Stochastic Delays in Deep Reinforcement Learning

Guest Lecture for ECE209AS

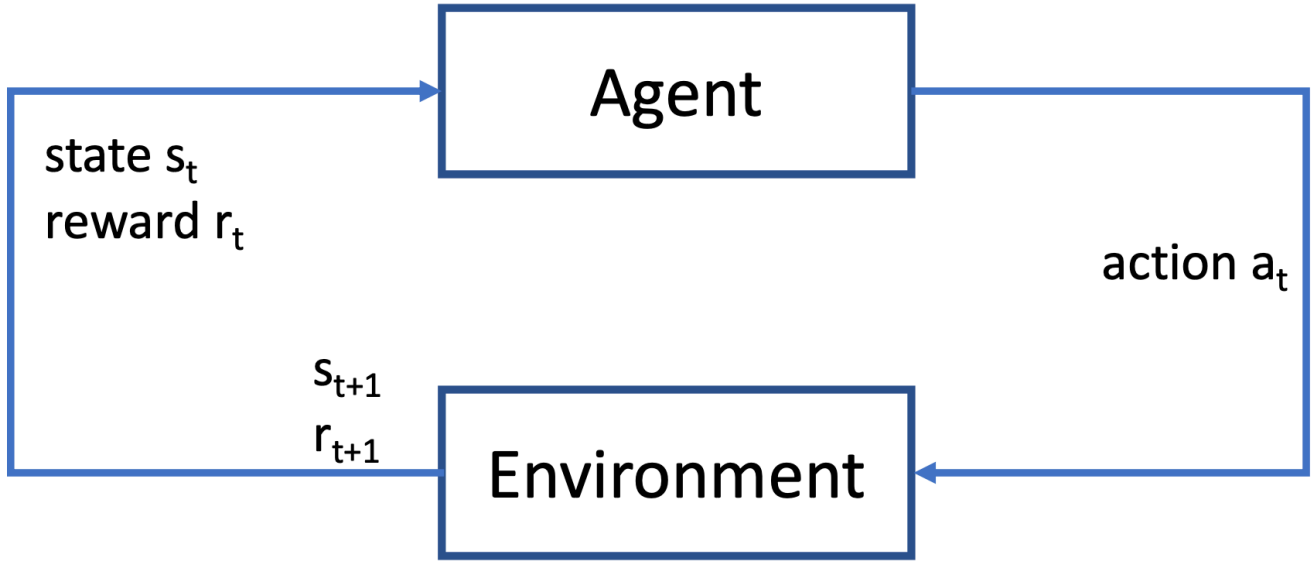
Sandeep Singh Sandha

University of California Los Angeles
25-Feb-2021

Recap from Bharathan Balaji's Lecture

Reinforcement learning is assumed to be trained and evaluated in a discrete time.

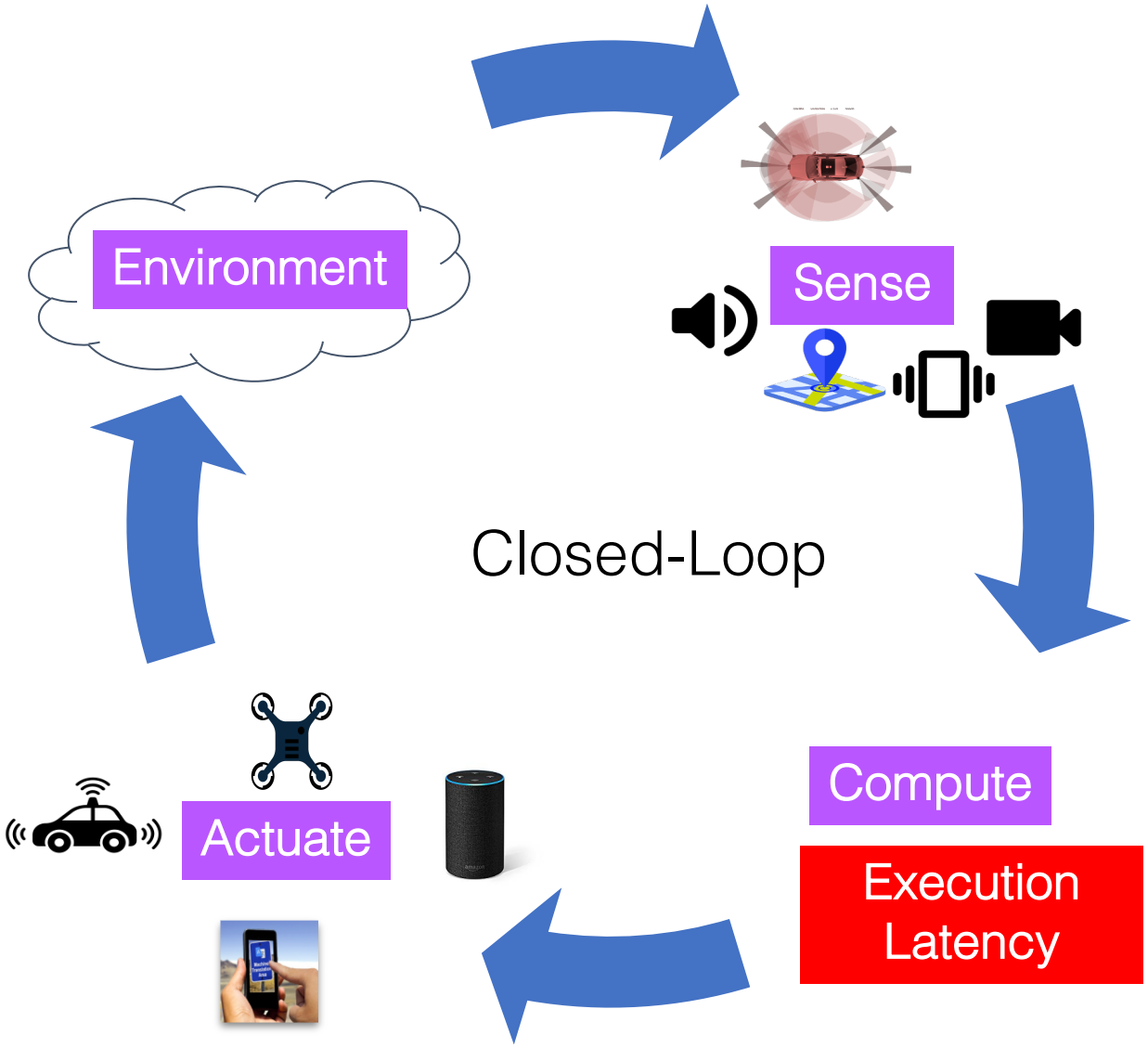
- Terminology
  - State, *s*
  - Action, *a*
  - Reward, *r*
  - Horizon, H



- Policy: π given input state *s* gives action *a*. Parametrized by θ
- Objective:

$$J(\theta) = \max_{\theta} \mathbb{E}[\sum_{t=0}^{H} r(s_t, a_t)|\pi_\theta]$$

The impact of Timing on Closed-Loop System



Outline

Delays in Deep RL

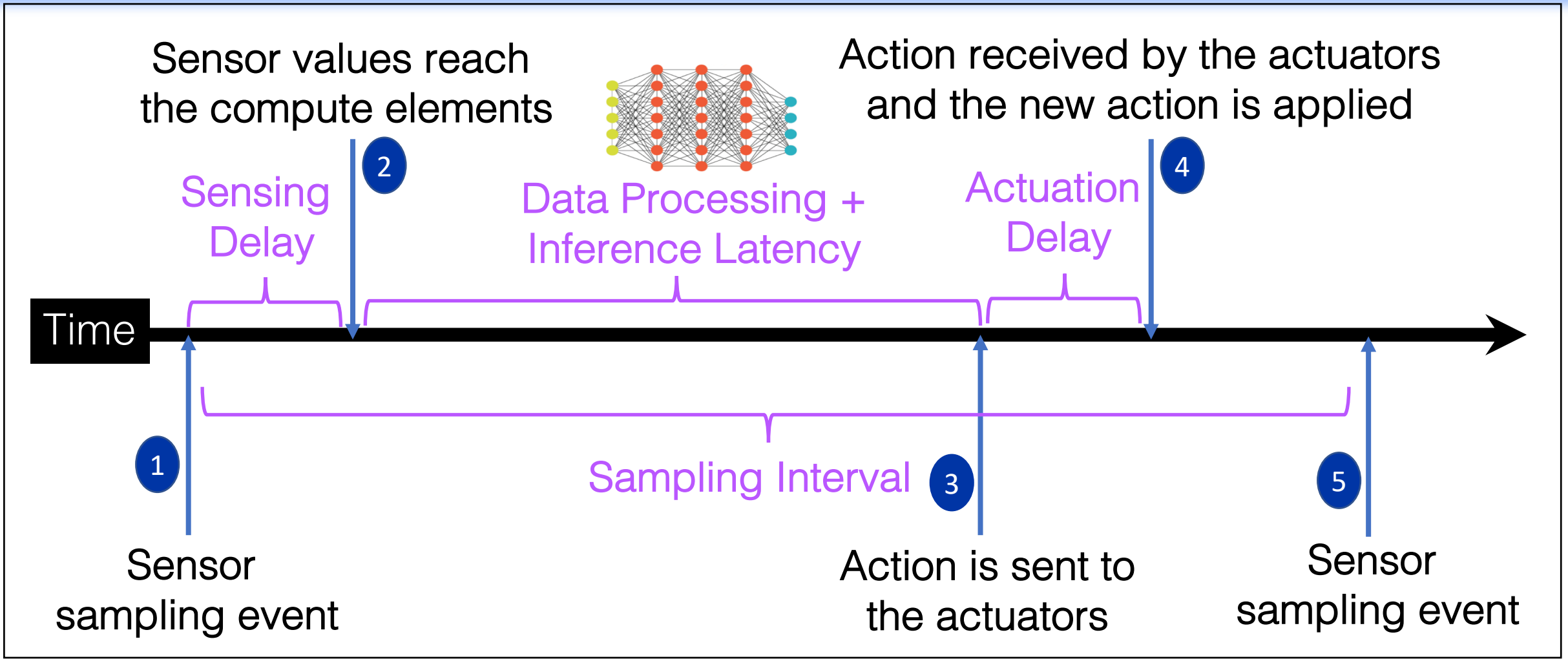
- Delay stochasticity & its impact

Related works in control systems and RL

Proposed Approach & evaluation: Time-in-State RL

Discussion

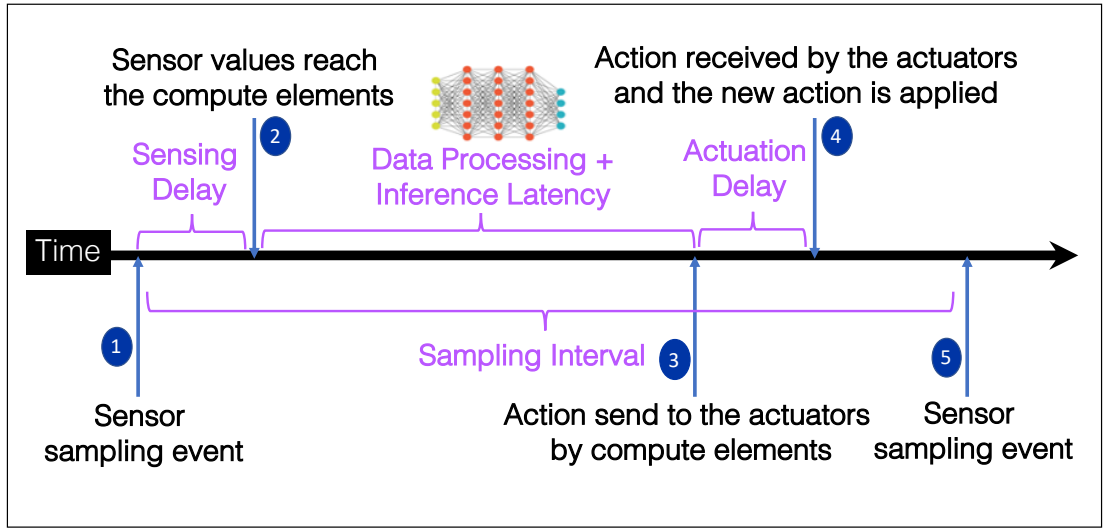
Delays in a typical Deep RL: Sensing to Actuation Pipeline



