Winter Term 21/22

Adversarial Self-Supervised Learning with Digital Twins

Lecture-5: Sim2Real

Prof. Dr. Holger Giese (holger.giese@hpi.uni-potsdam.de)

Christian Medeiros Adriano (christian.adriano@hpi.de) - **"Chris"**

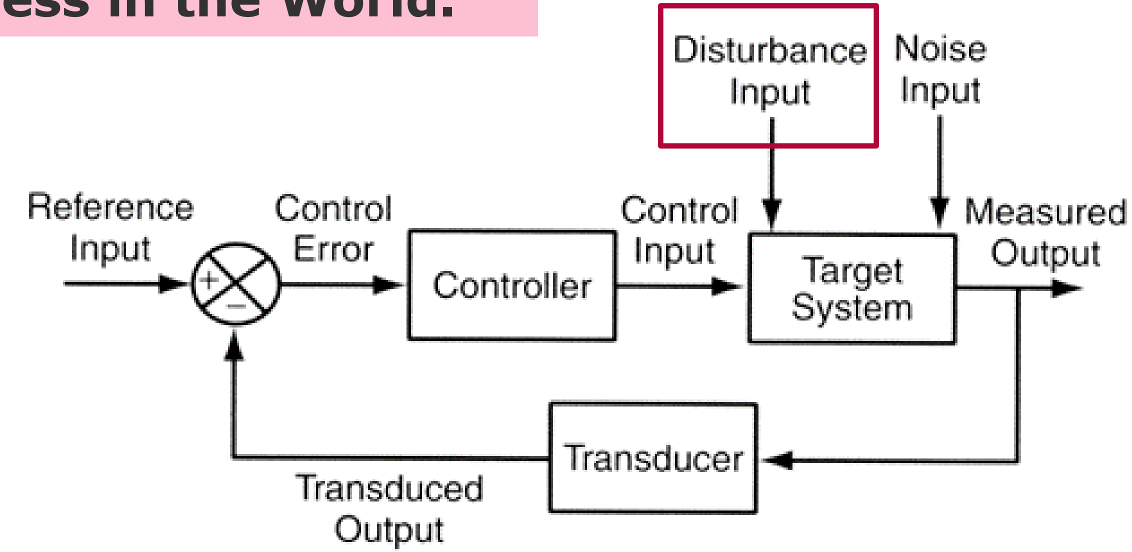
He Xu (he.xu@hpi.de)

Infrastructure to run experiments

"Success in the Lab is not guarantee of success in the World."

Feedback loop models

"When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one." **Vladimir Vapnik**

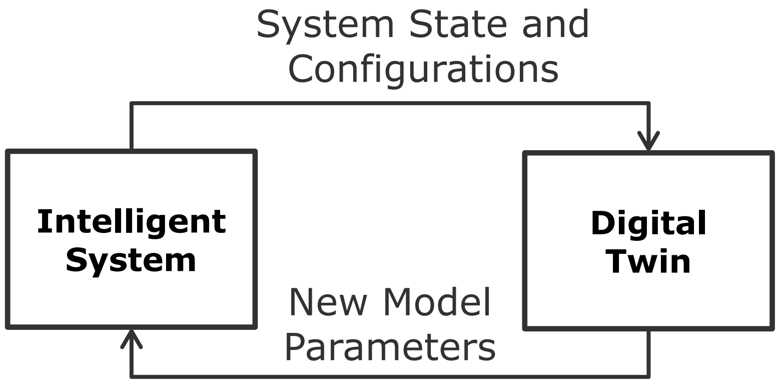


Simulation models

"Thinking is acting in an imagined space" **Konrad Lorenz**

"Perception is a generative act" – [Gross et al. 1999]

" Consciousness is a controlled hallucination " - [Seth et al. 2000]



Operating in Sparse, Safety Critical, Uncertainty World

- Operation produces few if any training examples (relevant events are rare)
- Predictive performance is necessary but not sufficient
- Simulation requirements for robustness are functions of the operational context, hence:
  - Requirements are domain-specific, which make them more challenging to automate
  - Their reification happens outside the machine learning algorithms

Requirements Engineering Questions

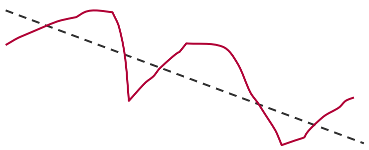
1. Which simulation requirements do we need?
2. How do we verify that these simulation requirements are being met?
3. How to incorporate these simulation requirements into training?
4. How do we transfer the simulation outcomes to production?

