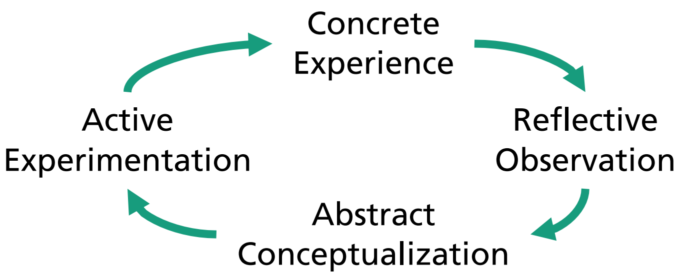
Learning from Experience

Auftaktveranstaltung ADA Lovelace Center am 4. Dezember 2019

Dr.-Ing. Christopher Mutschler

What do we mean with “Learning from Experience”?
An unbiased first view

- Process of learning through experience (aka Hands-On-Learning)
- “Experiential Learning”-Model by Kolb et al.:

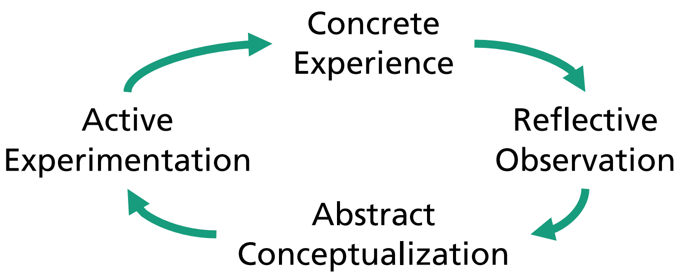


http://www2.le.ac.uk/departments/gradschool/training/resources/teaching/theories/kolb

What do we mean with “Learning from Experience”?

~~An unbiased first view~~

- Process of learning through experience (aka Hands-On-Learning)
- “Experiential Learning”-Model by Kolb et al.:



- No Teacher, simply meaning-making direct experience, but the learner must
  - be willing to be actively involved in the experience;
  - be able to reflect on the experience;
  - possess and use analytical skills to conceptualize the experience; and
  - possess decision making and problem solving skills in order to use the new ideas gained from the experience.

C’mon, why don’t we use a neural network for this?!

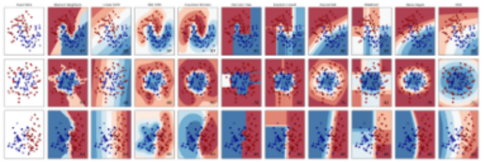
http://www2.le.ac.uk/departments/gradschool/training/resources/teaching/theories/kolb

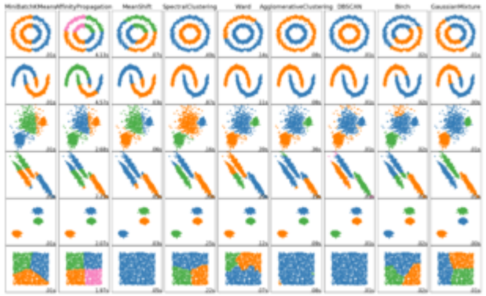
Learning from Experience
Continual Learning

- *Continual Learning* is the ability of a model to
  - Learn continually from a stream of data,
  - build on what was previously learnt (i.e., positive transfer), and
  - remember previously seen tasks.
- Applications? Relevance?
- Technical requirements:
  - Efficiency
  - Adaptiveness
  - Scalability