Execution Latency = Data Processing + Inference Latency

Stochastic Delays in Deep RL

Sensing to Actuation Pipeline

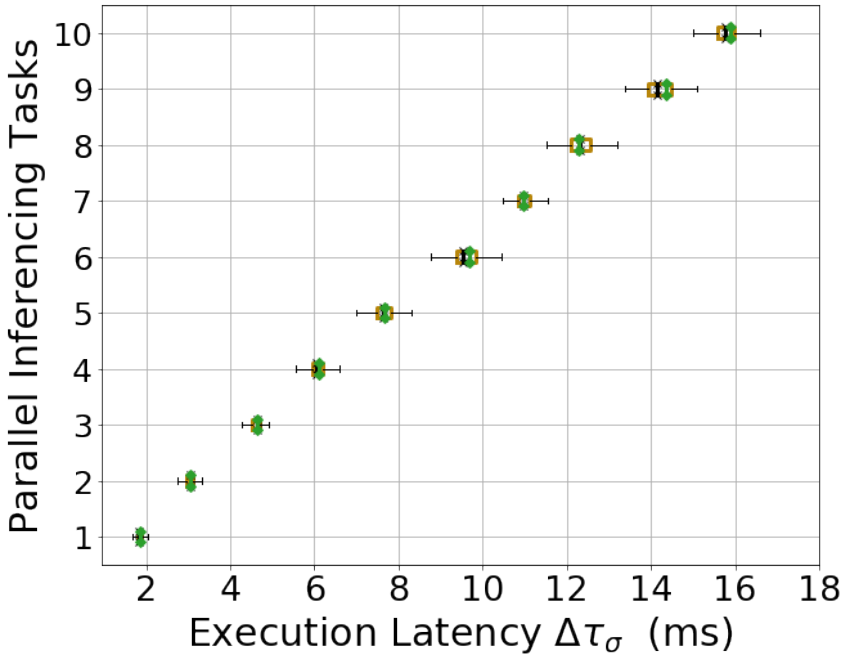


Multitude of Reasons

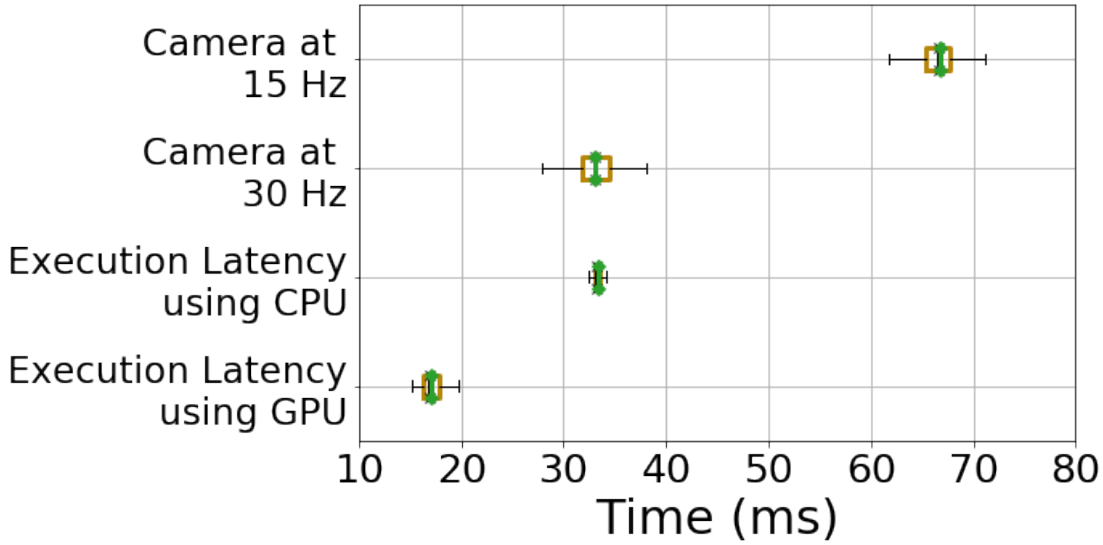
- Multitenancy
- Hardware heterogeneity
- Complexity of NN/Policy
- Thermal throttling
- Cloud & Network state

Variations in Delay can be Large

Multitenancy on Intel neural compute stick



Choices on 1/18th scale autonomous car



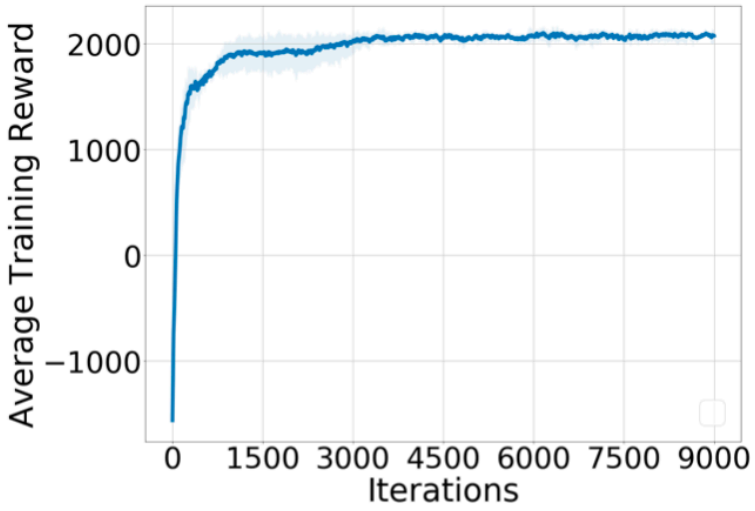
Complexity of NN on GAP8 edge device

No. of CNN layers	2	3	4
Network parameters	54k	157k	267k
Execution Latency	7.5ms	19.75ms	55.85ms

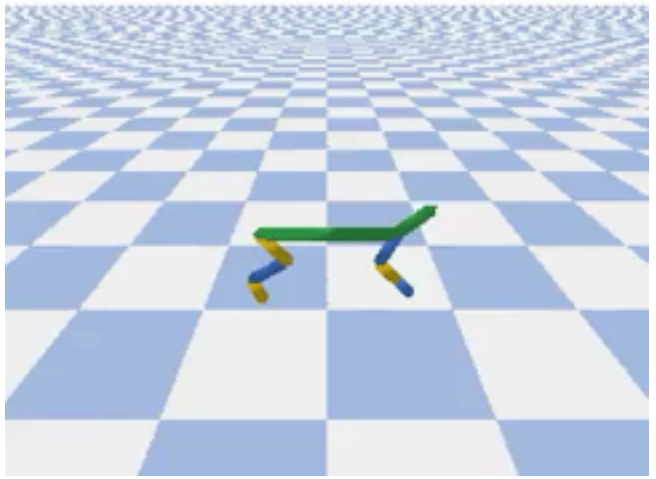
Policies can Fail due to Delay Variations!!

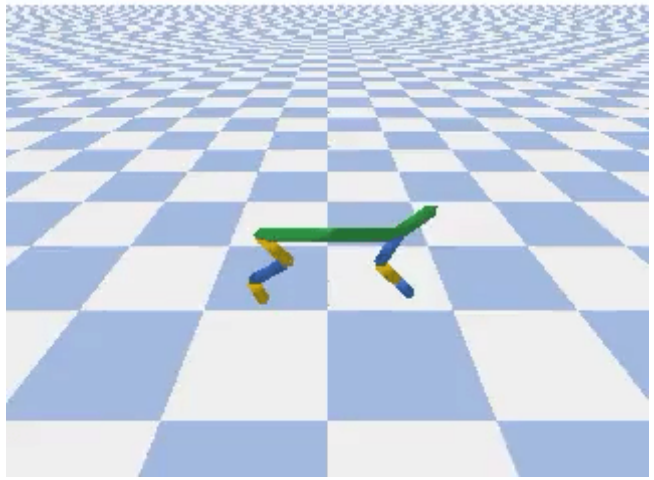
Training setting: Simulator default for HalfCheetah

- Sampling interval: 4.12 ms
- Execution latency: 0 ms
- State = 26 variables (positions, angles, velocities angular velocities) describing different joints.
- Action = 6 continuous variables (torque set for 6 joints, each between -1 and 1).
- Reward = based on the progress made by the robot.



HalfCheetah



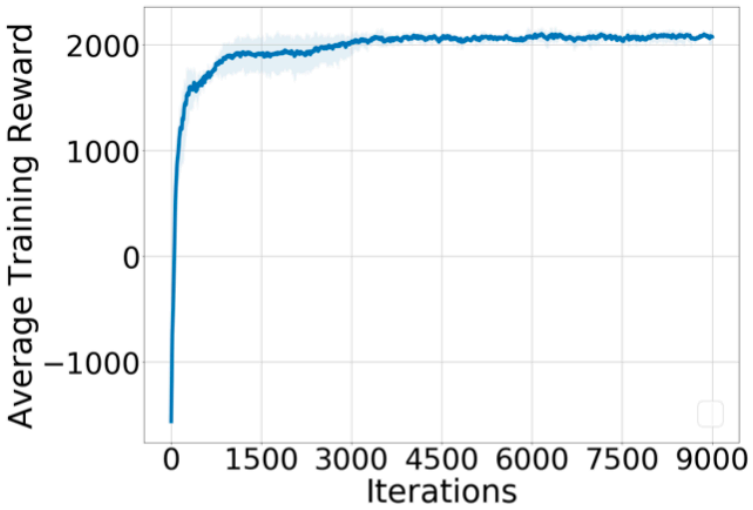


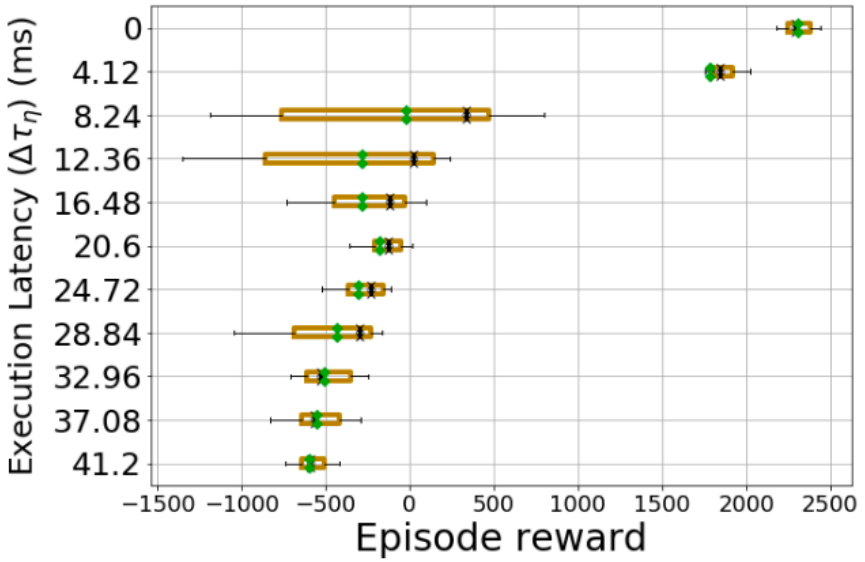
HalfCheetah robot:
https://github.com/bulletphysics/bullet3/blob/afa4fb54505fd071103b8e2e8793c38fd40f6fb6/examples/pybullet/gym/pybullet_envs/robot_locomotors.py

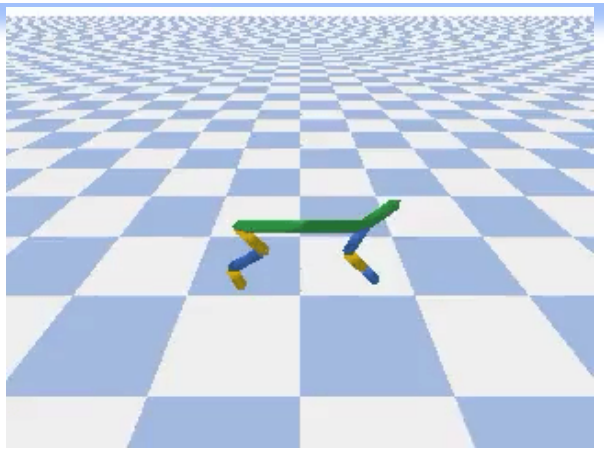
Policies can Fail due to Delay Variations!!

Testing with delays: HalfCheetah

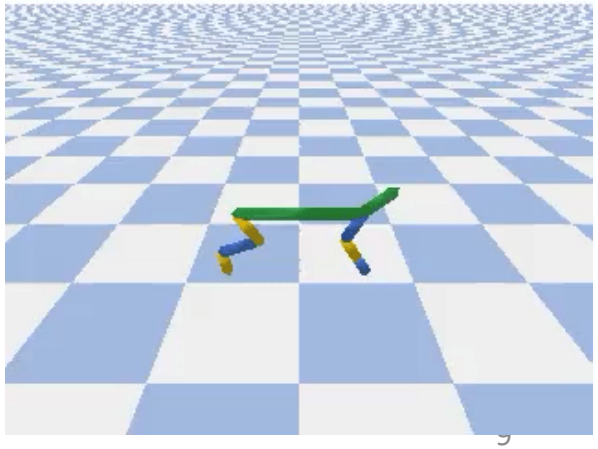
Vanilla Policy doesn't work with latencies

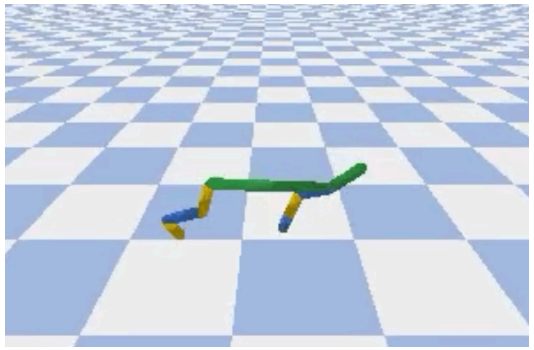




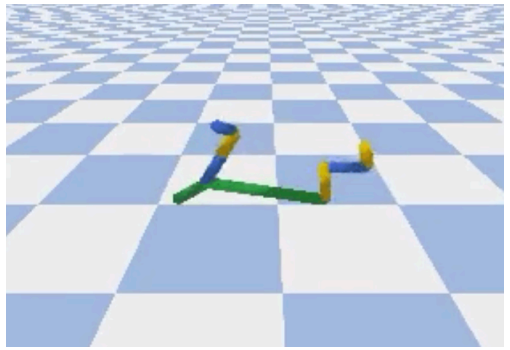


Latency = 20.6ms



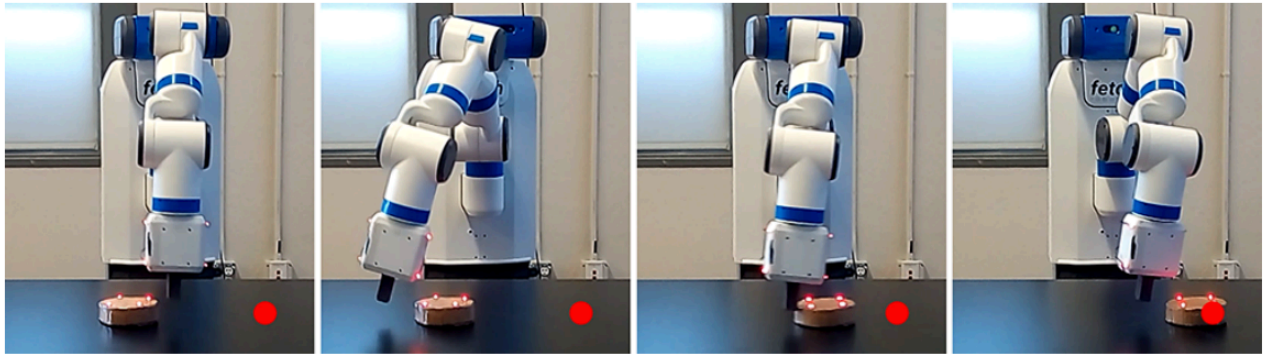


20.6 ms delay



Policies can Fail due to Delay Variations!!