Advantages of Simulated Data

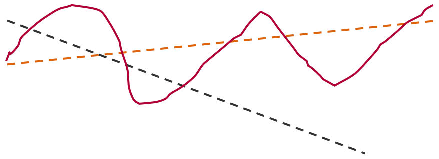
- **Cheaper**: do not require long and error-prone system executions
- **Faster**: can focus on a certain system state
- **More Scalable**: can be parallelizable, be enriched via data fusion, etc.
- **Safer**: does not risk catastrophic incidents in production
- **Higher Quality**: pre-processed to be labeled, reduced measurement error, missing data, etc.
- **Less Biased / More Diverse**: not limited to the real-world probability distributions

However, simulation is also difficult

- Difficult to accurately & efficiently model sensors & the real-world
- Small modeling errors can compound in large control errors (why? because they are correlated)



Random error



Compounded error

- If there is some pattern (even accidental) in the data, the ML model will explore it.
- Simulator make trade-offs between fidelity and speed, usually favoring the latter
- Even if we model everything accurately, we still need to get the parameters right.
- How to measure phenomena that is not directly observable in the data? In physical systems they correspond, for instance, to damping, inertia, friction, etc. Or in our systems, they can correspond to network instability, surge in user requests, hardware failures, tec.
- The more accurate the model more parameters, more data, slower simulation.

DeepRacer [Balaji 2019]

