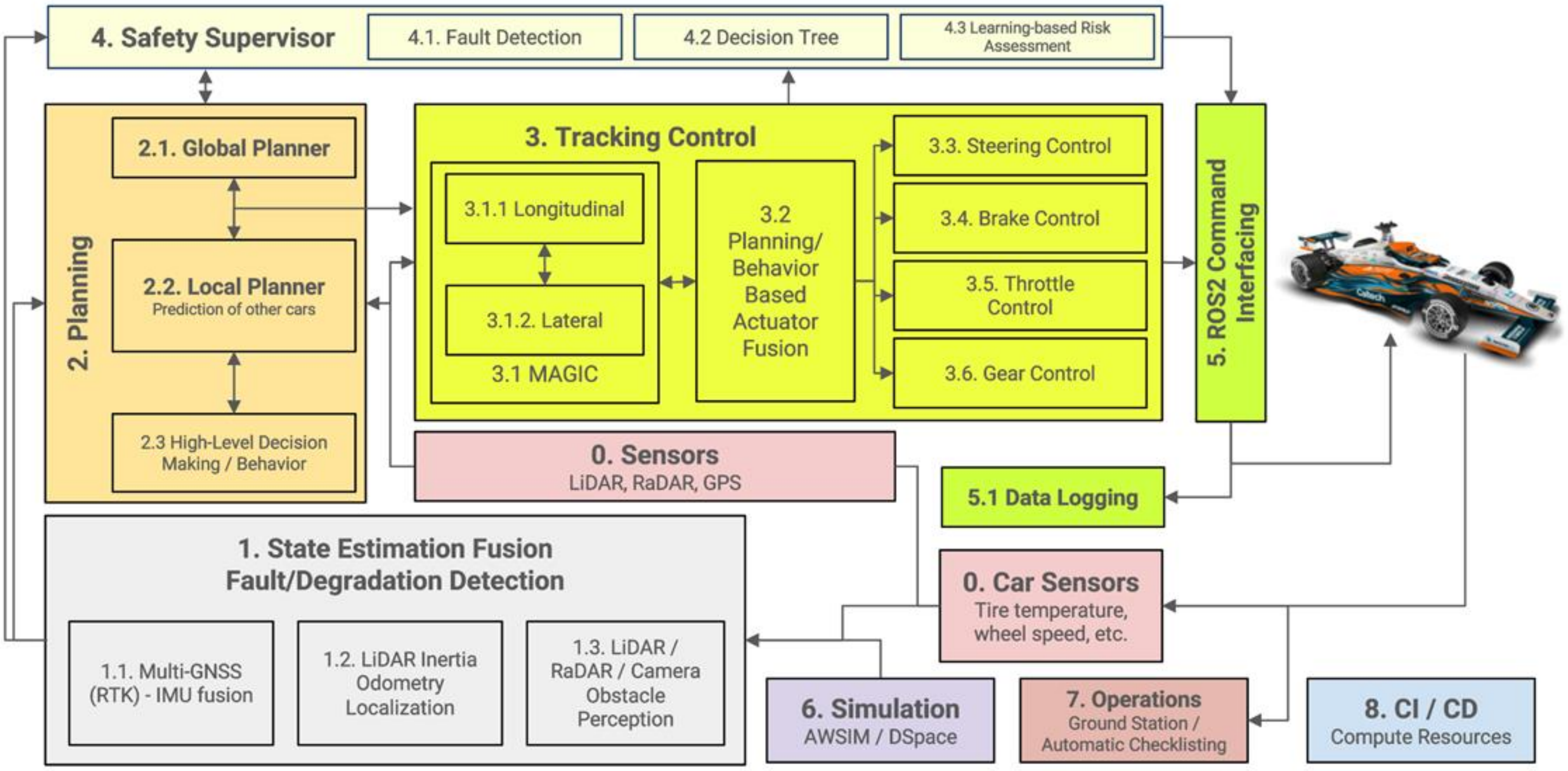


Path decorrelation for efficient online learning

Operational safety through modular design

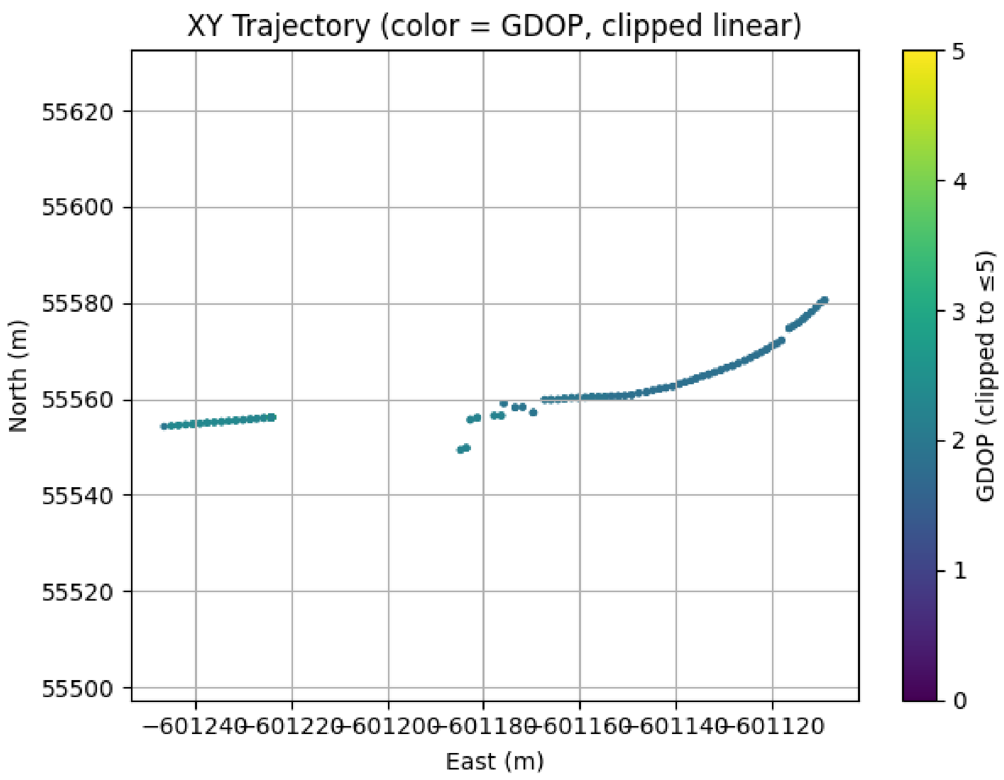


Best Lap: 1 2:37.130
LAP 2 1:34.023
Speed (MPH)
0.00
1 2 3
Steering
BRAKE
THROTTLE
RPM
2865

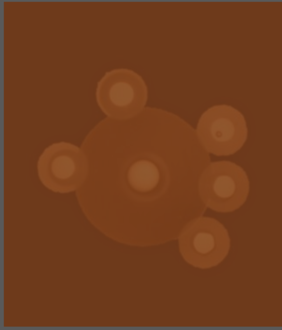


4. Safety Supervisor
4.1. Fault Detection
4.2 Decision Tree
4.3 Learning-based Risk Assessment
2. Planning
2.1. Global Planner
2.2. Local Planner
Prediction of other cars
2.3 High-Level Decision Making / Behavior
3. Tracking Control
3.1.1 Longitudinal
3.1.2. Lateral
3.1 MAGIC
3.2 Planning/ Behavior Based Actuator Fusion
3.3. Steering Control
3.4. Brake Control
3.5. Throttle Control
3.6. Gear Control
5. ROS2 Command Interfacing
0. Sensors
LiDAR, RaDAR, GPS
5.1 Data Logging
1. State Estimation Fusion
Fault/Degradation Detection
1.1. Multi-GNSS (RTK) - IMU fusion
1.2. LiDAR Inertia Odometry Localization
1.3. LiDAR / RaDAR / Camera Obstacle Perception
0. Car Sensors
Tire temperature, wheel speed, etc.
6. Simulation
AWSIM / DSpace
7. Operations
Ground Station / Automatic Checklisting
8. CI / CD
Compute Resources