Model	Success	Trials
all	0.89 ± 0.06	28
fixed action timestep	0.29 ± 0.11	17



Policy success drops from 89% to 29%.

[Peng18] Peng, Xue Bin, et al. "Sim-to-real transfer of robotic control with dynamics randomization." 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018.

Delay Handling in Control Systems

Related work	Remarks
Real-time control [Lu15, Lee16, Rajkumar16]	❖ Need carefully engineered hardware and software stack. ❖ Delays can vary on commodity platforms across different hardware, network variations, environmental factors, multitenancy, and cloud.
Control systems [Nilsson98, Bequette03, Ryu04, Liberzon15, Stefano19]	❖ Classical controllers can be compensated for fixed and stochastic delays. ❖ Delay compensation using damping components, energy-based controllers [Ryu04, Stefano19] or Lyapunov-based controllers [Nilsson98, Bequette03, Liberzon15]. ❖ The DNN-based controller trained via RL is a black box with no known mechanisms to compensate for delays.

[Nilsson98] Nilsson, Johan. "Real-time control systems with delays." (1998).
[Bequette03] Bequette, B. Wayne. Process control: modeling, design, and simulation. Prentice Hall Professional, 2003.
[Ryu04] Ryu, Jee-Hwan, et al. "Stability guaranteed control: Time domain passivity approach." IEEE Transactions on Control Systems Technology 12.6 (2004): 860-868.
[Liberzon15] Liberzon, Daniel. "Quantization, time delays, and nonlinear stabilization." IEEE Transactions on Automatic Control 51.7 (2006): 1190-1195.
[Lu15] Lu, Chenyang, et al. "Real-time wireless sensor-actuator networks for industrial cyber-physical systems." Proceedings of the IEEE 104.5 (2015): 1013-1024.
[Lee16] Lee, Edward Ashford, and Sanjit A. Seshia. Introduction to embedded systems: A cyber-physical systems approach. Mit Press, 2016.
[Rajkumar16] Rajkumar, Raj, Dionisio De Niz, and Mark Klein. Cyber-physical systems. Addison-Wesley Professional, 2016.
[Stefano19] De Stefano, Marco, et al. "Time-delay Compensation Using Energy Tank for Satellite Dynamics Robotic Simulators." IEEE International Conference on Intelligent Robots and Systems. 2019.

Delay Handling in RL

Related work	Remarks
Xie et. al [Xie18], Mahmood et. Al.[Mahmood18]	❖ Observe variable delays have detrimental impact on the deep RL controllers. ❖ Doesn't propose any solution.
Ramstedt et. al [Ramstedt19], Chen et. Al [Chen20]	❖ Modify state with the past action. ❖ Assumes fixed discrete delay during training. ❖ Cannot handle continuous delay variations.
Domain Randomization [Peng18, Tan18, Andrychowicz20]	❖ Shows success of domain randomization to handle variable delays for real robots. ❖ We use domain randomization as a baseline.

[Xie18] Xie, Zhaoming, et al. "Feedback control for cassie with deep reinforcement learning." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.
[Mahmood18] Mahmood, A. Rupam, et al. "Setting up a reinforcement learning task with a real-world robot." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.
[Tan18] Tan, Jie, et al. "Sim-to-real: Learning agile locomotion for quadruped robots." arXiv preprint arXiv:1804.10332 (2018).
[Andrychowicz20] Andrychowicz, OpenAI: Marcin, et al. "Learning dexterous in-hand manipulation." The International Journal of Robotics Research 39.1 (2020): 3-20.
[Peng18] Peng, Xue Bin, et al. "Sim-to-real transfer of robotic control with dynamics randomization." 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018.
[Ramstedt19] Ramstedt, Simon, and Chris Pal. "Real-time reinforcement learning." Advances in Neural Information Processing Systems. 2019.
[Chen20] Chen, Baiming, et al. "Delay-Aware Model-Based Reinforcement Learning for Continuous Control." arXiv preprint arXiv:2005.05440 (2020).

Delay Variations impact the State Transitions

$$p(s_{t+1} | s_t, a_t)$$

State transition model gives us the consequence of action taken by agent.

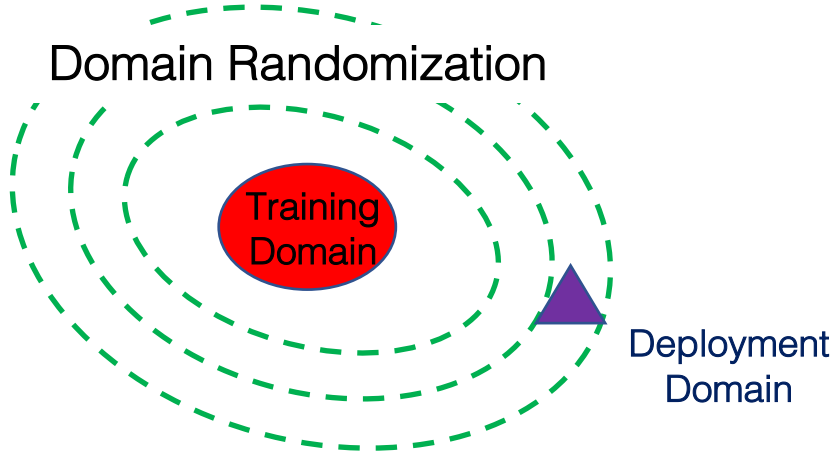
Delays impact the consequence of action, thereby results in distribution mismatch between domain.

[Peng18] Peng, Xue Bin, et al. "Sim-to-real transfer of robotic control with dynamics randomization." 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018.

Existing Approach: Domain Randomization

Domain randomization:
Train a policy over a distribution of parameters.

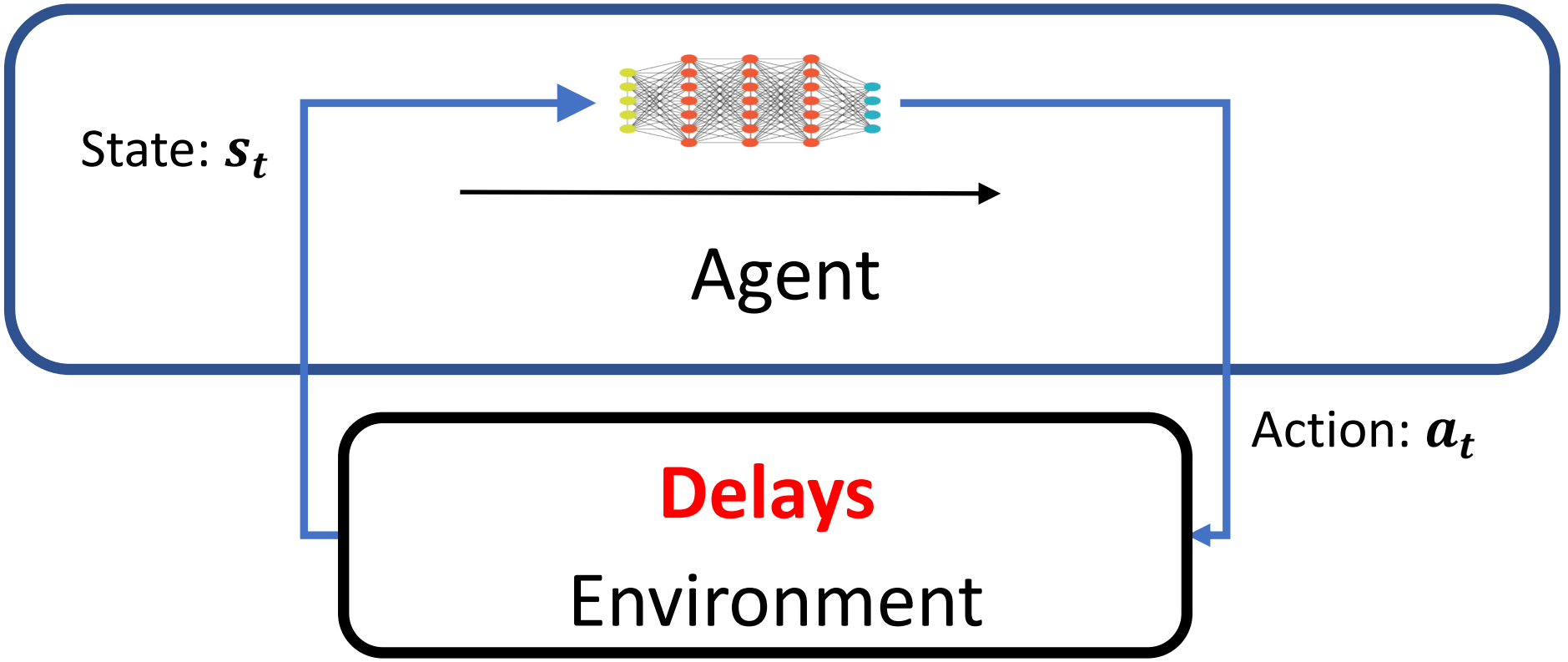
Delays: Expose policy to variable delays during training.



We use domain randomization as our baseline

Subsume Delays in the Environment

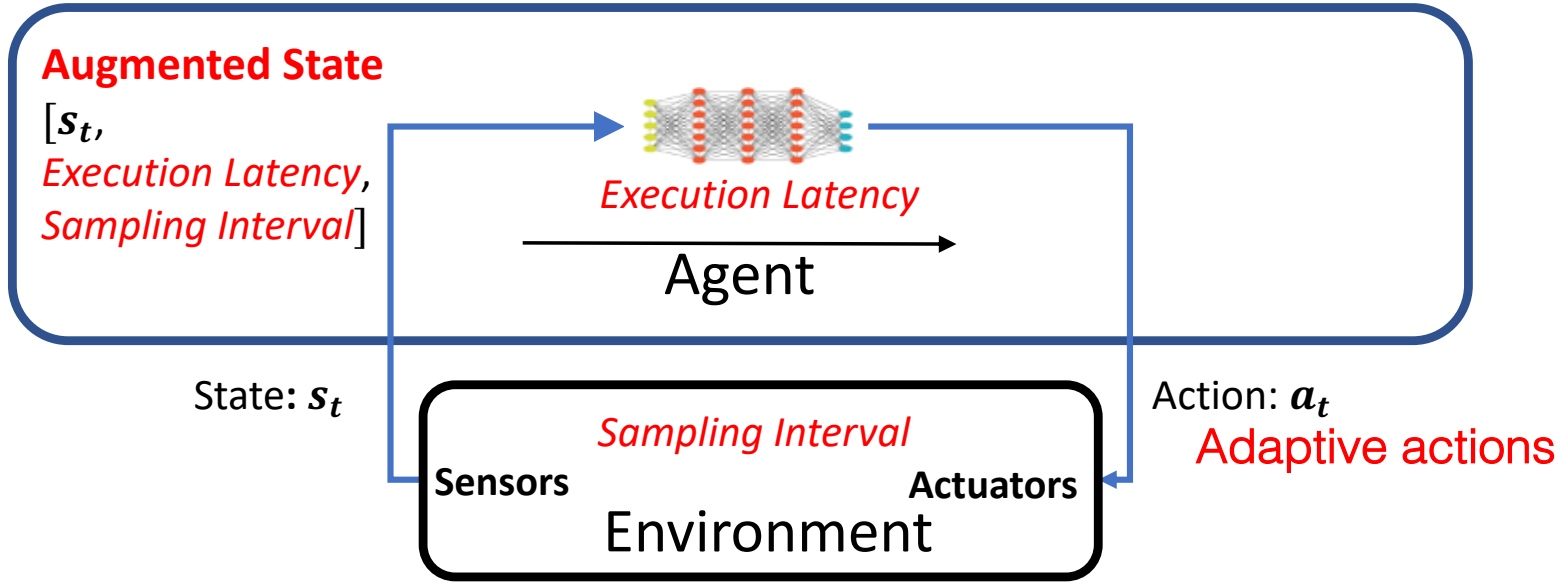
Or if variable delays are present inherently in the training environment



Equivalent to the domain randomization approach.

Our Approach: Time-in-State RL

Application can monitor and adapt to the continuous changes in the temporal properties at runtime.