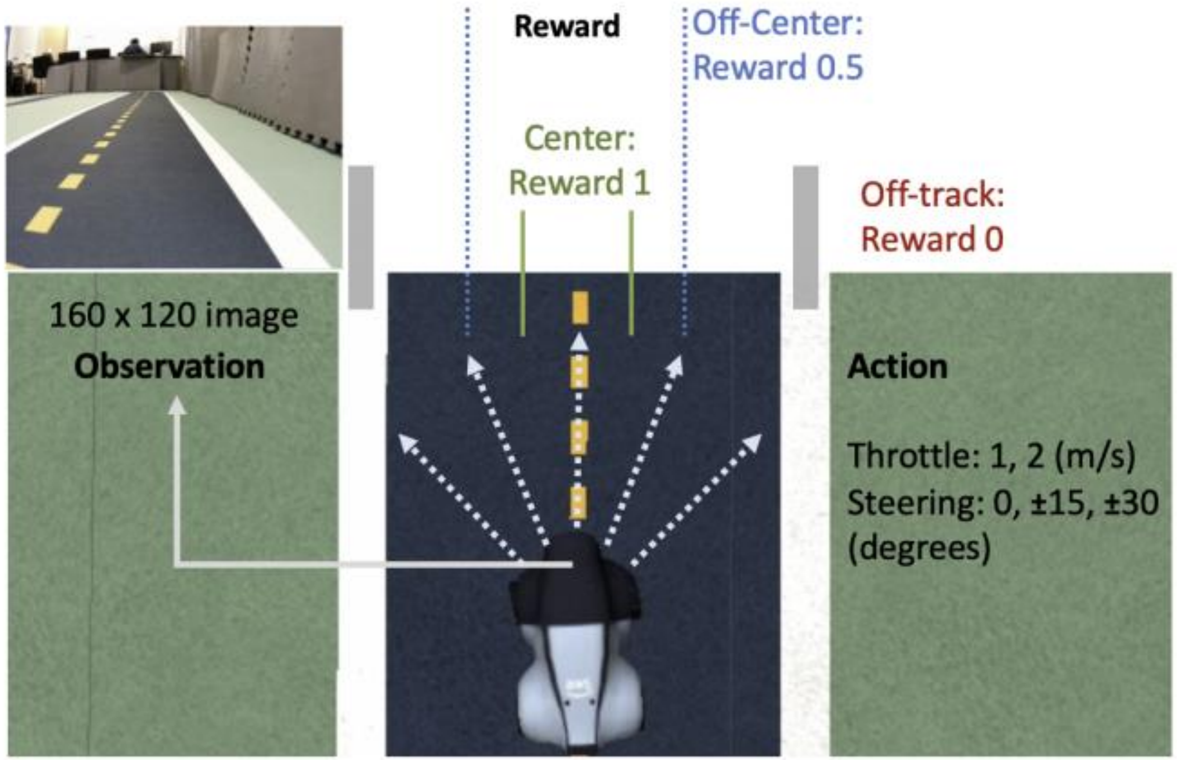


Fig. 1: Observation, action and reward for DeepRacer agent

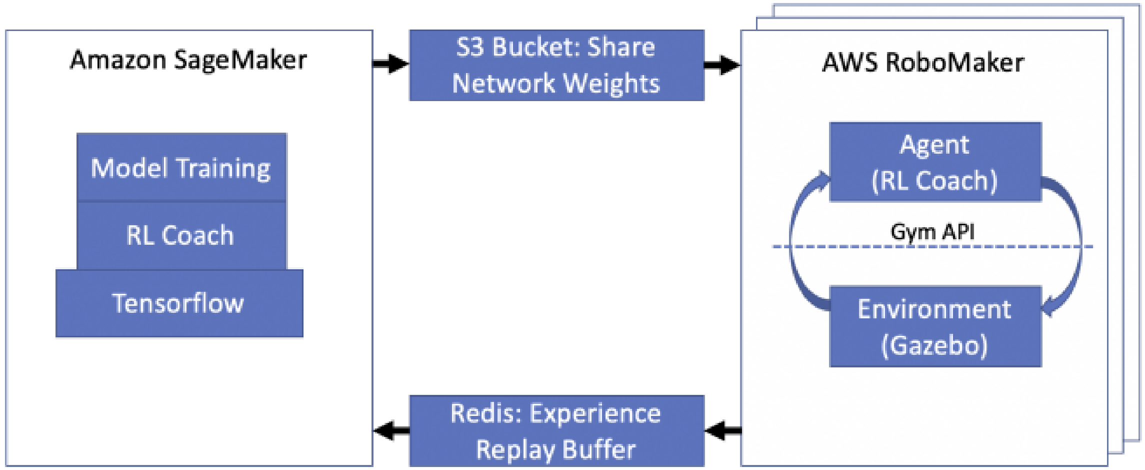


Fig. 2: Training the agent with DeepRacer distributed rollouts

Sim2Real Calibration

- Height, angle and the field of view of the simulation camera to match the real images
- Use the same frame rate 15fps
- Producer-consumer mechanism to ensure one action per image

Balaji, B., et al. (2019**). Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning**.

Sim2Real Transfer for Reinforcement Learning [Kaspar 2020]