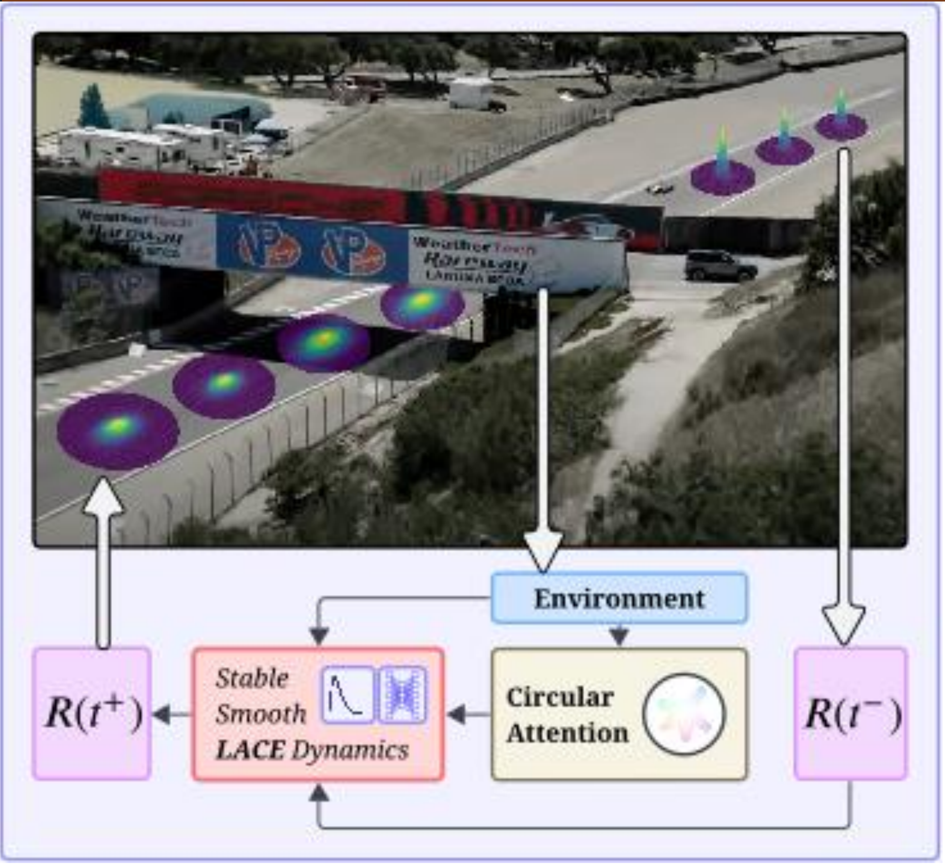
Modeling the dynamics of uncertainty

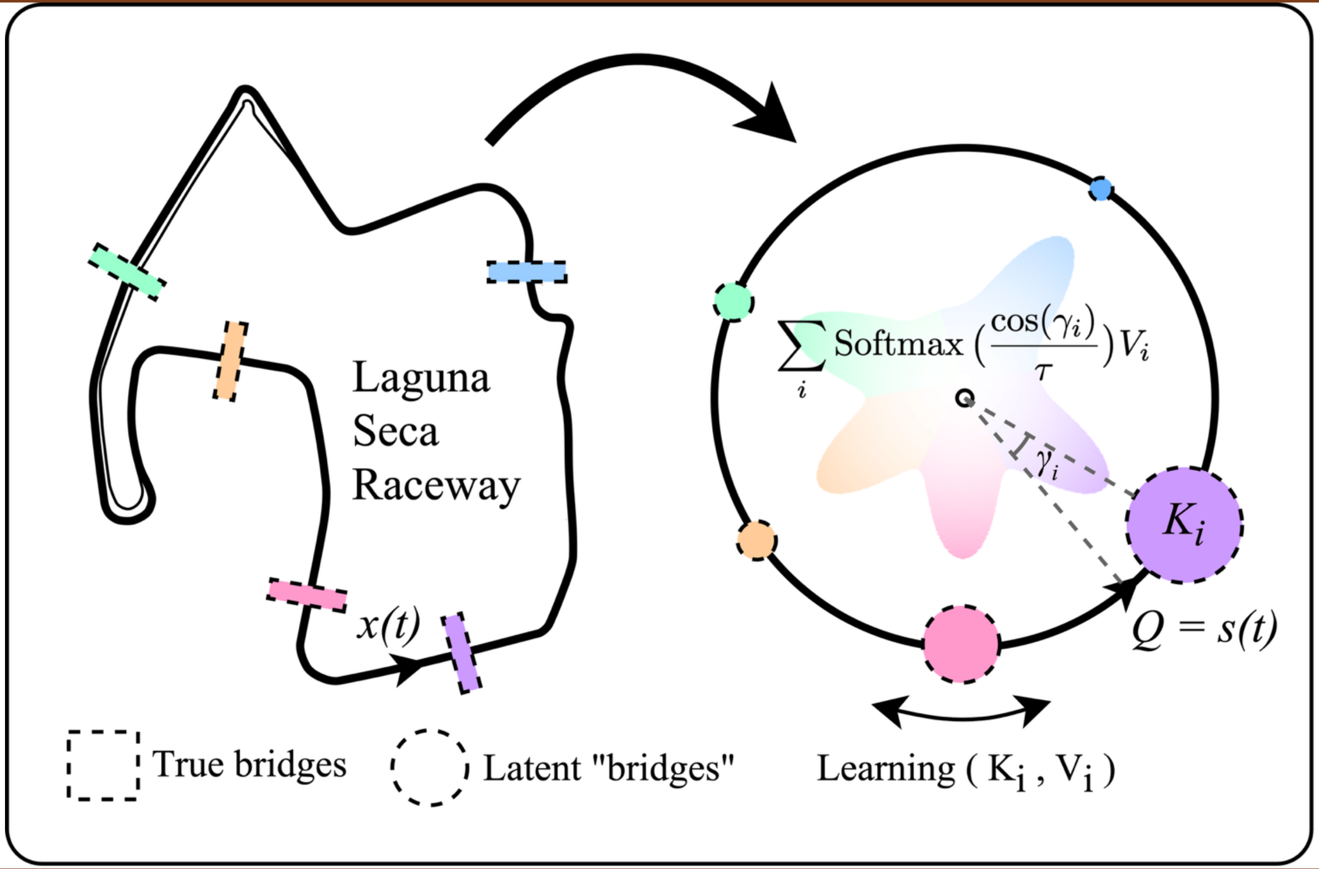


Chen, Zimmermann, **Berrueta**, Chung, "Environment-aware learning of smooth GNSS covariance dynamics for autonomous racing," *IEEE ICRA* (In Review)