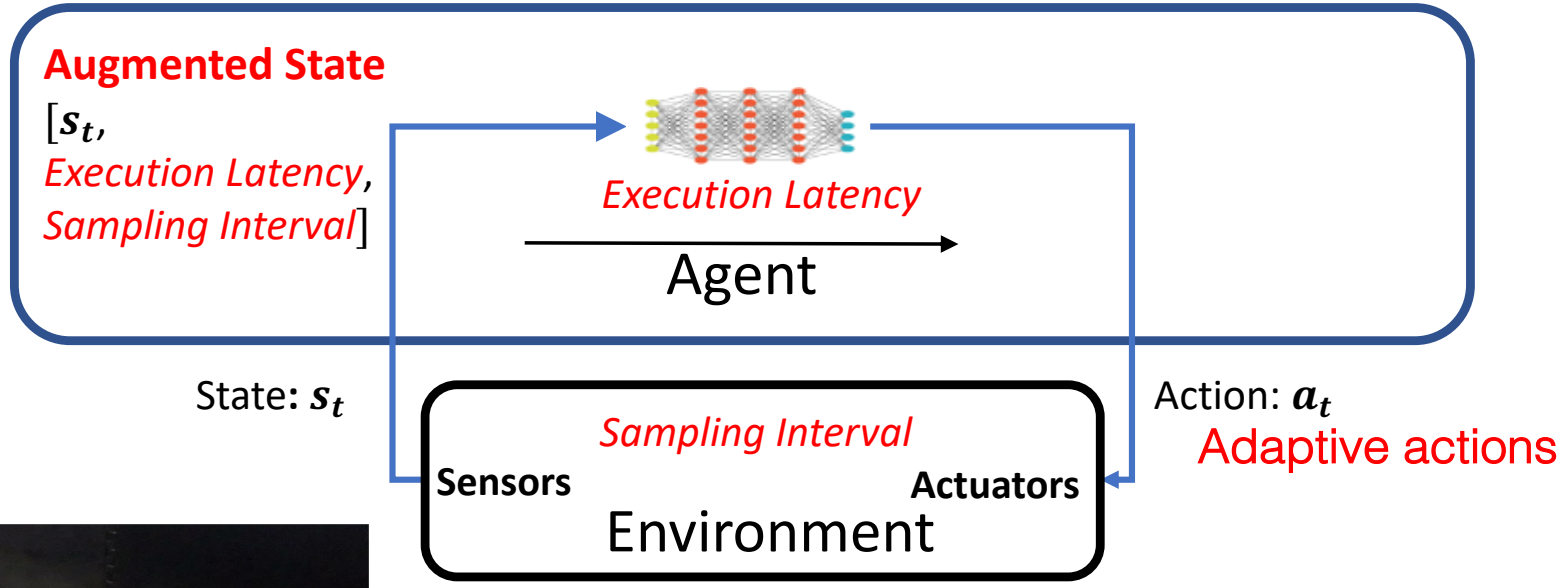


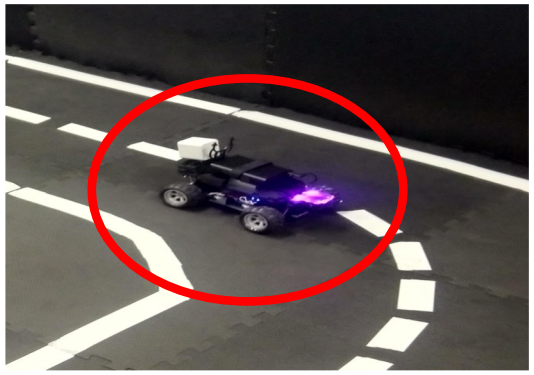
Time-in-State:

- Domain randomization (vary delays during training) +
- Include delays as part of state.

Our Approach: Time-in-State RL

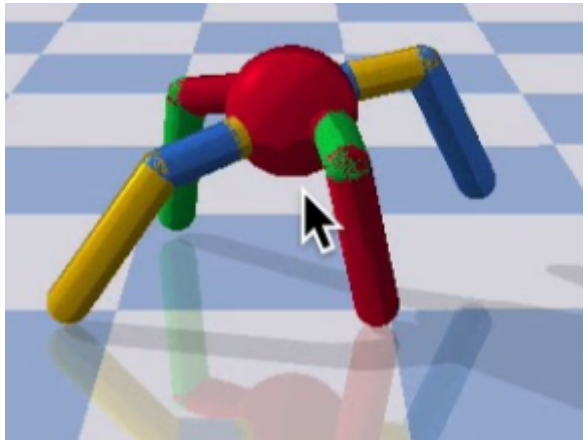
Application can monitor and adapt to the continuous changes in the temporal properties at runtime.



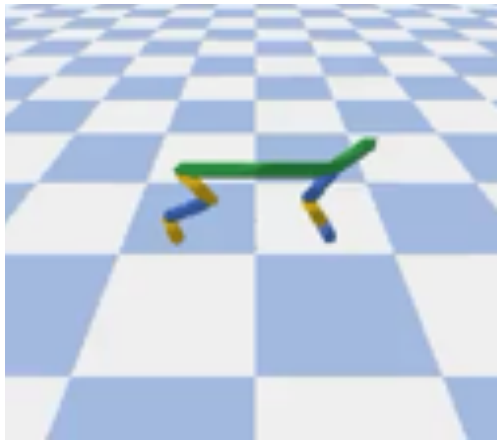


Action adaptation driven by time and state both.
E.g., At turns car can slow down significantly if delays are higher.

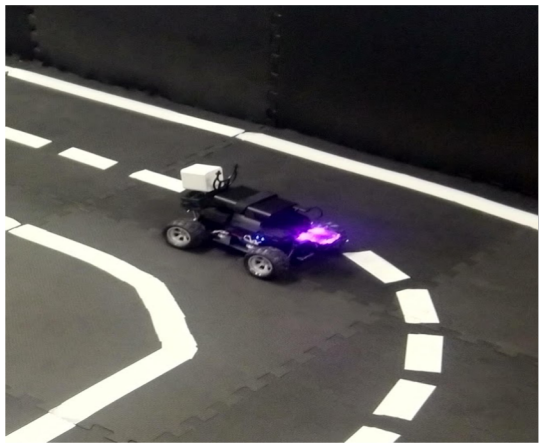
Evaluation of Time-in-State RL



Ant task



HalfCheetah task



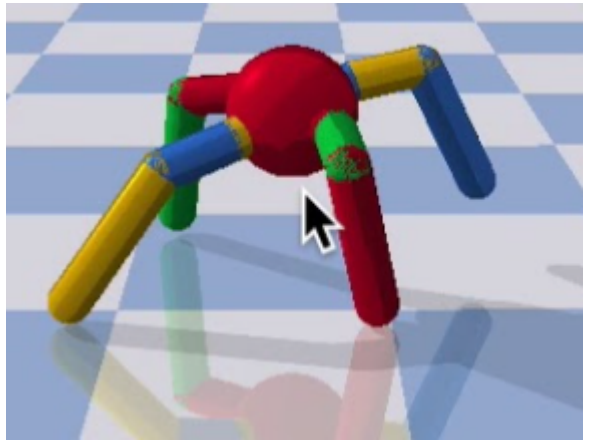
Autonomous car

Low dimensional: PyBullet simulator

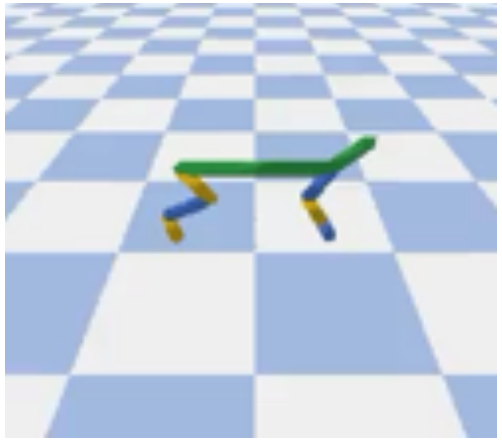
High dimensional: Gazebo simulator & 1/18th scale car

[Balaji19] Balaji, Bharathan, et al. "DeepRacer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning." arXiv preprint arXiv:1911.01562 (2019).

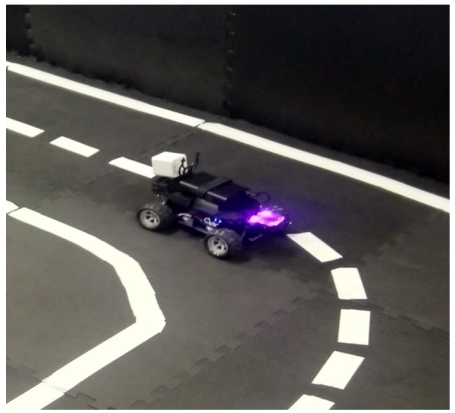
Evaluation of Time-in-State RL



Ant task



HalfCheetah task



Autonomous car

Low Dimensional: Ant and HalfCheetah in Pybullet simulator.

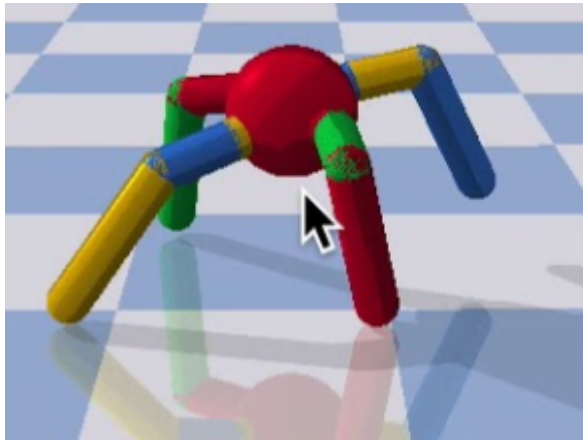
- Delays directly added to state.
- Reward = *Progress made by robot*

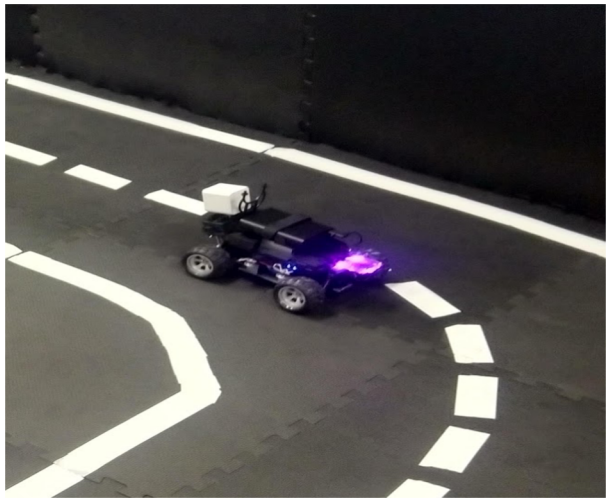
High Dimensional: Autonomous car in Gazebo simulator & on a real car.

- Fuse delays as another modality: Multimodal fusion
- Reward = *Stay close to the center-line*

Ant and Deepracer

- Ant
  - State space: 28 variables.
  - Action: 8 variables (torque control of 8 joints).
  - Reward: progress made by the robot.
- Deepracer
  - State space: 120 X 160 image
    track width
  - Action: 15 discrete (3 speeds, 5 angles).
    - Speed: 1.2, 1.5, 1.8 m/s
    - Angles: 30, 15, 0, -15, -30
  - Reward: Distance from the center line.
    - 1.0 max reward (Car-center on the track center-line)
    - 0.001 (Car center - track center > 0.4% of track width).
    - -30 (Crash: when outside the track, or *Car center - track center > track width*)

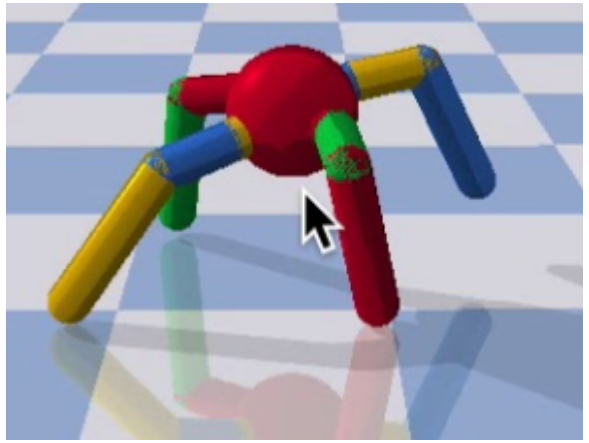




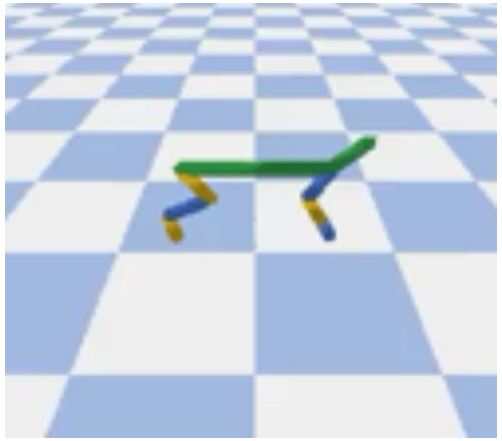
https://pybullet.org/

[Balaji19] Balaji, Bharathan, et al. "DeepRacer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning." arXiv preprint arXiv:1911.01562 (2019).