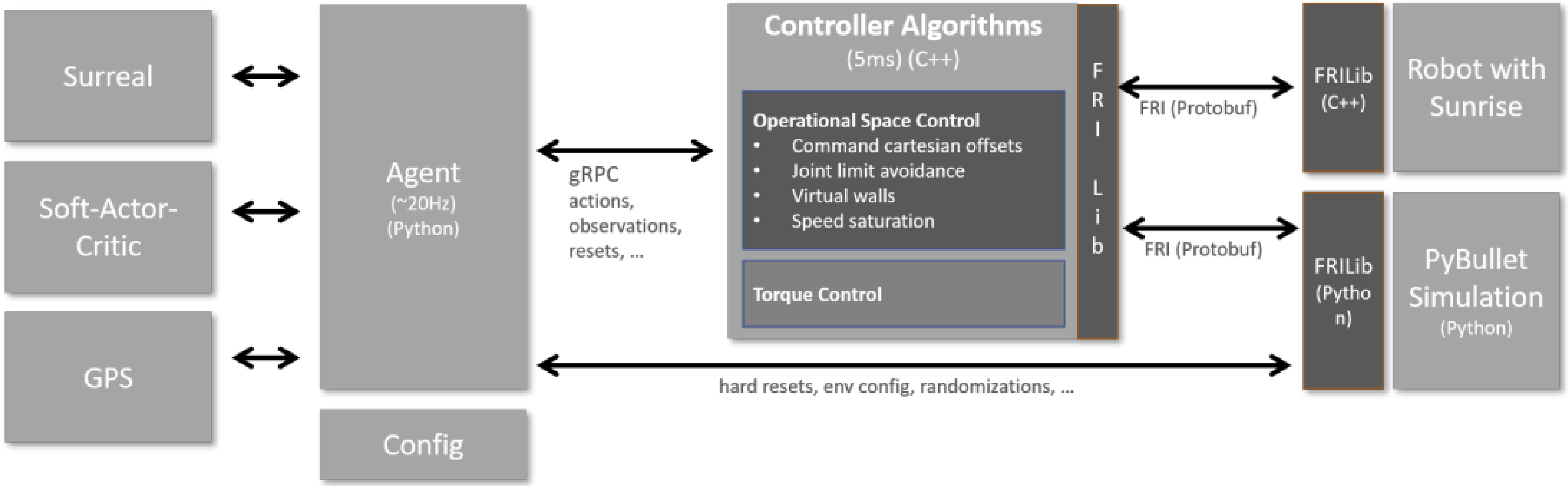


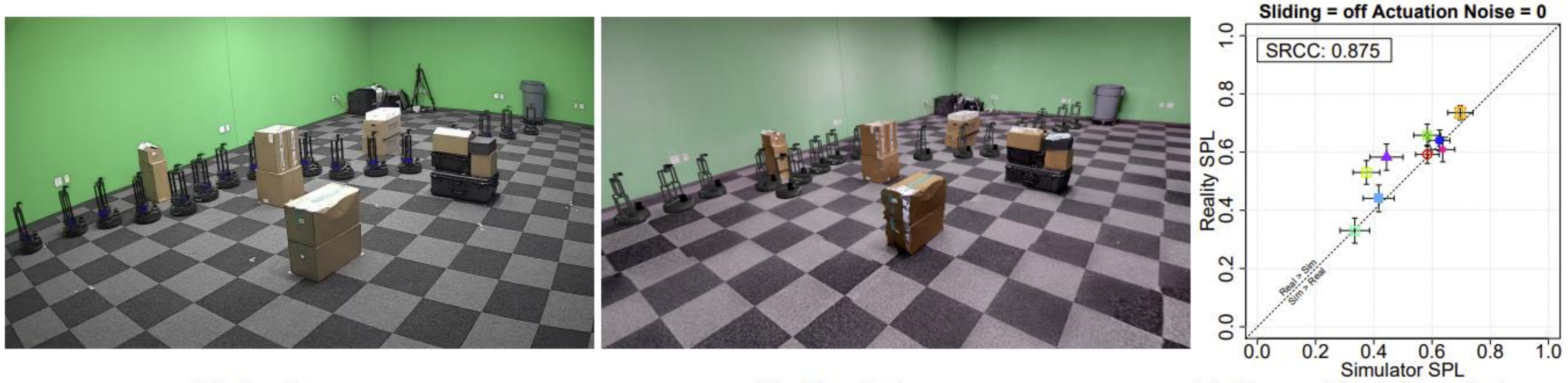
Fig. 2: Architecture for learning and controlling robot and simulation

Sim to Real Transfer

- Simulation environment (PyBullet)
- Dynamics and Environment Randomization
- System Identification
- Mitigate that initial simulation transferred to the real robot works too poorly.