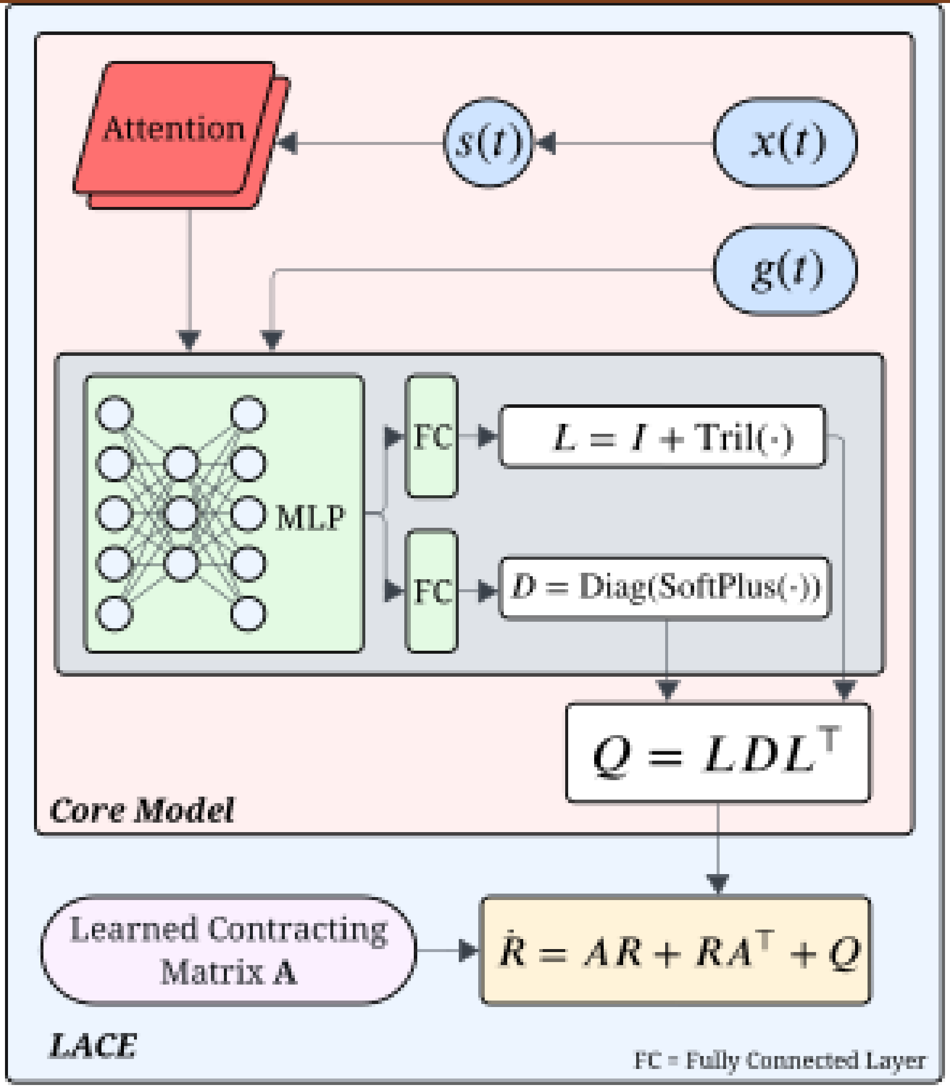



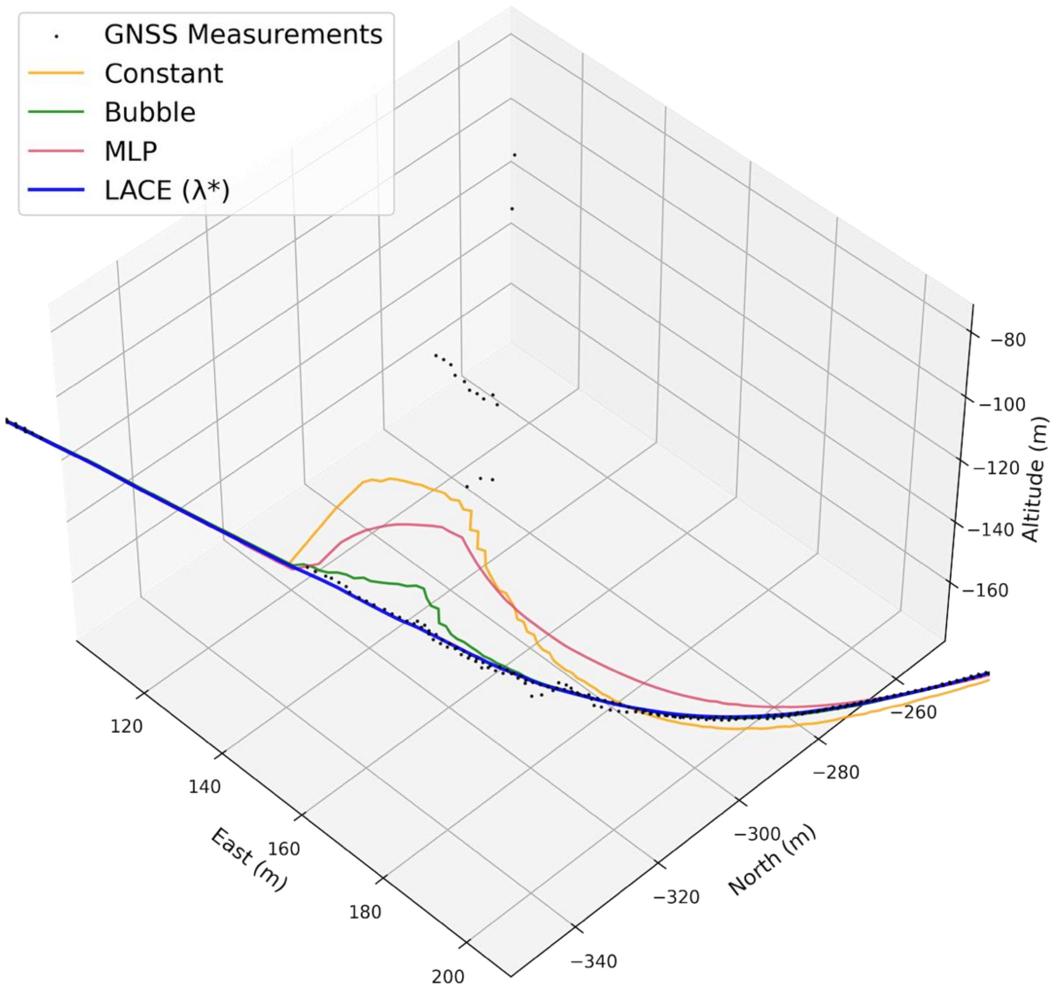
Modeling the dynamics of uncertainty



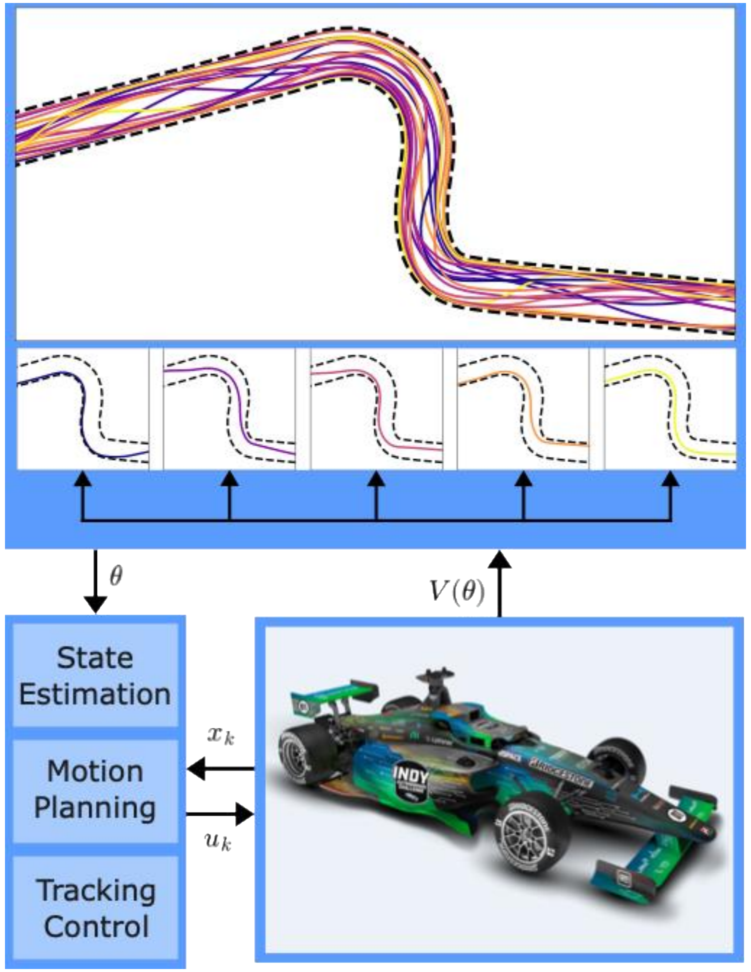


Modeling the dynamics of uncertainty





Online learning of closed-loop racelines



Engine parameters, brake bias, margin to track bounds, etc.

Deep Kernel Learning

Propose θ ⇄ Evaluate $V(\theta)$

Thompson Sampling

Lap time

Lathrop, **Berrueta**, et al, "Learning to optimize autonomous racing strategies with compound kernel exploration," *IEEE RAL* (In Preparation)