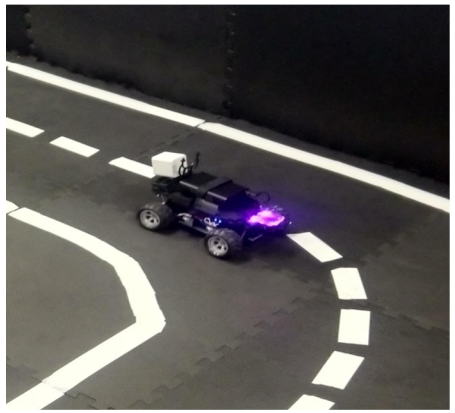
Evaluation of Time-in-State RL



Ant task



HalfCheetah task



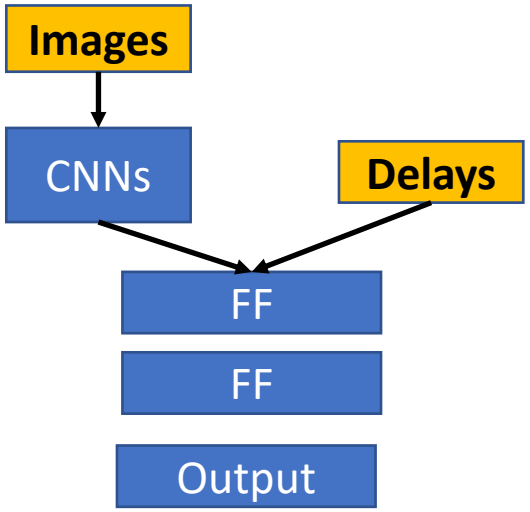
Autonomous car

Low Dimensional: Ant and HalfCheetah in Pybullet simulator.

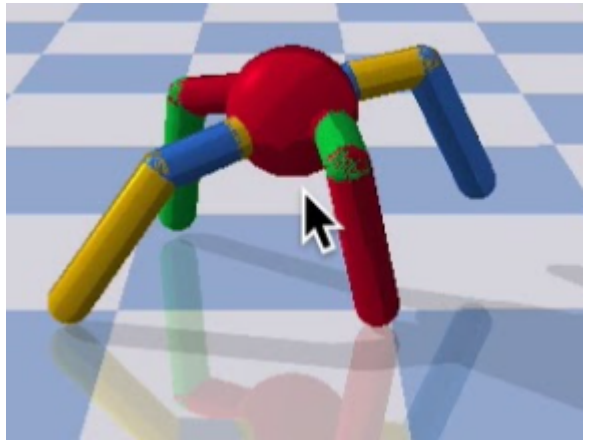
- Delays directly added to state.
- Reward = *Progress made by robot*

High Dimensional: Autonomous car in Gazebo simulator & on a real car.

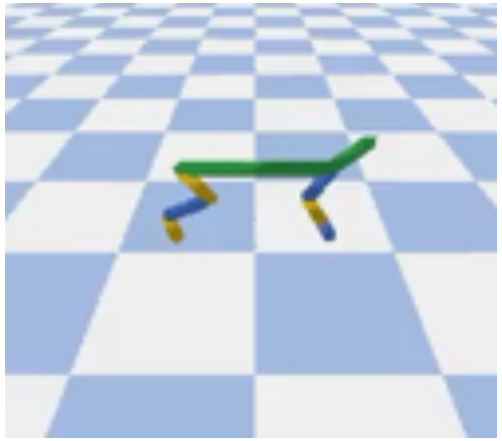
- Fuse delays as another modality: Multimodal fusion
- Reward = *Stay close to the center-line*



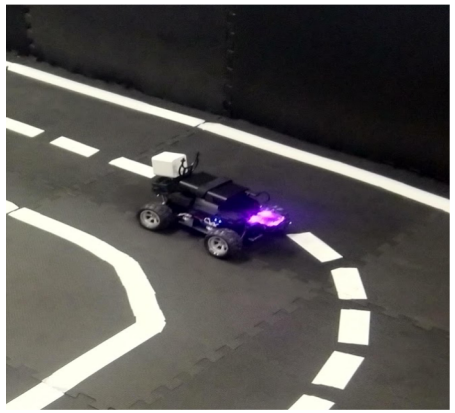
Default Delays in Simulators



Ant task



HalfCheetah task



Autonomous car

Low Dimensional: Ant and HalfCheetah in Pybullet simulator.

- Execution latency: 0 ms
- Sampling interval: 4.12 ms

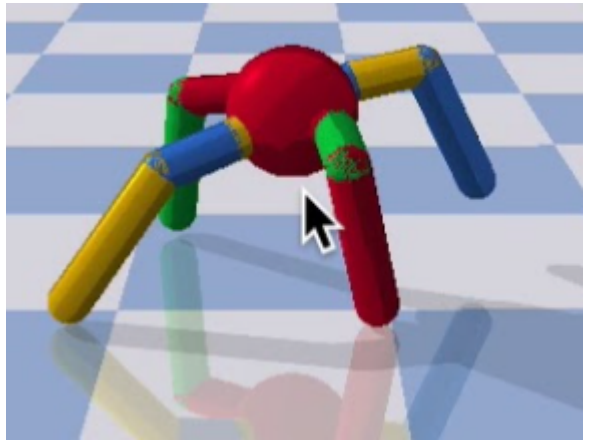
High Dimensional: Autonomous car in Gazebo simulator & on a real car.

- Execution latency: ~10ms (depends on server machine)
- Sampling interval: 66ms (camera is running at 15hz)

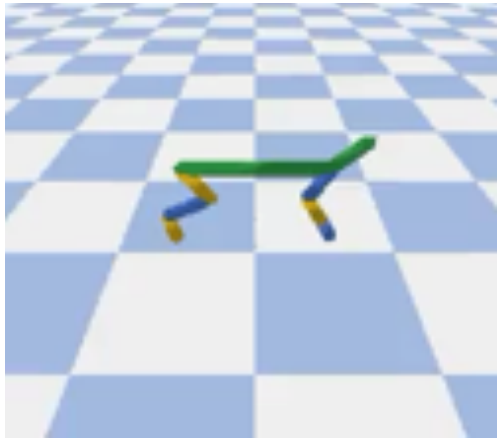
https://pybullet.org/

[Balaji19] Balaji, Bharathan, et al. "DeepRacer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning." arXiv preprint arXiv:1911.01562 (2019).

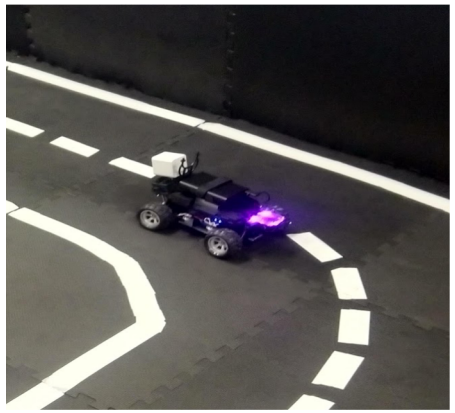
Delay Variations



Ant task



HalfCheetah task



Autonomous car

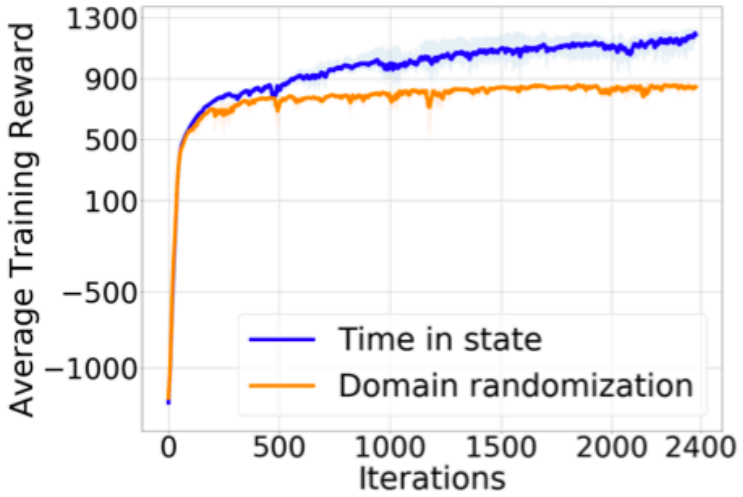
Low Dimensional: Ant and HalfCheetah in Pybullet simulator.

- Execution latency: (0 ms – 41.2 ms)
- Sampling interval: (4.12 ms – 41.2 ms), max(4.12 ms, Ex. latency)
- Jitter of 4.12 ms.

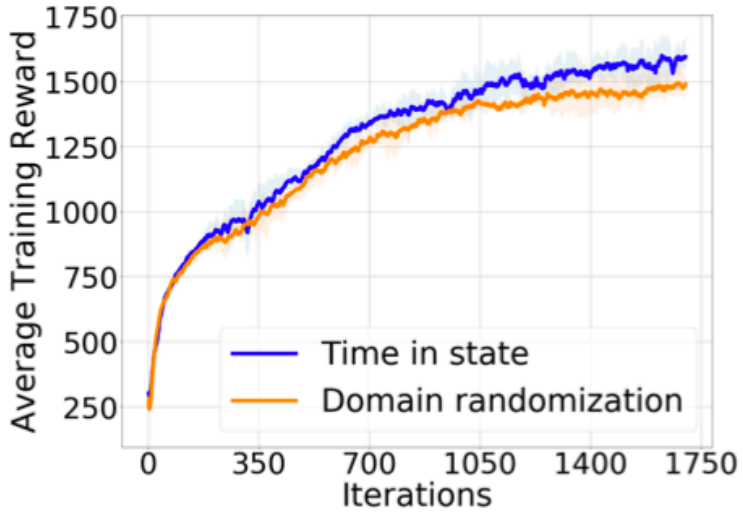
High Dimensional: Autonomous car in Gazebo simulator & on a real car.

- Execution latency: (20 ms – 120 ms)
- Sampling interval: (33 ms – 120 ms), max(33 ms, Ex. latency)
- Jitter of ~5 ms.

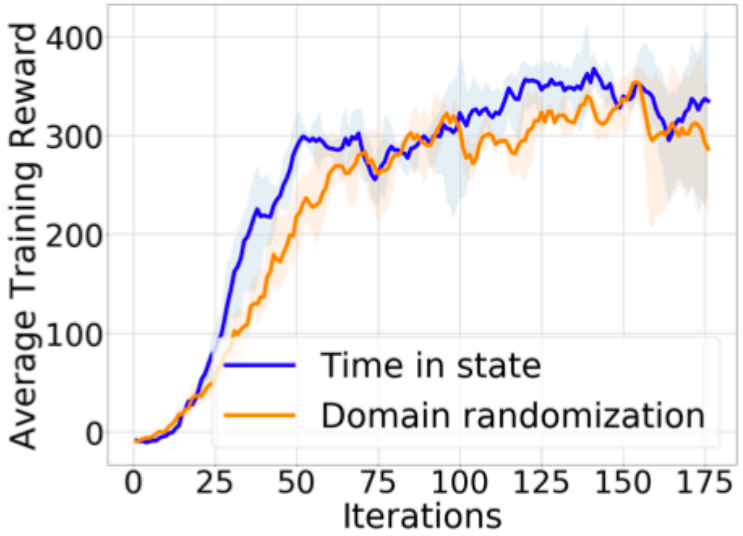
Learning Curves of Policies



HalfCheetah task



Ant task



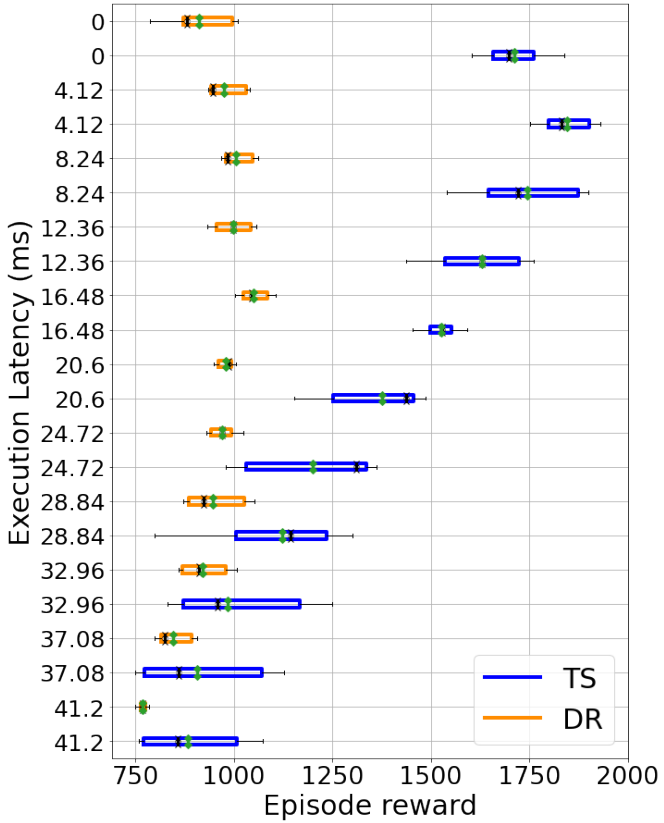
Autonomous car

Time-in-State policies achieve higher training reward.