Kaspar, M., Osorio, J. D. M., & Bock, J. (2020). **Sim2real transfer for reinforcement learning without dynamics randomization.** In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4383-4388). IEEE.

Sim2Real Predictivity [Kadian 2020]



(a) Reality. (b) Simulation. (c) Sim-vs-Real Correlation.

Fig. 1: We measure the correlation between visual navigation performance in simulation and in reality by virtualizing reality and executing parallel experiments. (a): Navigation trajectory in a real space with obstacles. (b): virtualized replica in simulation. (c): we propose the Sim2Real Correlation Coefficient (SRCC) as a measure of simulation predictivity. By optimizing for SRCC, we arrive at simulation settings that are highly predictive of real-world performance.

Problem: Sim-vs-Real Correlation Coefficient (SRCC) to quantify predictivity = **0.18**

Reason: The gap is largely due to AI agents learning to exploit simulator imperfections – abusing collision dynamics to 'slide' along walls , leading to shortcuts, through otherwise non-navigable space.

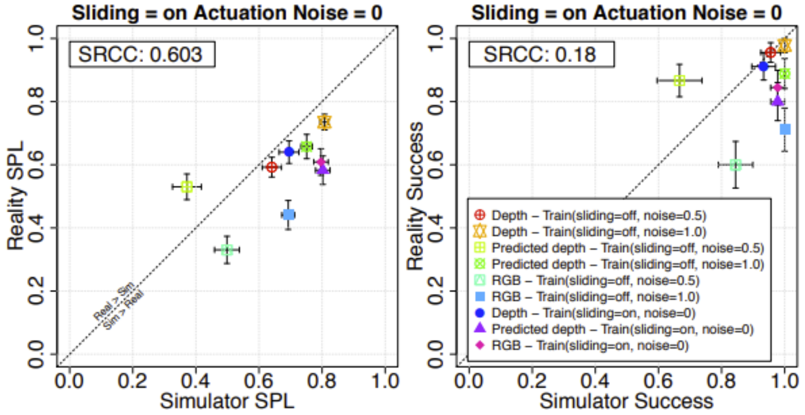


Fig. 5: SRCC_{SPL} (left) and $\text{SRCC}_{\text{Succ}}$ (right) plots for AI Habitat Challenge 2019 test-sim setting in the CODA environment. We note a relatively low correlation between real and simulated performance.

Solution: Their experiments show that it is possible to tune simulation parameters to improve sim2real predictivity (improving SRCC_Succ from 0.18 to 0.844) – increasing confidence that in-simulation comparisons will translate to deployed systems in reality.

Kadian, A., Truong, J., Gokaslan, A., Clegg, A., Wijmans, E., Lee, S., ... & Batra, D. (2020). **Sim2Real predictivity: Does evaluation in simulation predict real-world performance?**. *IEEE Robotics and Automation Letters*, *5*(4), 6670-6677.

Further readings recommended

Traoré, R., Caselles-Dupré, H., Lesort, T., Sun, T., Díaz-Rodríguez, N., & Filliat, D. (2019). **Continual reinforcement learning deployed in real-life using policy distillation and sim2real transfer**

How to prevent catastrophic forgetting in sim2real

Xiao, C., Lu, P., & He, Q. (2021**). Flying Through a Narrow Gap Using End-to-End Deep Reinforcement Learning Augmented With Curriculum Learning and Sim2Real**.

IEEE Transactions on Neural Networks and Learning Systems.