Online learning of closed-loop racelines

Experiments

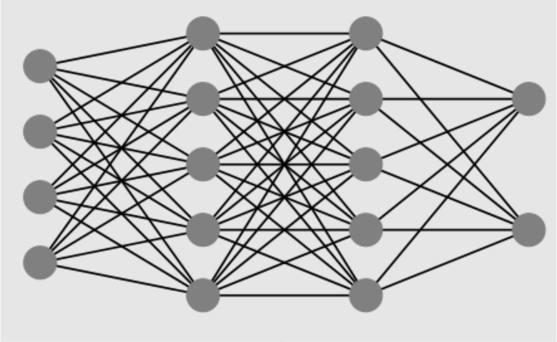
Simulations

Expert data

TRACK TITAN

θ

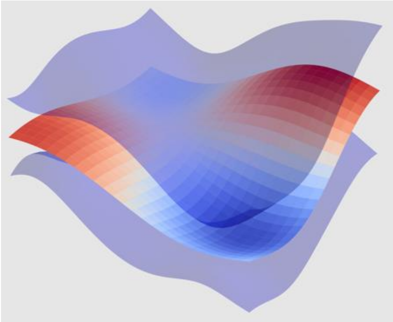
Representation learning



γ

$f_\gamma(\theta) \longrightarrow$

Deep GP with compound kernel



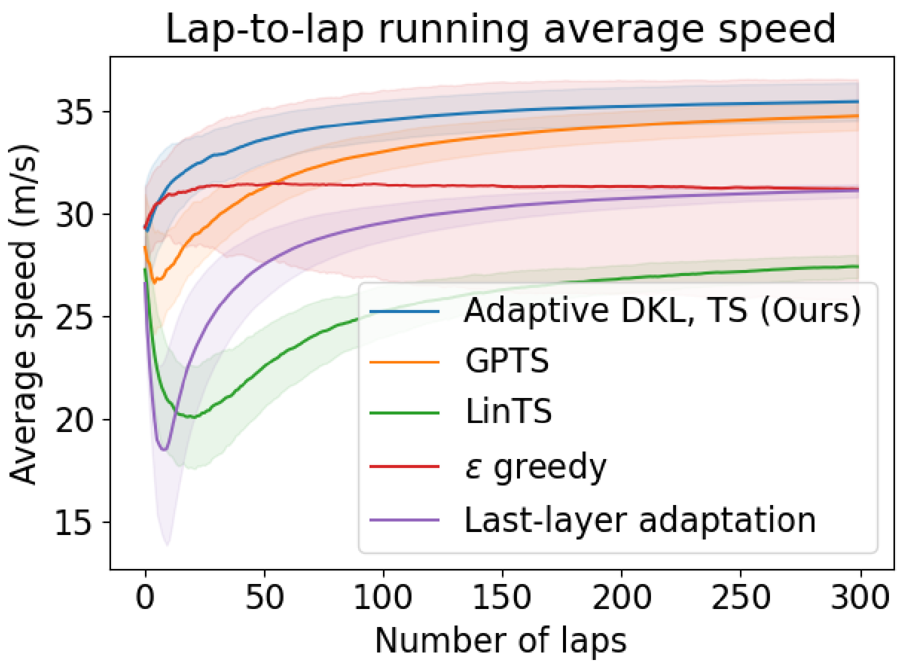
$$\hat{V}(\theta') \sim N\big(\mu(\theta'), k^*(f_\gamma(\theta), f_\gamma(\theta'))\big)$$

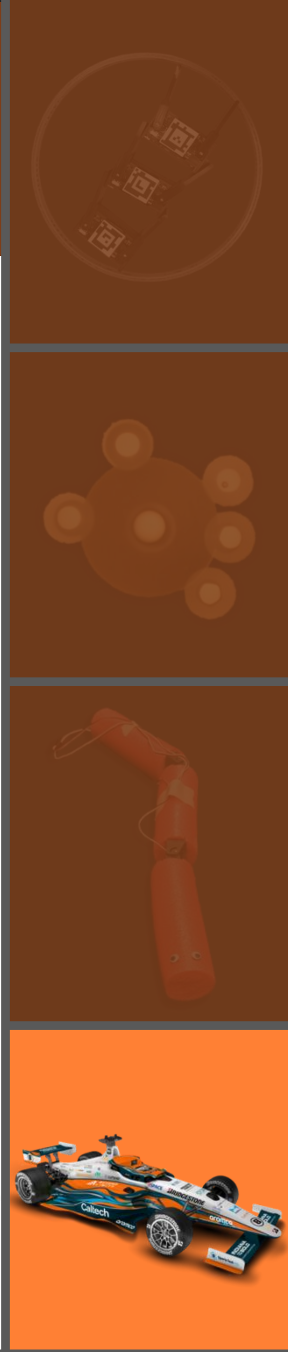
$$k^* = k(f_\gamma(\theta), f_\gamma(\theta')) + k(\theta, \theta')$$