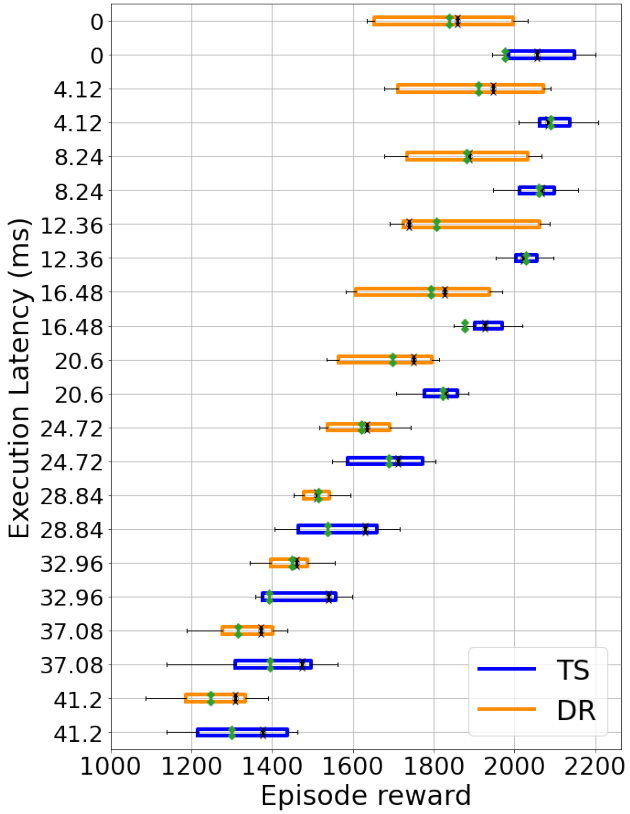
Training Algorithm: Proximal Policy Optimization

Results: TS vs DR in PyBullet Simulator

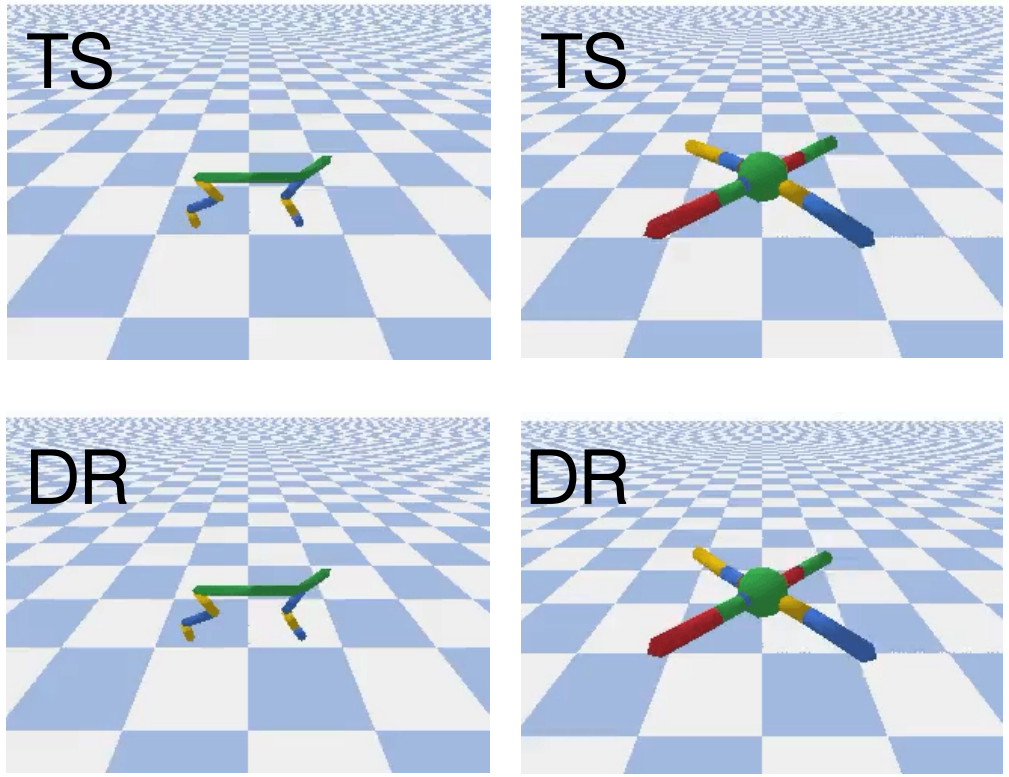
HalfCheetah task



Ant task

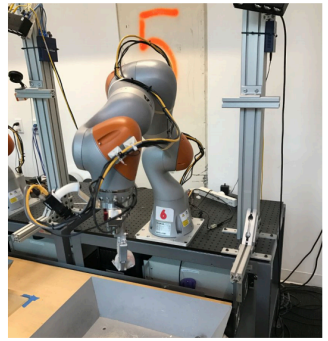


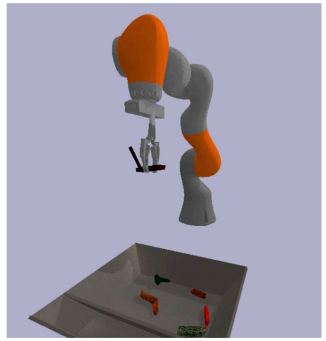
Ex. Latency = 20.6 ms



Policies for the Real Robots

- **Training on real robot [**Bai19**]**: 7 KUKA robots running for 2-3 months.
- 608,000 real-world grasps to achieve best performance.

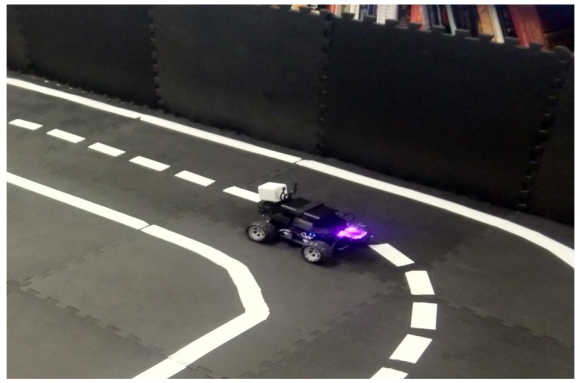


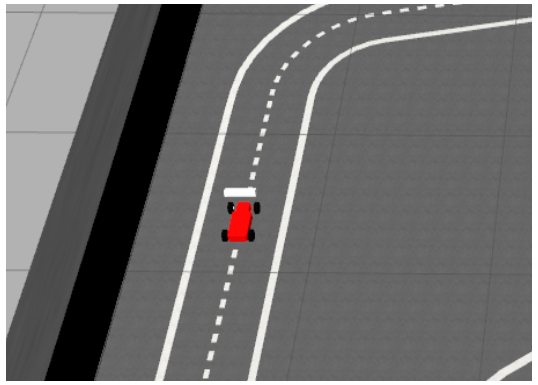


Kuka robot [Bai19]

Training using simulator

- Faster, cheap and safe.



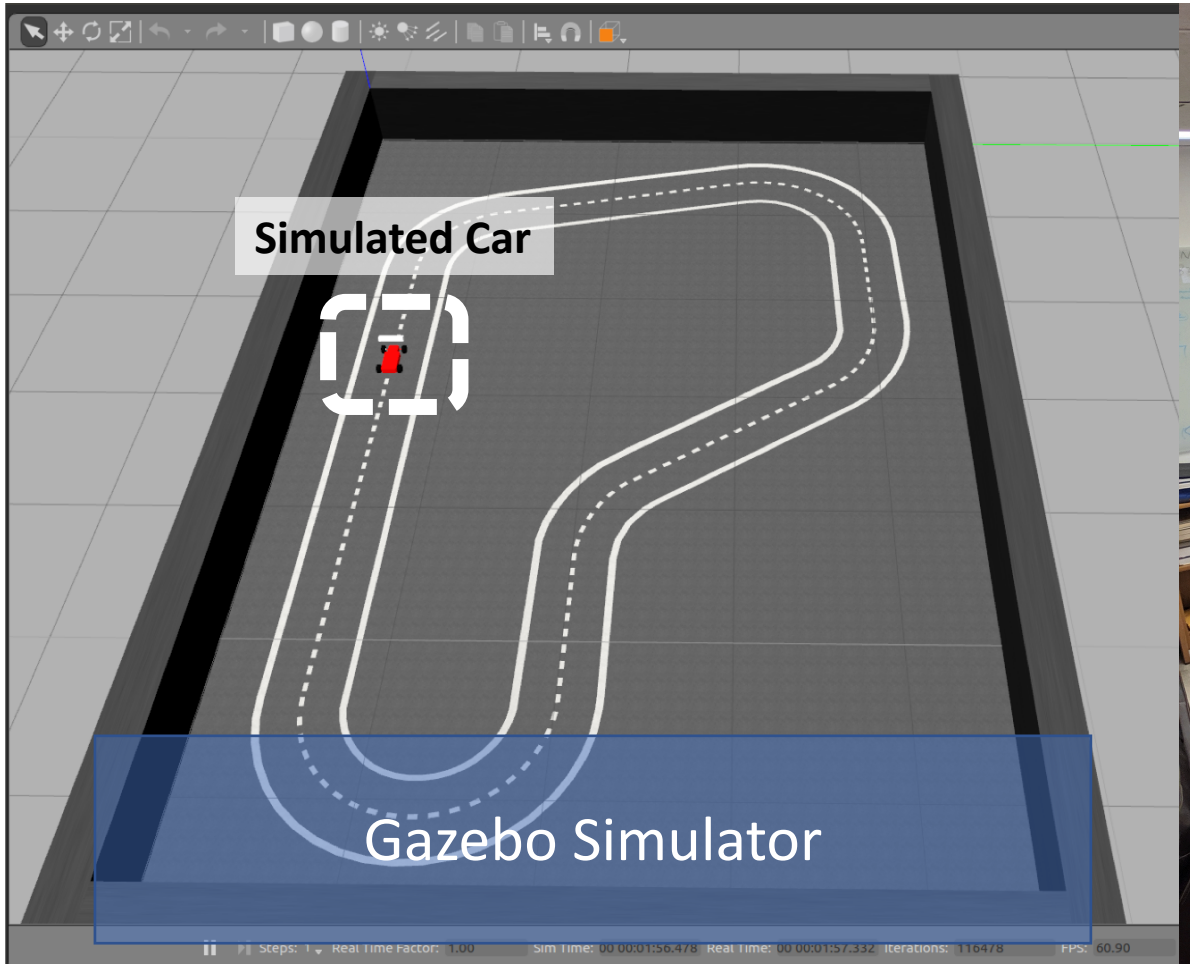


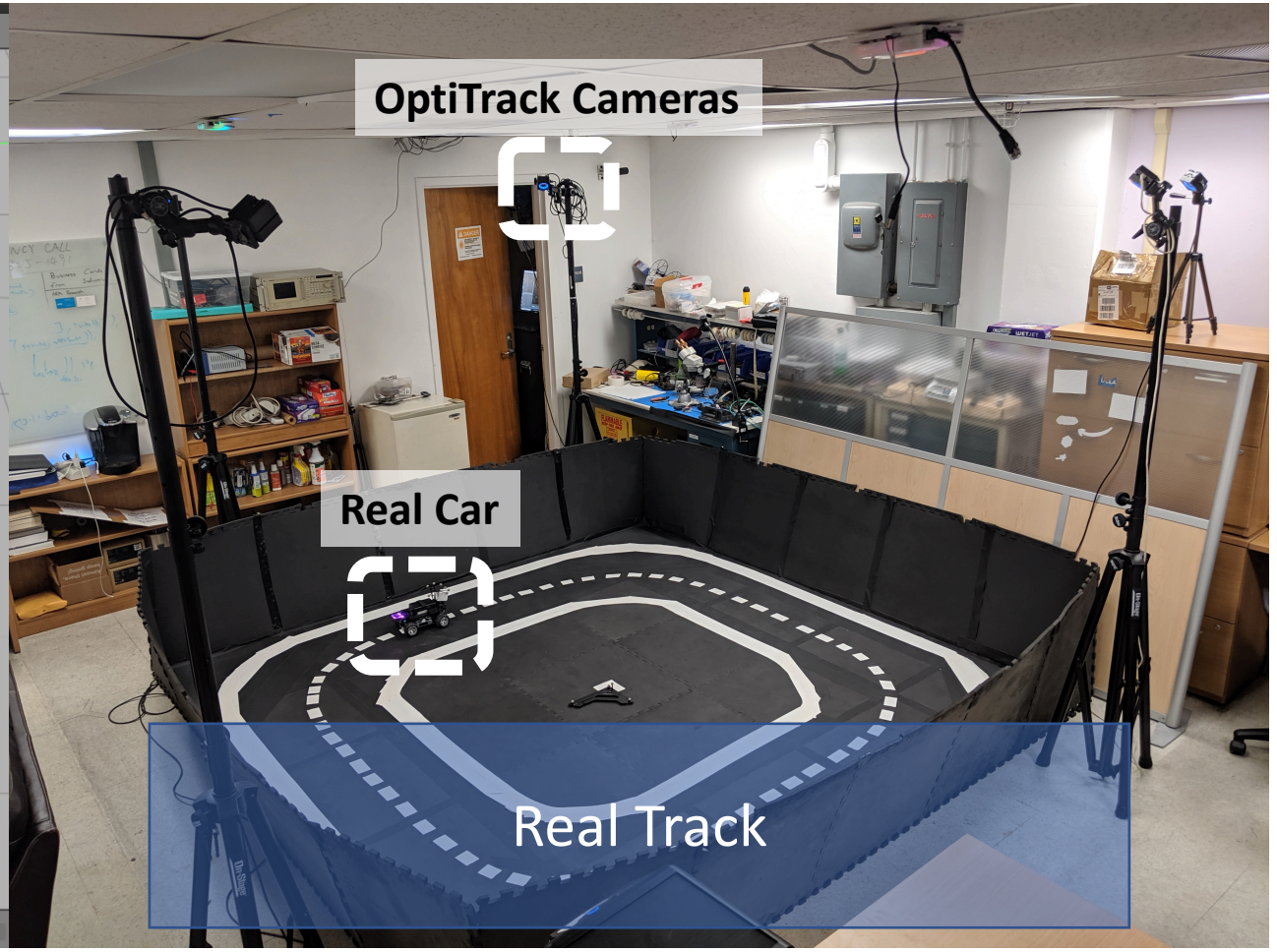
Autonomous car

Hand manipulation
[Andrychowicz20]

[Bai19] https://sim2real.github.io/assets/slides/bai-Learning_to_Grasp_Using_Simulation_Yunfei_Bai_Google_X.pdf
[Andrychowicz20] Andrychowicz, OpenAI: Marcin, et al. "Learning dexterous in-hand manipulation." The International Journal of Robotics Research 39.1 (2020): 3-20.