Learning from Experience
The technical view

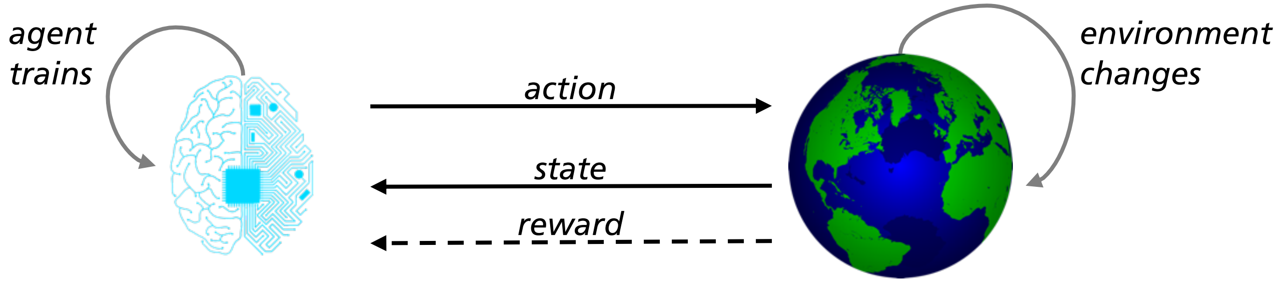
- Unsupervised learning
  - Modeling probability distributions
  - Clustering
  - Detecting Anomalies
- Supervised learning
  - Derive a function that maps input data to output data
  - Goal: Minimize empirical risk
  - Many aspects to consider: Bias-variance tradeoff, function complexity and regularization, inductive Bias, non-linearities, non *i.i.d.*, fairness, …

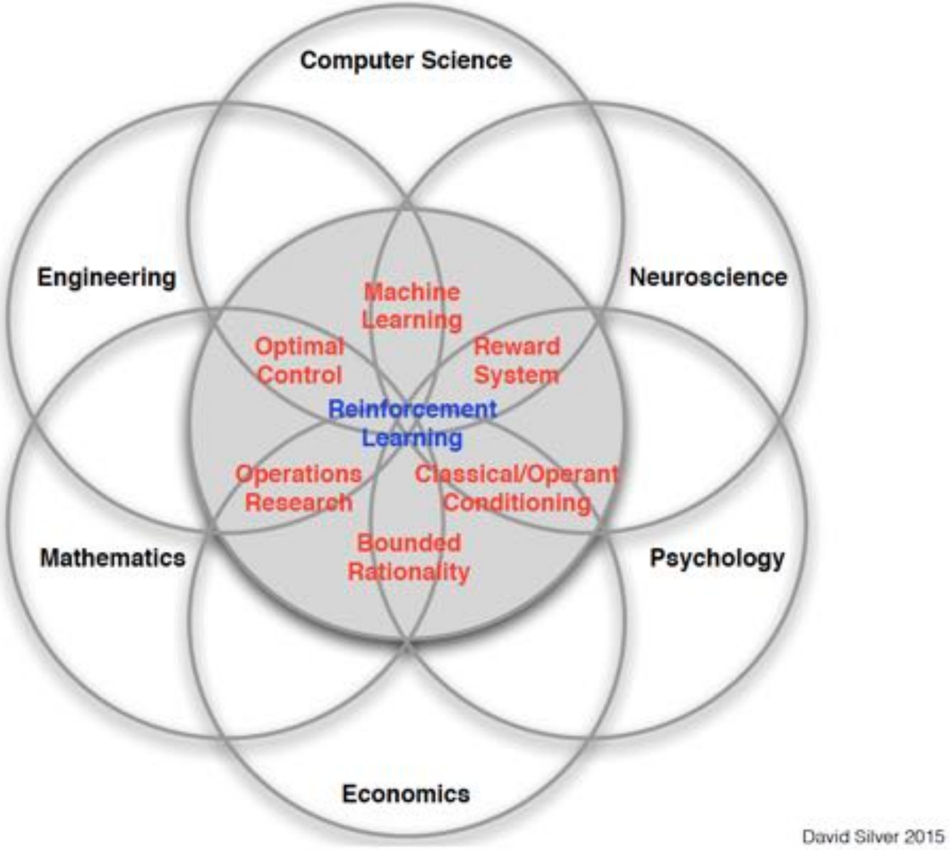




Learning from Experience
The technical view: Reinforcement Learning

- Derive a sequence of actions that maximize a notion of cumulative reward (a *policy*)
- Solve the “Markov Decision Process (MDP)” in interaction with the environment