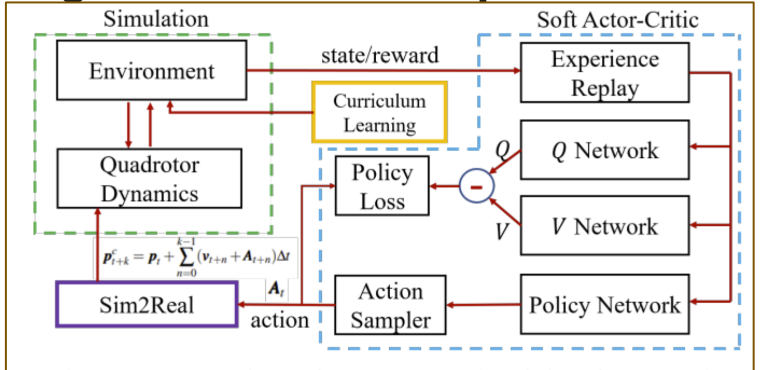


Fig. 3: An overview of our proposed training framework.

Read in more detail and check citations

Kaspar, M., Osorio, J. D. M., & Bock, J. (2020). **Sim2real transfer for reinforcement learning without dynamics randomization.** In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 4383-4388). IEEE.

Kadian, A., Truong, J., Gokaslan, A., Clegg, A., Wijmans, E., Lee, S., ... & Batra, D. (2020). **Sim2Real predictivity: Does evaluation in simulation predict real-world performance?**. *IEEE Robotics and Automation Letters*, *5*(4), 6670-6677.

Balaji, B., Mallya, S., Genc, S., Gupta, S., Dirac, L., Khare, V., ... & Karuppasamy, D. (2019**). Deepracer: Educational autonomous racing platform for experimentation with sim2real reinforcement learning**. *arXiv preprint arXiv:1911.01562*.

Why do we need a simulation of the world?

- Observations not reliable, are high dimensional and multi-modal
- Reward from real-world is hard.
- The real world usually does not behave like an MDP (Markov Decision Process)
- ?

Sources of Difficulties

1. Latent variables
2. Brittle mechanisms
    1. might not be independent and
    2. Interventions can alter mechanisms
3. Sparsity of relevant events (failures, rewards)

Mitigating Solutions

1. model the world as a Partially Observable MDP (POMDP)
2. monitor the mechanisms for unexpected changes
3. simulate the transitions in a hypothetical world to accelerate learning

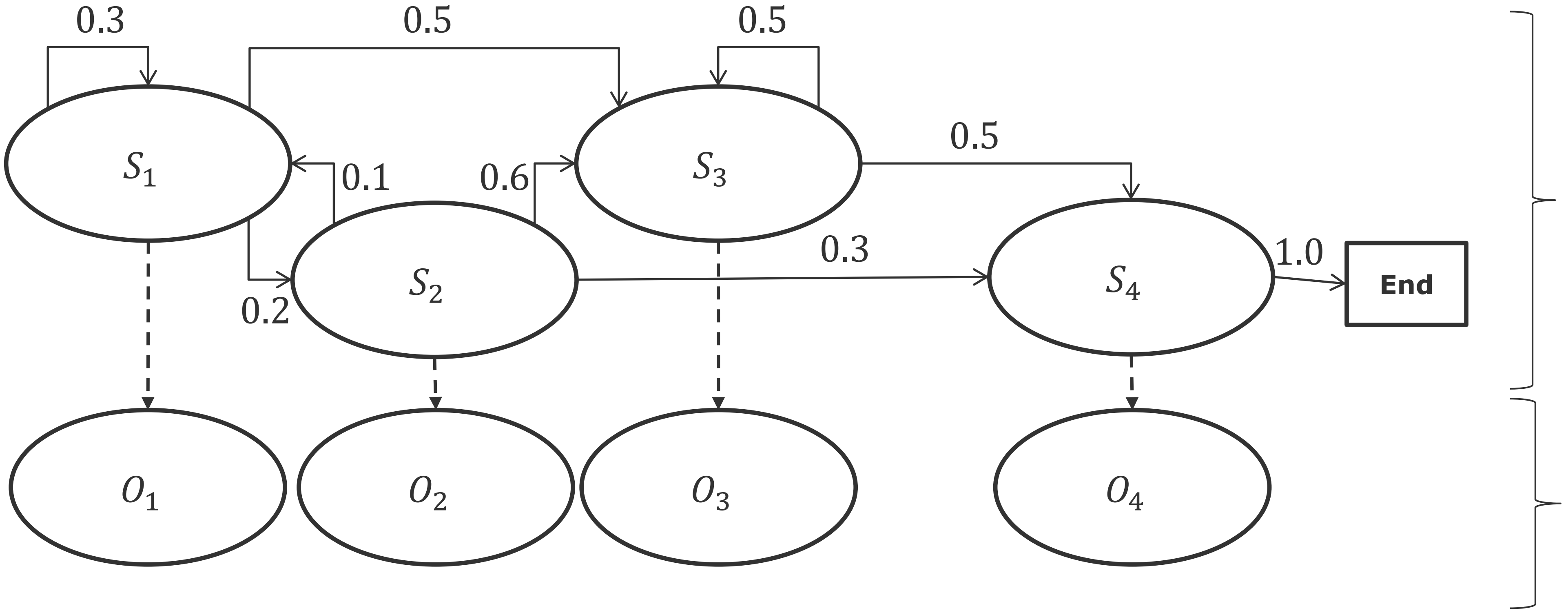
Definitions (informal)