Sim2Real: Experimental Setup

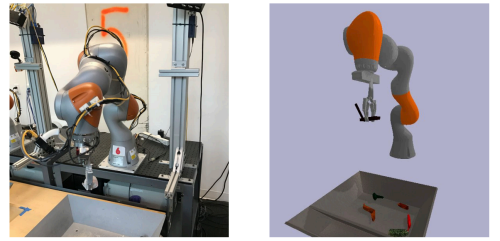




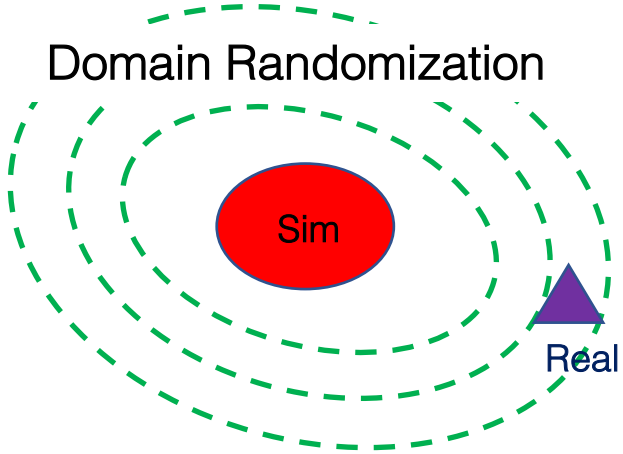
Policies for the Real Robots: Challenges

System identification

- Match simulator and the real world.
- Mass, friction, joint properties, actuator forces.



Kuka robot



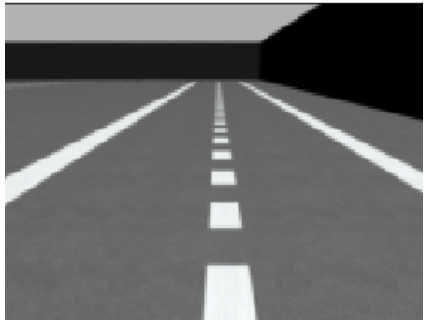
Hand manipulation

Domain randomization

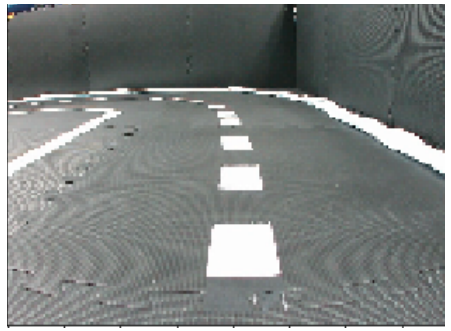
- Account for the differences in parameters.
- Sensing gap, temporal variations.
- We propose using Time-in-State RL for temporal variations.

Autonomous car

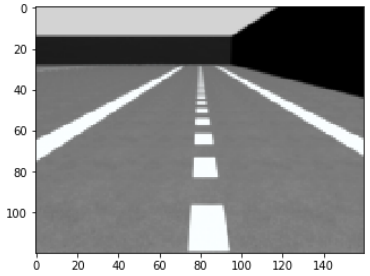
Autonomous Car: Sensing GAP



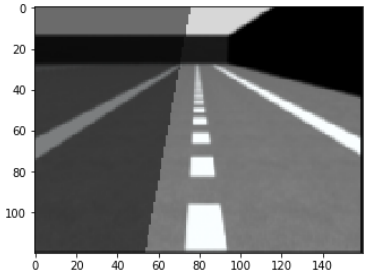
Simulator Track



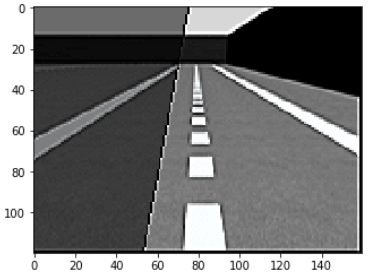
Real Track



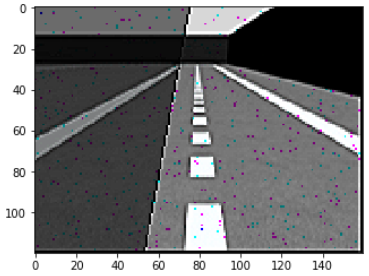
Brightness



Shadows



Sharpness



Salt and Paper

Domain randomization on camera images in simulator.

Evaluation on 1/18th Car

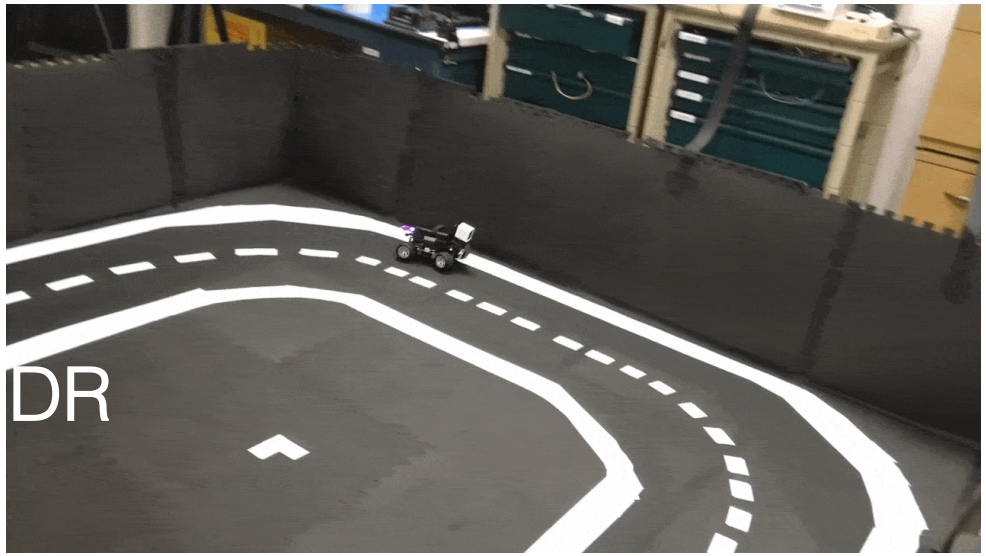
Latency	20ms	60ms	100ms
DR	20	11	7
TS	20	17	13

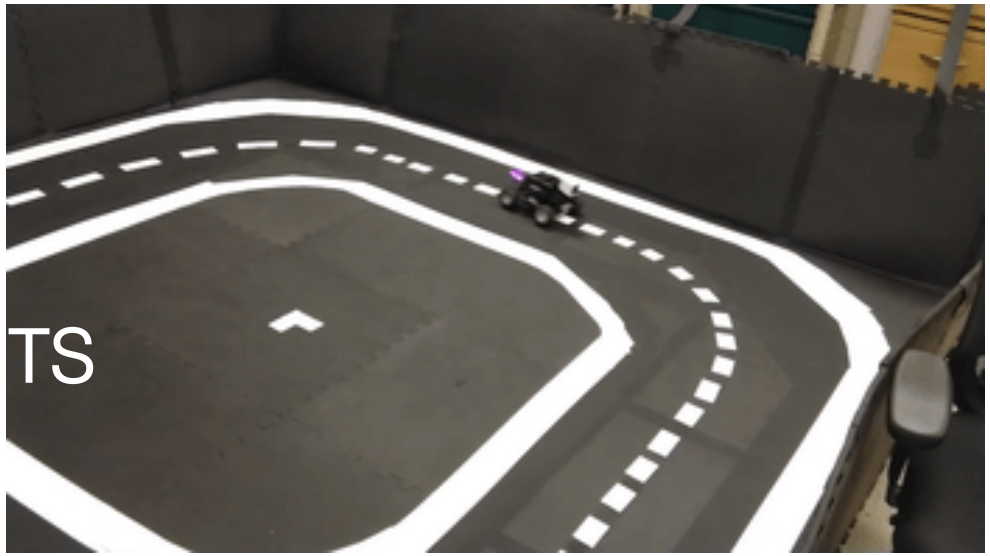
Laps Completed (out of 24)

Latency	20ms	60ms	100ms
DR	1.45m/s	1.44m/s	1.45m/s
TS	1.50m/s	1.45m/s	1.40m/s

Action Adaptation

Actions: Steering and speed
Car camera: 30 Hz, sampling interval = (27 ms – 37 ms)





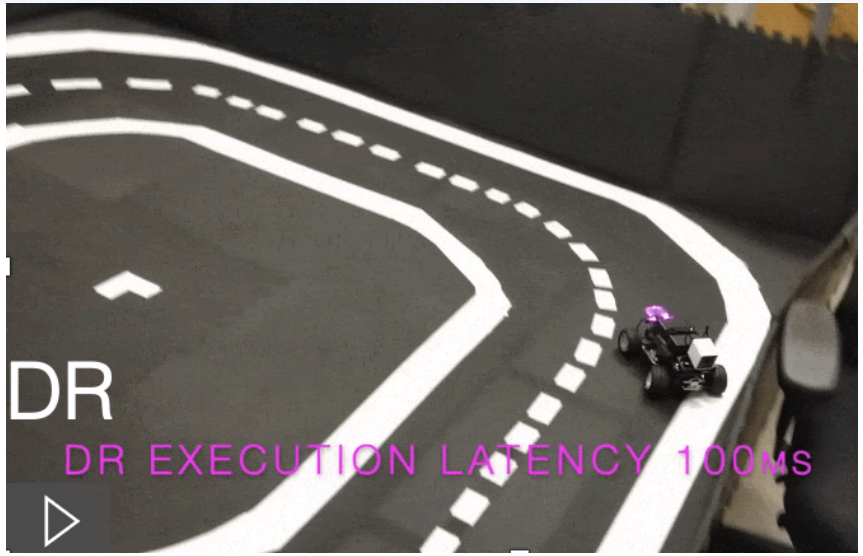
Evaluation on 1/18th Car

Latency	20ms	60ms	100ms
DR	20	11	7
TS	20	17	13

Laps Completed (out of 24)

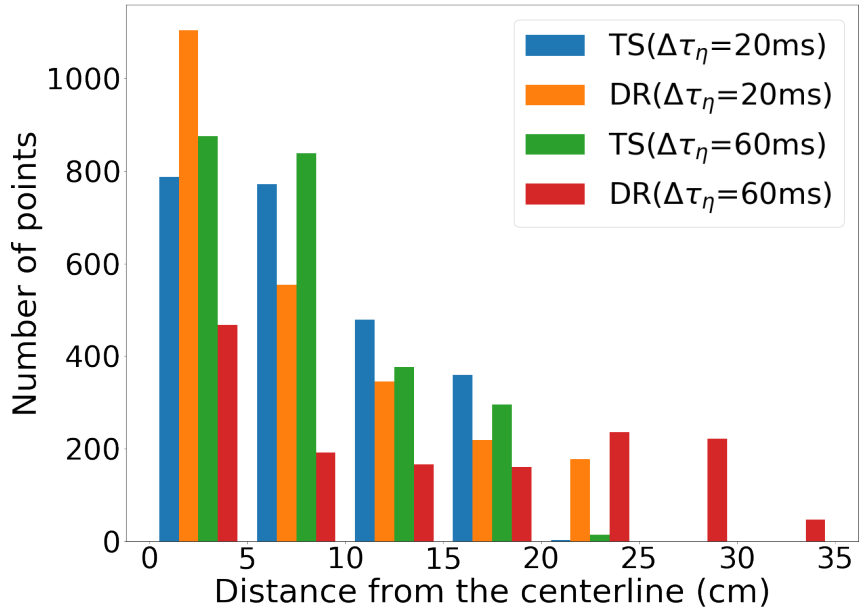
Latency	20ms	60ms	100ms
DR	1.45m/s	1.44m/s	1.45m/s
TS	1.50m/s	1.45m/s	1.40m/s

Action Adaptation

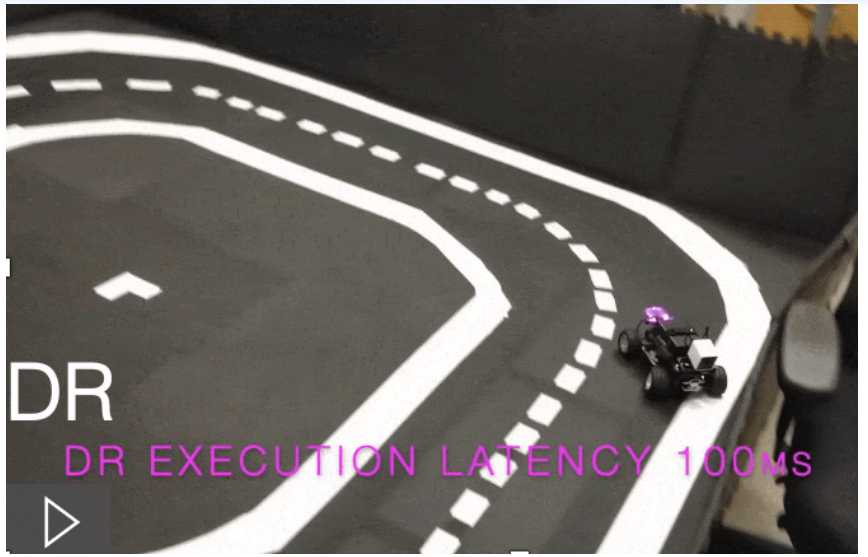


Actions: Steering and speed
Car camera: 30 Hz, sampling interval = (27 ms – 37 ms)

Evaluation on 1/18th Car



Distance measured



Actions: Steering and speed
Car camera: 30 Hz, sampling interval = (27 ms – 37 ms)

Discussion

- When does delay variation impact significantly?
  - Environment is time evolving such that delay has an impact on the outcome (State-Transitions) of an action.