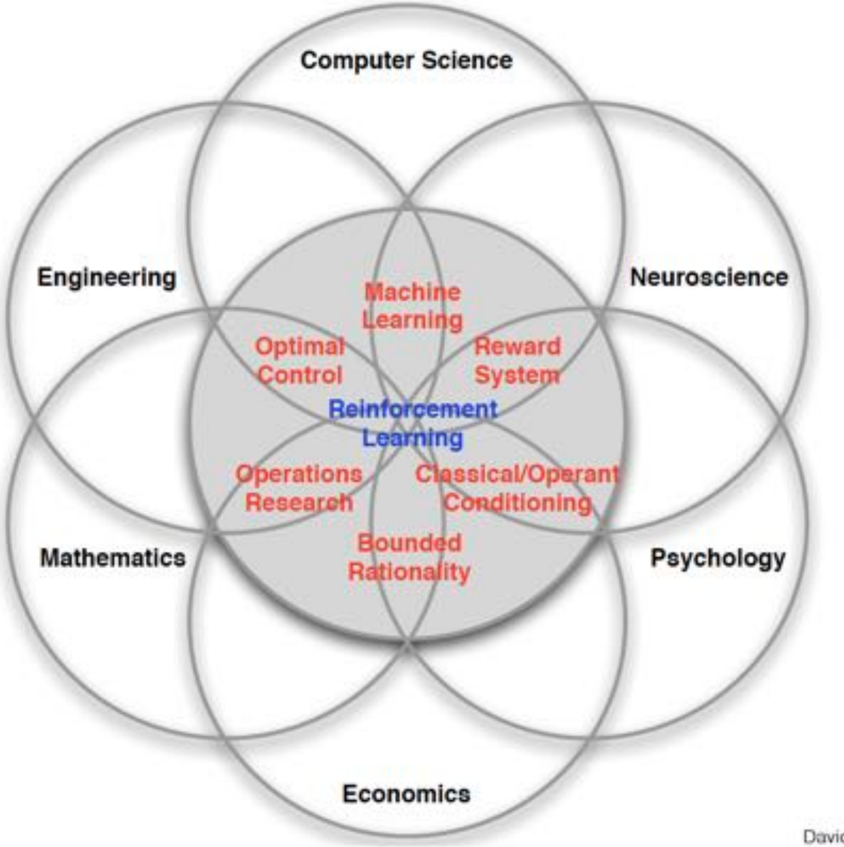




Learning from Experience
The technical view: Reinforcement Learning

- Run in *experiments*
- Data is provided step by step; data *evolves*
- From many experiments the agent learns a function (e.g. a neural network) that assigns probabilities (or values) to states or state-action-pairs

- Challenges:
  - Labeled samples not available
  - Reward may have a delay
    → sub-optimal actions do not get corrected
  - Balance exploration vs. exploitation
  - State spaces may become large