Mechanism: is a sequence of causal events that propagates (usually through multiple paths) the effect of an intervention until one or more outcome variables

Intervention: consists of fixing to value or varying within a range of values one or more variables. An intervention on a node disconnects the node from its parents, i.e., make any intervention on the parents ineffective to the node.

Condition: consists of filtering events (rows on a probability matrix) by the value of variable (node). Conditioning does not cut the incoming arrows to the node, i.e., the node remains connected to its parents.

Hidden Markov Model - HMM



Assumptions

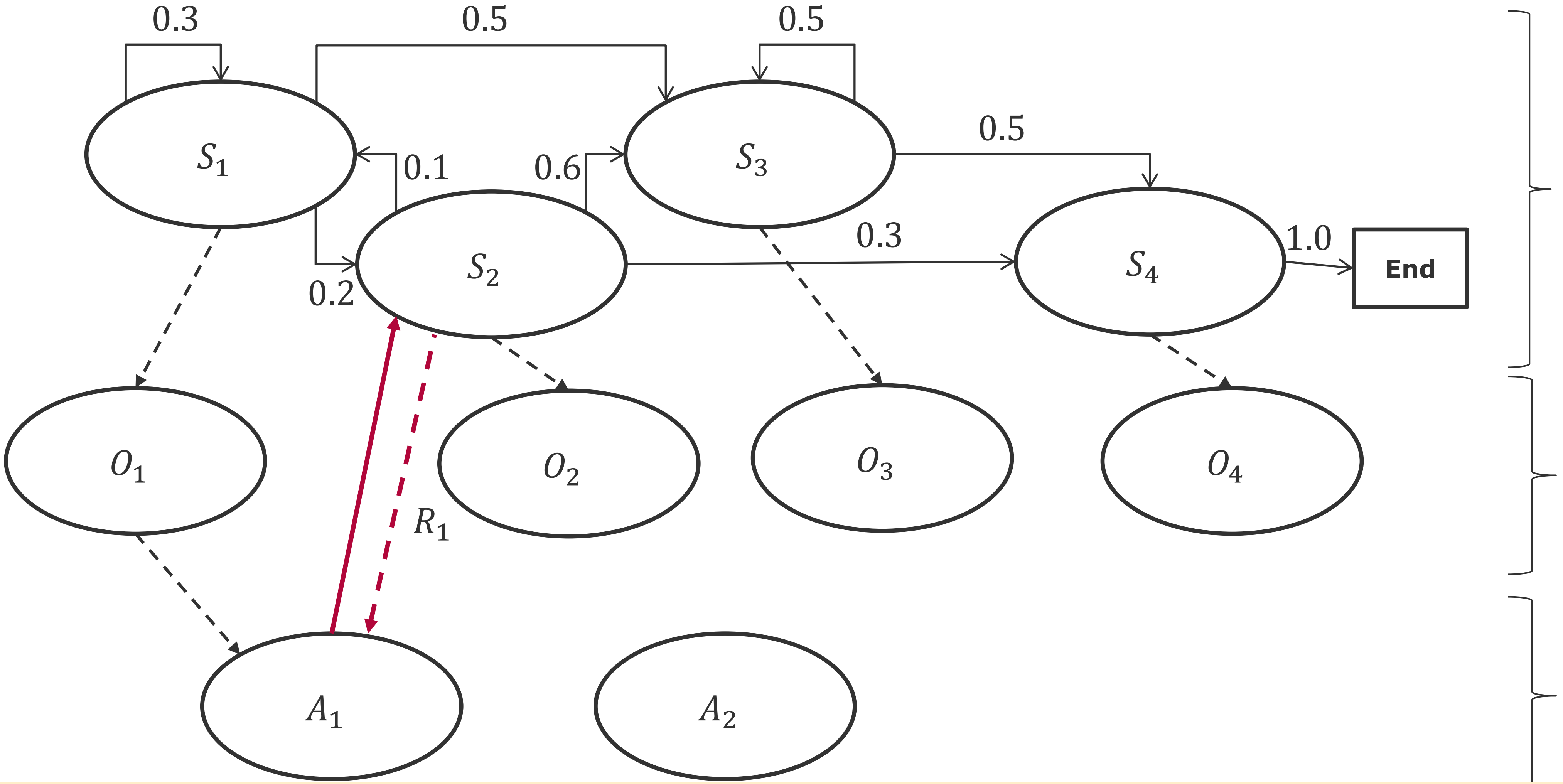
First order Markovian: $P(S_{t+1}|\, S_t) \;=\; P(S_{t+1}|\, S_t, S_{t-1}\,, \ldots, S_{t-n}\,)$