- What if we add more timing noises/jitters to the Time-in-State?
  - TS policy becomes closer to the DR.
    TS policy also learns to be robust than being adaptive.

Future Work

- Delay estimation using previous delays
- Delay indicators as inputs to network
  - CPU's load, network state, Resource availability

Acknowledgements

Prof. Mani Srivastava

Dr. Luis Garcia

Dr. Bharathan Balaji

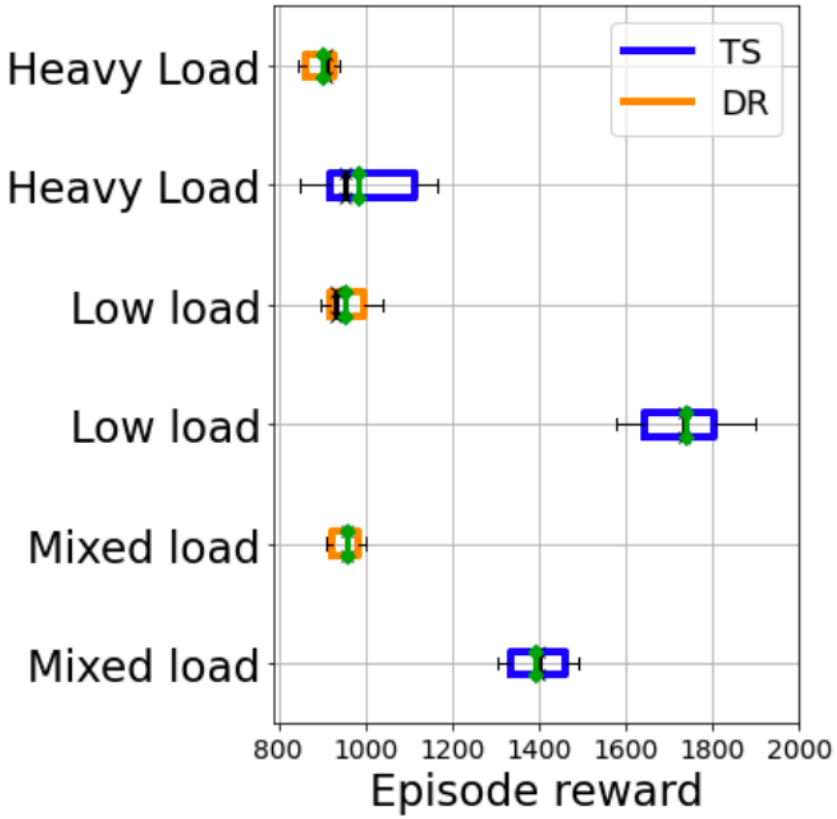
Prof. Fatima Anwar

Reference: Sandeep Singh Sandha, Luis Garcia, Bharathan Balaji, Fatima Anwar, Mani Srivastava, “Sim2Real Transfer for Deep Reinforcement Learning with Stochastic State Transition Delays,” Conference on Robot Learning (CoRL), 2020.
https://github.com/nesl/Time-in-State-RL/

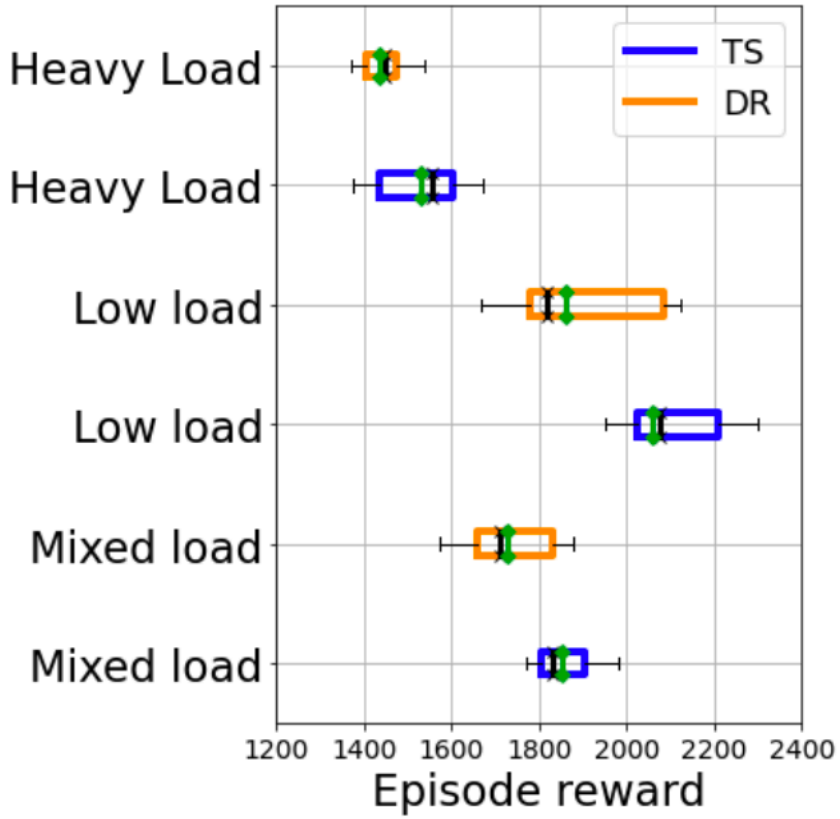
Funded in part by ARL, NSF, and SRC/DARPA.

Q & A

Backup: Evaluation for Simulated Multitenancy Setting



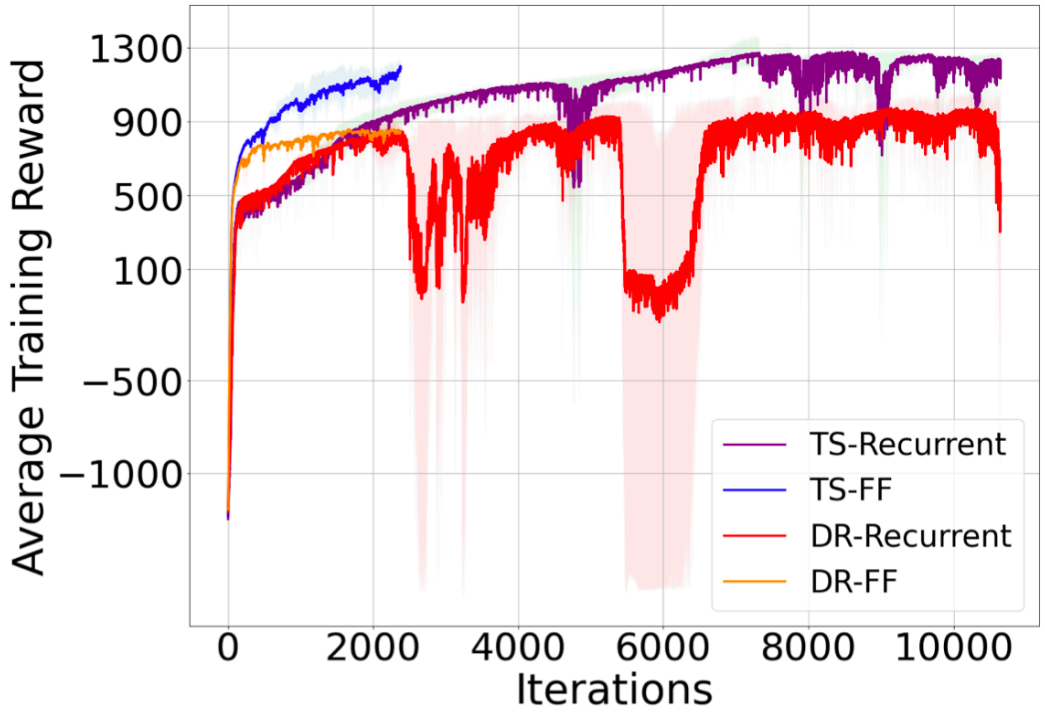
(a) HalfCheetah task

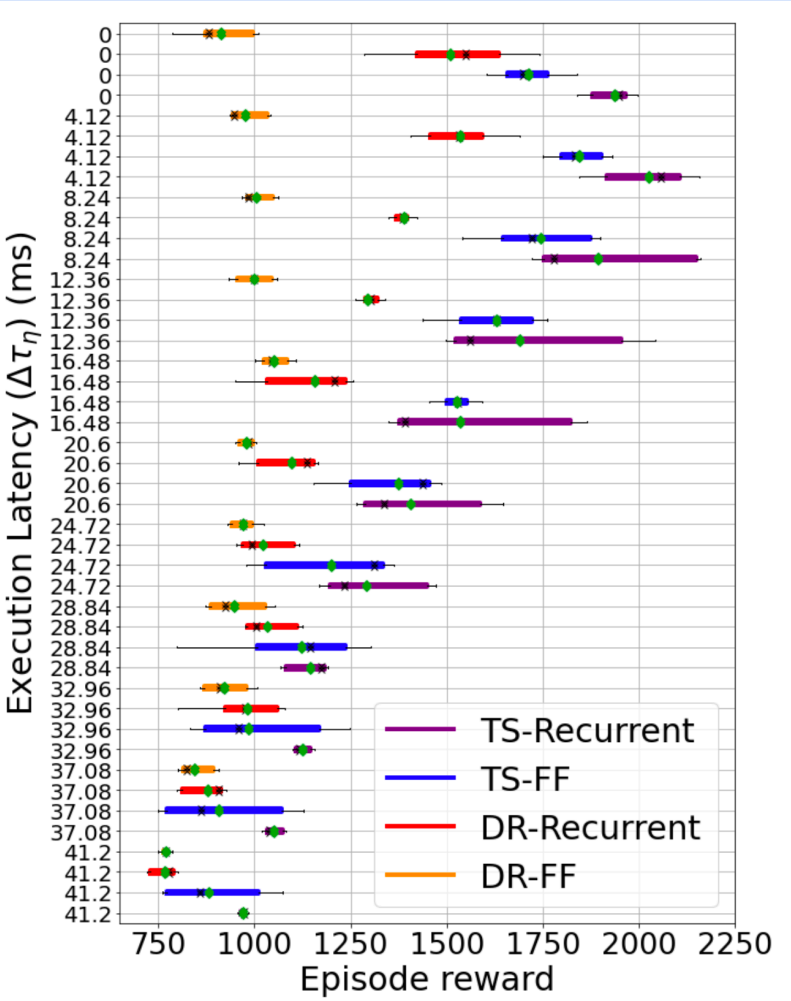


(b) Ant task

This behavior can be explained from degradation of policies with higher latencies.

Backup: Evaluation of Recurrent Policies: HalfCheetah





TS is still a better approach.

Backup: Mathematical Formulation

- Expected return $J(\pi) = \mathbb{E}_{r\sim p(\tau|\pi)} \left[\sum_{t=0}^{T-1} r(o_t, a_t) \right]$

Where $p(\tau|\pi)$ represents, the log likelihood of the trajectory

$$\tau = (o_0, a_0, o_1, \ldots, a_{T-1}, o_T)$$

- Finally, goal of agent is to learn optimal policy π^* = Max(J)

$p(\tau|\pi)$: Log likelihood of trajectory.

Backup: Mathematical Formulation

Log likelihood of trajectory $p(\tau|\pi)$ when following a particular policy.

$$p(\tau|\pi) = p(o_0)\prod_{t=0}^{T-1} p(o_{t+1}|o_t, a_t)\pi(o_t, a_t)$$

An assumption which is often made:
Observation (o) = State of the system (s).

$p(s_{t+1}|s_t, a_t)$: State transition model.

Backup: Mathematical Formulation

$$p(s_{t+1} \mid s_t, a_t)$$

State transition model gives us the consequence of action taken by agent and is **determined by dynamics and sensing of the environment[*]**.

The *sampling rate* and *inferencing latencies* are one of the important factors deciding the time for the agent to act and hence the consequence of action, thereby impacting **State transition model.**

Peng, Xue Bin, et al. "Sim-to-real transfer of robotic control with dynamics randomization." *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018.

Backup: Mathematical Formulation

$p(s_{t+1}|s_t, a_t)$

Source domain

$p^*(s_{t+1}|s_t, a_t)$

Target domain

Problem

Parameterizes the dynamics of simulation

$p(s_{t+1}|s_t, a_t, \mu)$

Source domain: Sim

$p^*(s_{t+1}|s_t, a_t)$

Target domain: Sim/Real

Domain Randomization

$p(s_{t+1}|[s_t, sr, il], a_t, \mu)$

Source domain: Sim

sr and il Varied during runtime

$p^*(s_{t+1}|s_t, sr^*, il^*, a_t)$

Target domain: Sim/Real

sr and il of target domain are used

Time-in-State