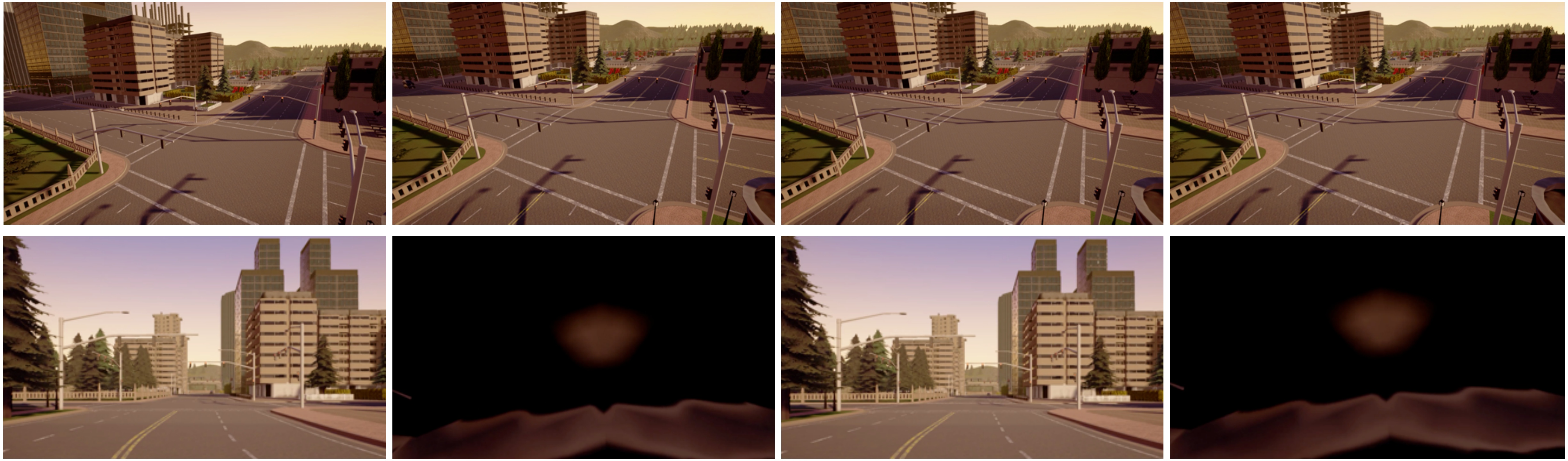
David Silver 2015

Learning from Experience
The technical view: Reinforcement Learning & Deep Learning

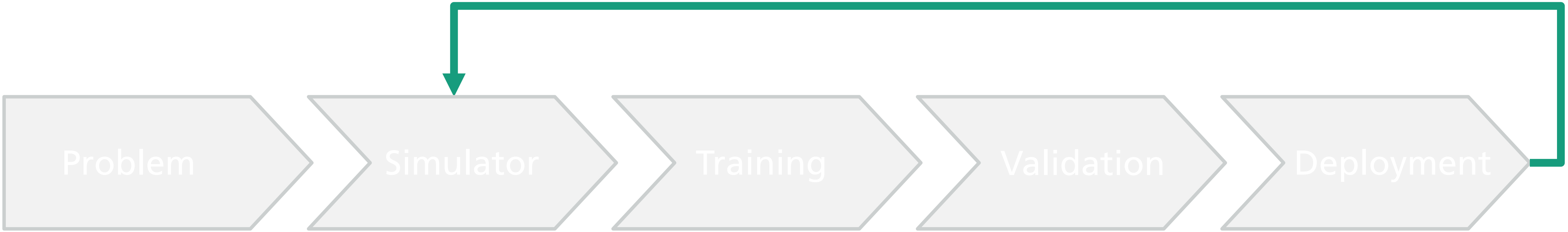
- Problem: in reality state spaces are large and can hardly be enumerated
- Many environments do not provide an abstract form of a state

  → Combine RL with Deep Learning

Case Study: Car Crash Scenario
Intelligent ADAS using (Deep) RL



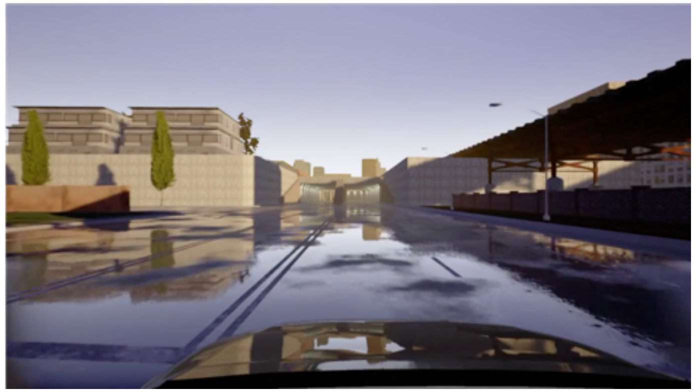
Case Study: Car Crash Scenario
How to get this into a running prototype



Case Study: Car Crash Scenario
Formalize the Problem

Case Study: Car Crash Scenario
Formalize the Problem

- State space:
  - everything in the simulator (too much and most of it irrelevant)
- Observation space (final 19 – after many trial-and-error attempts):
  - ego vehicle location in x, y; ego vehicle pitch, roll, yaw angles; ego vehicle speed in x, y; Distance + angle from 1 waypoint ahead; ego vehicle acceleration in x, y; throttle, steering, braking commands; timestep; Obstacle bounding box location in x, y; Obstacle bounding box extend in x, y
  - *Alternative: camera input*
- Action spaces:
  - Discrete: throttle 0 or 1, brake 0 or 1, steering discretizing in 7 points
  - Continuous: throttle in [0, 1], brake in [0,1], steering in [-1, +1]