Stationary:

$P(S_{t+1}|\, S_t) \;=\; P(S_t|S_{t-1})\, \forall t$ hidden states

$P(O_{t+1}|\, S_t) \;= P(O_t|S_{t-1})\, \forall t$ observations

Partial Observable Markov Decision Process - POMDP



While in the HMM we are passively observing and trying to infer the real state, in the POMDP we have partial control over the state transitions through the actions, whose effects produces a signal (reward Ri) for future actions.

Preliminary Work-Packages

1- Hidden Markov Model

Study how to move the HMM implemented in the Python side to the Java side

2- Failure Inject Mechanism

Study how to generate failure injects that reflect that failure propagation patterns

3- Reinforcement Learning

Study how to replace the Supervised Learning controller (Regression) with a Self-Supervised One (RL)

4- Monitoring

Study how to visualize the utility, risk, and mechanism gap between the Real and Simulated (Digital-Twin)

5- Adversarial Tests

Study how to generate stress tests that show how policies that have equivalent predictive outcome at training present distinct outcomes at adversarial test sets

Project Goals - Research Problems

1- Under-specification Problem **Simulation**

Goal: Show that different prediction models solve the task well for testing data, however, perform very differently in two distinct situations:

1.1 distinct hyper-parameters (prior knowledge)

1.2 out-of-distribution data (distribution shifts)

2- Value-at-Risk Problem **Sim2Real**

Goal: Show different rates of synchronization between Production and Simulation can lead to:

2.1 excessive cost of training and redeployment

2.2 increase in the risk of under-performance

3- Learning to Synchronize Problem **Feedback Loop**

Goal: Show that different strategies to learn when to train and redeploy require:

3.1 more data to achieve an average value-at-risk

3.2 longer time to converge

Next tasks

Think about answers to the for the following questions:

1. How to generate test-data?
2. How to generate distribution shifts? How much shift (how to measure it)?
3. How to train distinct models that perform equally well on test data?
4. How do you know that two models are distinct?
5. How do you know that they perform equally well on the test data?
6. Preliminary assignment and ideas for the work-packages

Suggested Deadline = Nov 16 (present during lecture)

Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

END