Exploring deep reinforcement learning for real-world autonomous driving systems

navya
be fluid

Victor Talpaert(1), Ibrahim Sobh(2), **B Ravi Kiran(3)**, Patrick Mannion(4), Senthil Yogamani(5), Ahmad El-Sallab(2), Patrick Perez(7)

(1) U2IS, ENSTA ParisTech, Palaiseau, AKKA Technologies (2) Valeo Egypt, Cairo **(3) Navya Labs**, **Paris** (4) Galway-Mayo Institute of Technology, Ireland (5) Valeo Vision Systems, Ireland (7) Valeo.ai, France

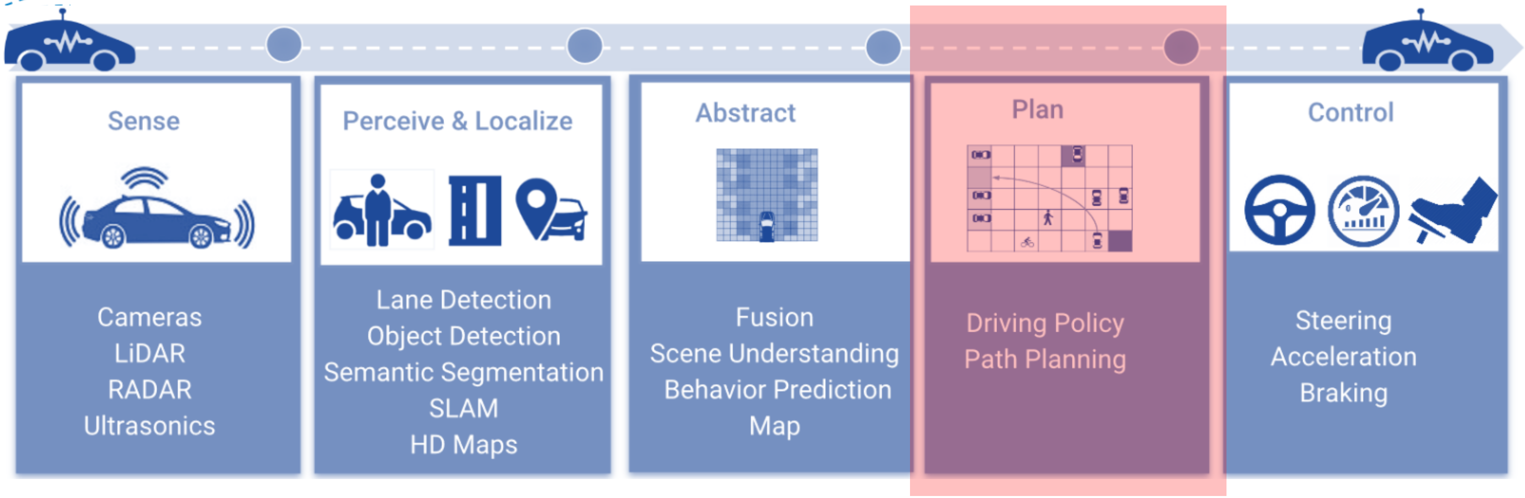
ravi.kiran@navya.tech

100%Autonomous 100%Driverless 100%Electric

OVERVIEW

- Quick overview of Reinforcement learning
  - Taxonomy of autonomous driving tasks
  - History & Applications
  - Taxonomy of methods in RL today
- Autonomous Driving Tasks
  - Which tasks require reinforcement learning
  - Which tasks require Inverse reinforcement learning
  - Role of simulators
- Challenges in RL for Autonomous driving
  - Designing reward functions, Sparse rewards, scalar reward functions
  - Long tail effect, Sample efficient RL/IL
  - Moving from Simulation to reality
  - Validating, testing and safety
- Conclusion
  - Current solutions in deployment in industry
  - Summary and open questions

AUTONOMOUS DRIVING



Scene interpretation tasks :
- 2D, 3D Object detection & tracking
- Traffic light/traffic sign
- Semantic segmentation
- Free/Drive space estimation
- Lane extraction
- HD Maps : 3D map, Lanes, Road topology
- Crowd sourced Maps

Fusions tasks:
- Multimodal sensor fusion
- Odometry
- Localization
- Landmark extraction
- Relocalization with HD Maps

Reinforcement learning tasks:
- Controller optimization
- Path planning and Trajectory optimization
- Motion and dynamic path planning
- High level driving policy : Highway, intersections, merges
- Actor (pedestrian/vehicles) prediction
- Safety and risk estimation

★★★☆☆

WHAT IS REINFORCEMENT LEARNING

Assessment
State-Action Evaluation
Other supervisor streams

reward

Real world / Simulator

state

RL Agent
States-to-Actions (Policy)
$\pi : \mathcal{S} \rightarrow \mathcal{A}$

Actions

Environment
Sensor stream
Cameras/Lidars/Radars

Learning what to do—how to map situations to actions optimally : an optimal policy*