Offline Online

Online learning of closed-loop racelines

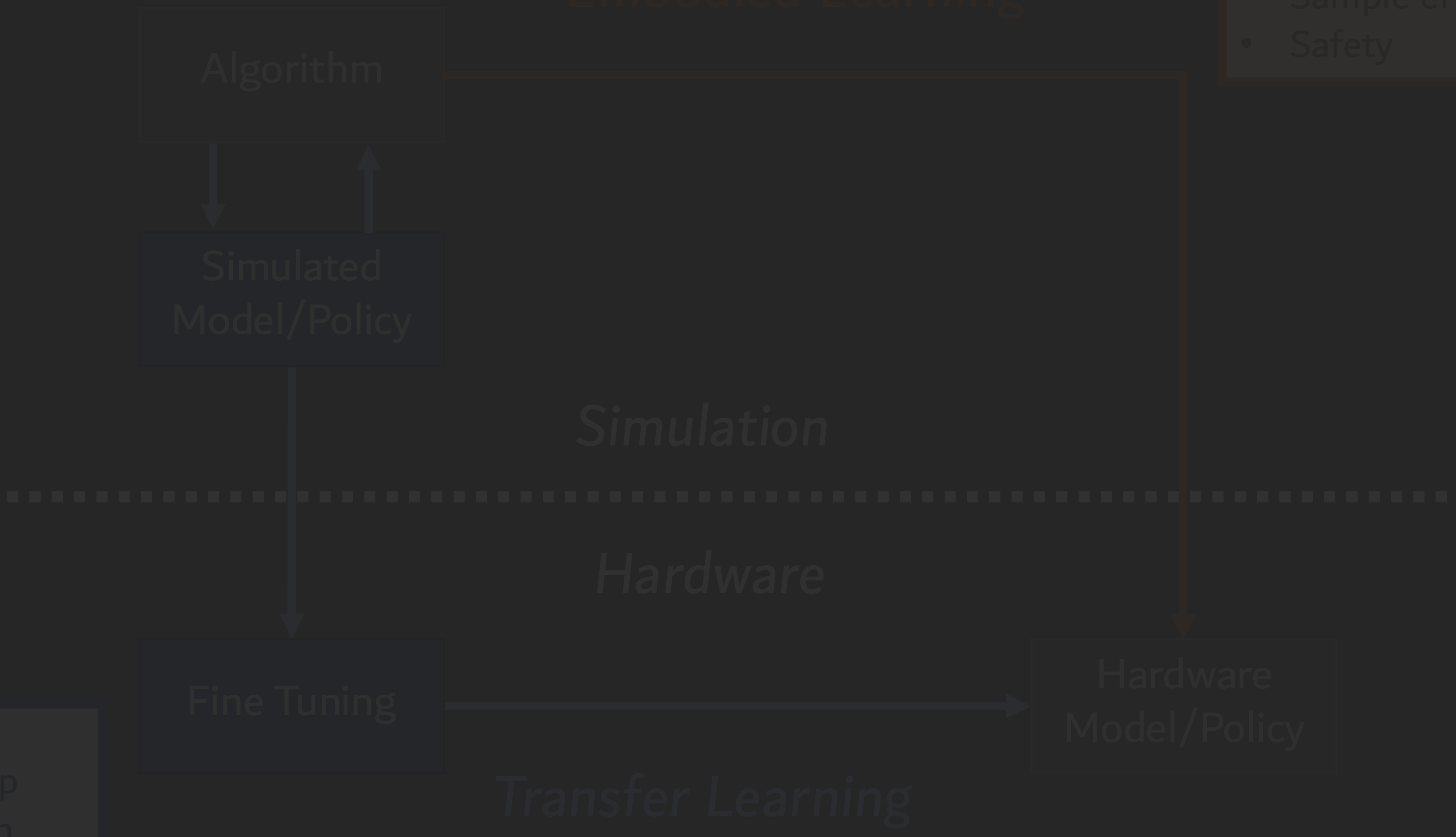
- Our approach can leverage strong priors to learn faster.
- Representation learning is key to finding better lap times.
- Since the MPC is the ultimate determinant of safety, this work is directly transferable to hardware.





Embodied Learning

Challenges:
- Data diversity
- Sample efficiency
- Safety



Challenges:
- Reality gap
- Perception
- Hardware

Embodied Learning

Challenges:
- Data diversity
- Sample efficiency
- Safety

Algorithm

Simulated Model/Policy

Simulation

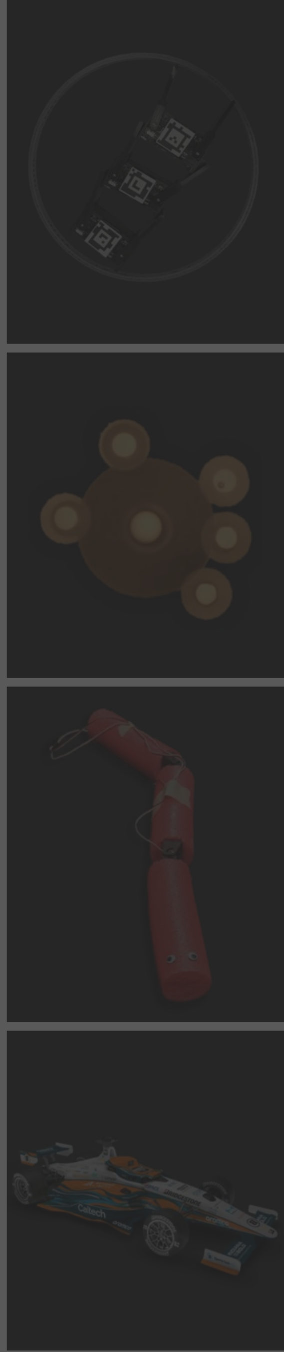
Hardware

Fine Tuning

Hardware Model/Policy

Transfer Learning

Challenges:
- Reality gap
- Perception
- Hardware



Questions?

Caltech
Continental
LUMINAR
dSPACE
KISTLER
VECTORNAV
FOSTECH
CALIAN
new eagle
verizon
indy wiring
AVI
8