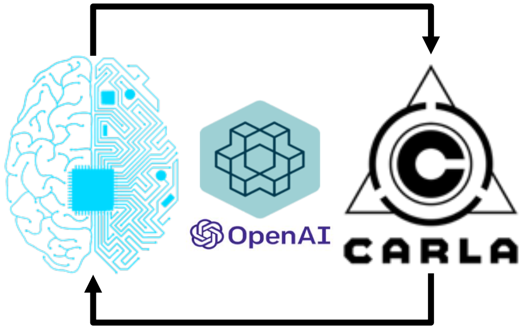


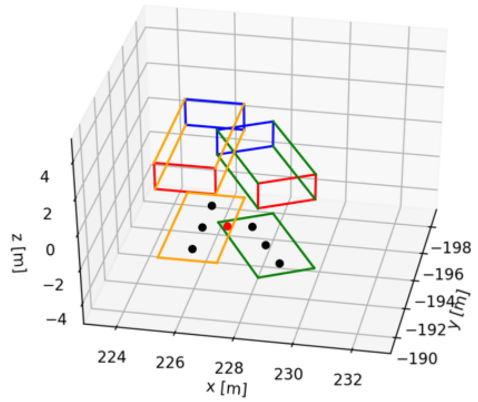
Case Study: Car Crash Scenario
Formalize the Problem

- Designing the setup:
  - Goal: define several skills (i.e., one agent = one skill) and train those low-level agents/skills in different scenarios/environments
  - Adhere to ethical rules
  - Train a high-level supervisor to select strategies in difficult situations
  - We can use other low-level controllers (e.g. MPC, or hard-coded rules)
- Reward process structure:
  - Many such problems were too hard to learn
  - We need to come up with a strategy to shape the rewards
    - *Curriculum Learning:*
      1. Go around the obstacle
      2. Go around the obstacle without collisions

Case Study: Car Crash Scenario
Setting up the simulator

- OpenAI gym
  - Standardized interface to develop and test RL applications
  - Many tools publicly available (tensorboard, algorithms, …)
- Use CARLA within the *OpenAI gym* environment
  - No native support as of today → build APIs
  - Get synchronized sensor data (HW-independent)
- Modification to CARLA
  - Physics engine does not provide the information we need
  - Cluster usage not out-of-the-box

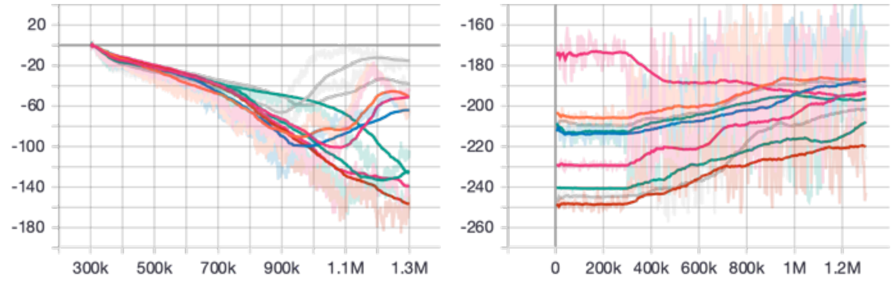


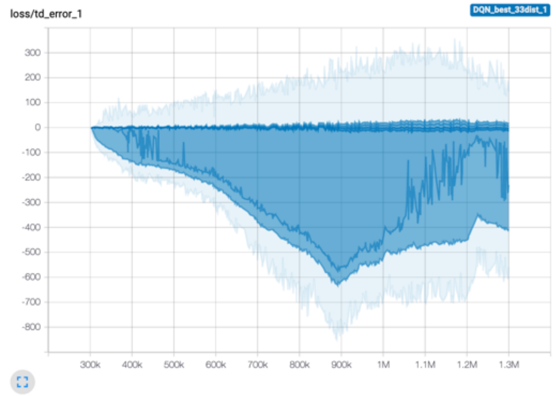


ADA LOVELACE CENTER Fraunhofer IIS

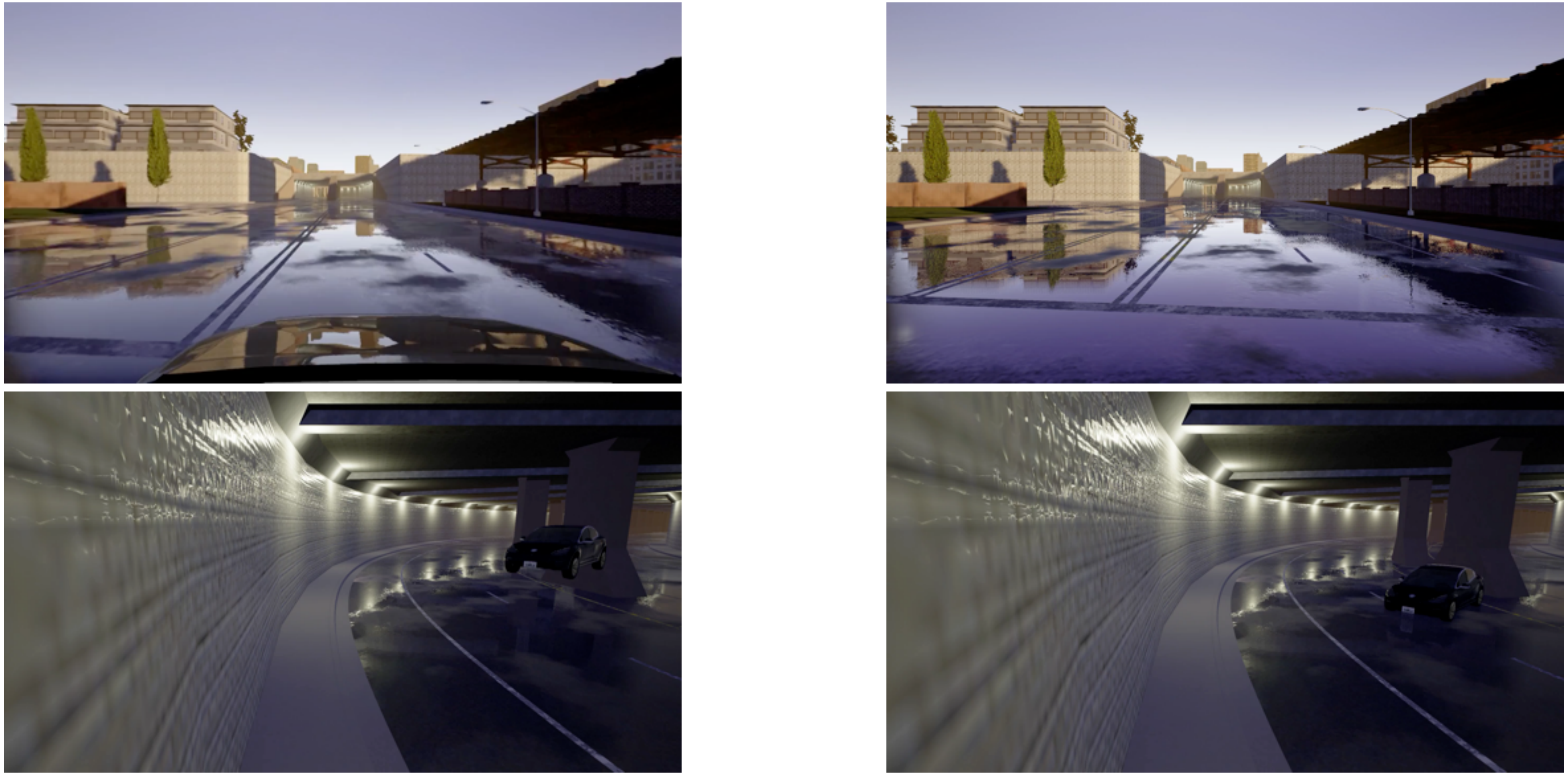
Case Study: Car Crash Scenario
Training the agents

- **Question #1: preprocessed state space or raw images?**
  - We implemented and tested both
  - Requires and good training and testing
  - Outcome: images contain information that we should use in combination with other sensors and already implemented pipelines
- **Question #2: discrete or continuous actions? which algorithm?**
  - *Continuous:* PPO2, DDPG/TD3, SAC; *Discrete:* PPO2, DDQN
  - Both converge to (roughly) the same solutions
- **Question #3: how to randomize your task?**
  - Not explicitly but implicitly by the simulator (low-level randomization)
  - Task-level randomization requires careful definition
- **Question #4: how to select among the available skills?**



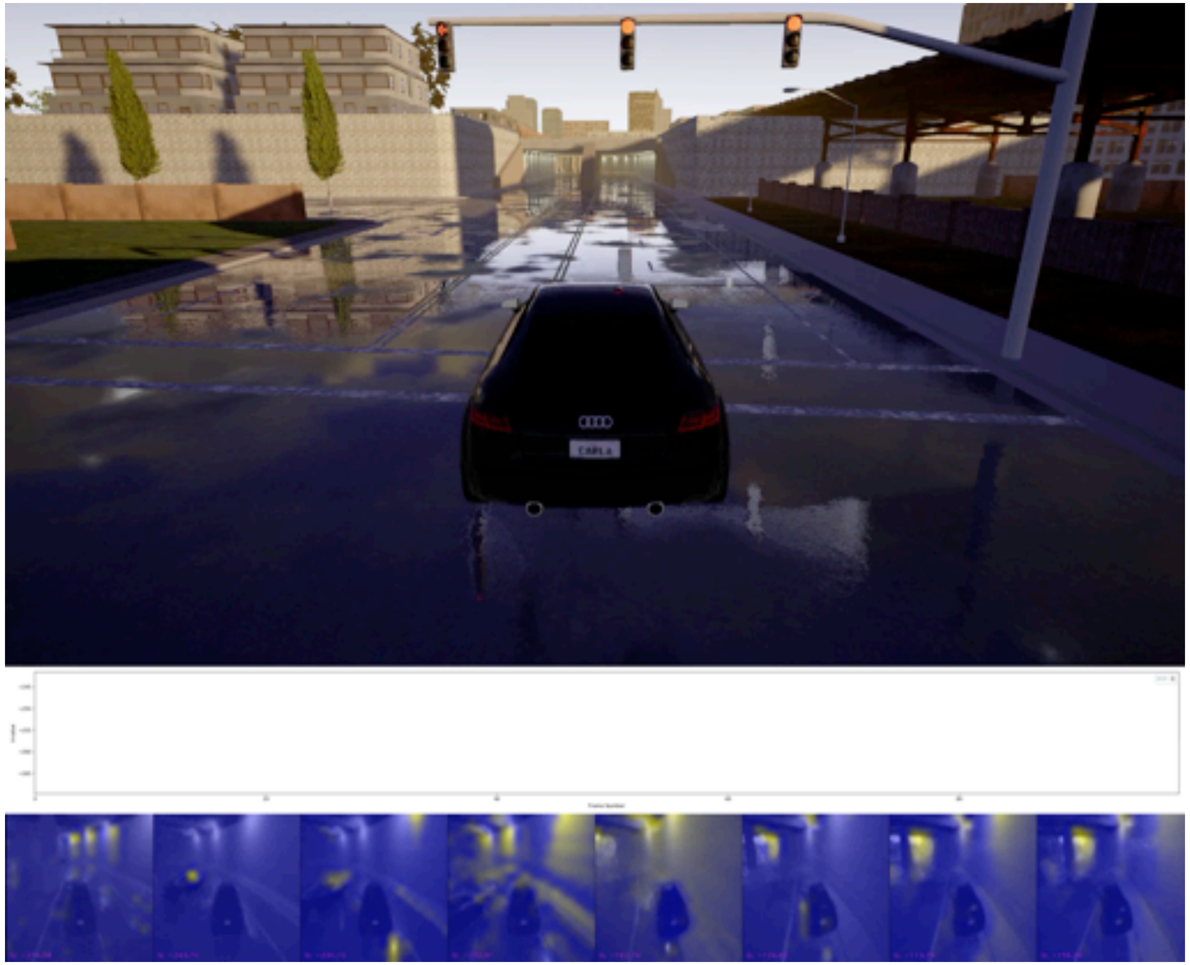


Case Study: Car Crash Scenario
Validating the agents

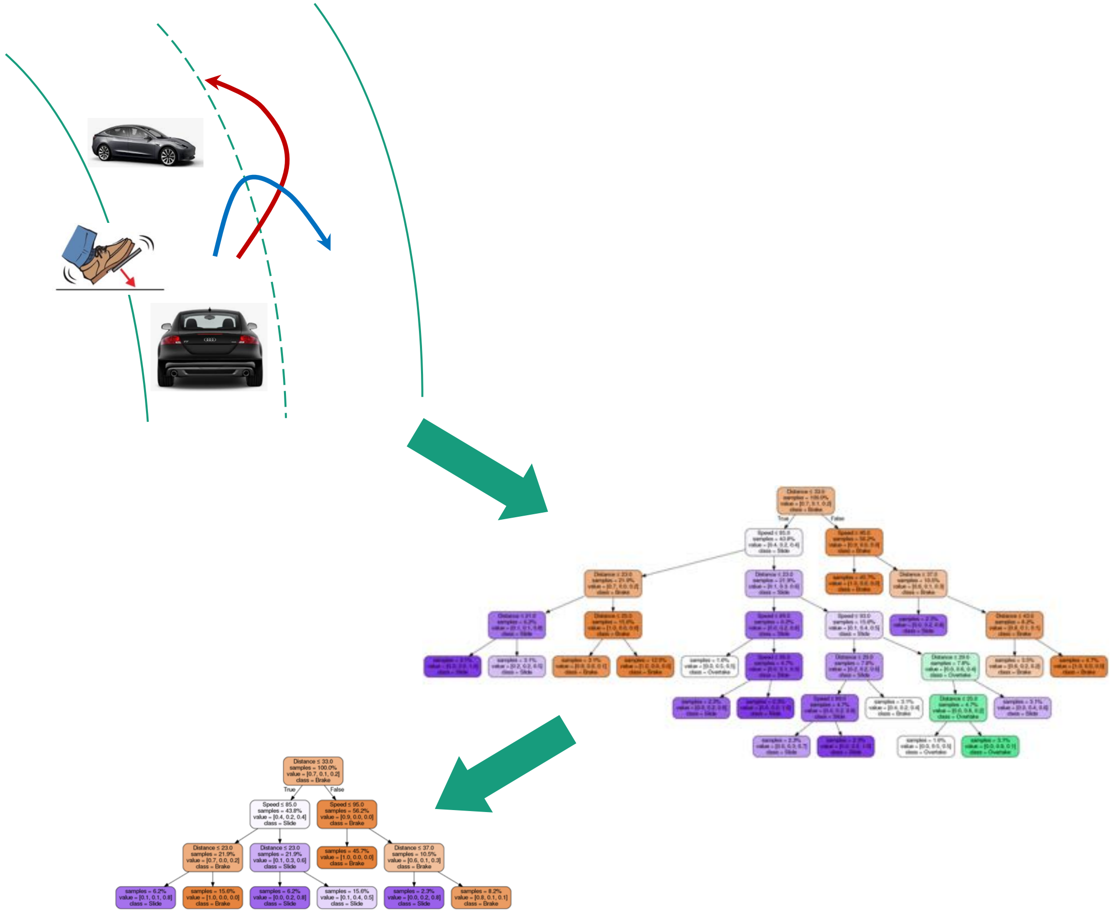


Case Study: Car Crash Scenario
Validating the agents

(low-level controller)



(high-level controller)



Case Study: Car Crash Scenario
Deployment

- Long story short: we did not do this yet ☺
  - Someone in the audience has a car and a racetrack for us?
- Usually with deployment of ML/RL models we see unexpected behavior of agents
- This might have various reasons:
  - The simulator is not true replica of the real world and we did not randomize enough
  - The agent exploits simulator mismatch (it ”cheats”)
  - Representation of sensors is unstable/different
  - …

Challenges

Current Challenges in Reinforcement Learning

Continual Learning	Explainability	A-priori Knowledge	Sample Efficiency
(Simulator) Mismatch	Sensor Degradation	Safe Exploration	Adversaries

Summary

- Continual Learning is a cross-sectional discipline in machine learning
- It is better understood as a requirement that aims for a bag of wishes:
  - Meta-learning, Few shot learning, Lifelong Learning, Multi-Task Learning, Transfer Learning
- RL can be considered as one method that inherently uses the concept of continual learning by design
- But there are also other powerful frameworks such as Bayesian Optimization, Online SVR, Online Convex Optimization…
- Applications at Fraunhofer include:
  - Adaptive Sensorics (re-configurating measurement devices; measurement planning)
  - Adapting models to changed environments (e.g. radio-based localization)
  - Optimizing material flow, logistics, …