*Maximization of the expected value of the cumulative sum of a received scalar reward

MACHINE LEARNING AND AI TODAY

Supervised Learning

Given input examples (X, Y)

Learn implicit function approximation

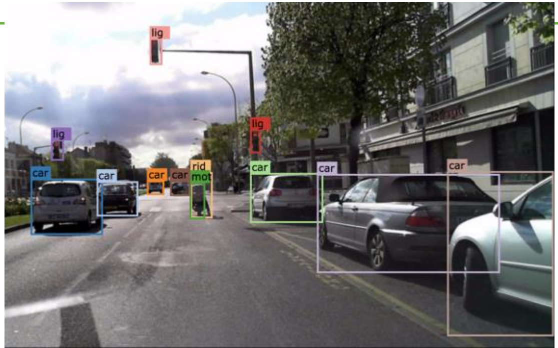
f: X → Y

(X: images) to (*Y*: class label)

$$R_{\text{emp}}(f) = \frac{1}{N}\sum_i L(f(x_i), y_i)$$

Empircal risk (loss function) : representing the price paid for inaccurate prediction

Predictions do not affect environment
(Samples are IID)



Reinforcement learning

Given input state space , rewards, transitions

Learn a policy from state-to-actions

π: S → A

(S vehicle state, images, A : speed, direction)

$$\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$$

Value function : long-term reward achieved

Predictions affect both what is observed as well as future rewards

Requires Exploration, learning and interaction

STATE SPACE, ACTIONS AND REWARDS

Vehicle state space

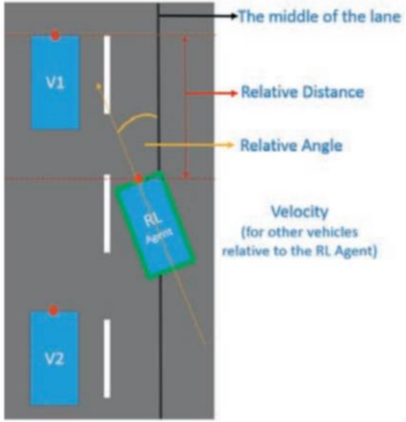
- Geometry (vehicle size, occupancy grid)
- Road topology and curvature
- Traffic Signs and laws
- Vehicle pose and velocity(v)
- Configuration of obstacle (with poses/v)
- Drivable zone

Actions

- Continuous control : Speed, steering
- Discrete control : up, down, left, right, …
- High level (temporal abstraction) : slow down, follow, exit route, merge

Reward (positive/negative)

- Distances to obstacles (real)
- Lateral error from trajectory (real)
- Longitudinal : Time to collision (real)
- Percentage of car on the road (sim)
- Variation in speed profile (real)
- Actor/Agent intentions in the scene
- Damage to vehicle/other agents (sim)



[Reinforcement Learning for Autonomous Maneuvering in Highway Scenarios](#)
[A Survey of State-Action Representations for Autonomous Driving](#)

ORIGINS OF REINFORCEMENT LEARNING

1950s
Optimal Control
Pontryagin
Bellman

1960s
Dynamic Programming
stochastic optimal control
Richard Bellman

1930s-70s
Trial/Error Learning
Psychology, Woodworth
Credit Assignment Minsky
Least Mean Squares (LMS)
Widrow-Gupta
Learning Automata
K-armed bandits

1980s
Temporal Difference
R. Sutton Thesis

1990s
Q-Learning, Neuro-DP (DP+ANNs)
Bertsekas/Tsitsiklis

2000s
Policy Gradient Methods
Sutton et al.

2006s
Monte-Carlo Tree Search for RL on Game tree for Go
Rémi Coulom & others

2005s
Neural Fitted Q Iteration
Martin Riedmiller

2014
Deterministic Policy Gradient Algorithms
David Silver et al.

2015
Playing Atari with Deep Reinforcement Learning
Deepmind Mnih et al.

2016
Asynchronous Deep RL methods A2C, A3C
Deepmind Mnih et al.

2015-2019
AlphaGo,AlphaZero
MCTS+DeepRL
Go Chess Shogi
DeepMind

AlphaStar
OpenAI Dota

Adaptive signal processing
Stochastic approximation theory
Animal Psychology and neuroscience
Robotics and Control theory

TERMINOLOGIES

MDP

$\mathcal{S}$: Set of States(discrete/continous)

$\mathcal{A}$: Set of Actions

$\mathcal{P}(s, s')$: Transition probabilities

$r(s, a)$: Reward at state given an action

assumption s_{t+1} is conditionally independant of the past states/actions give s_t, a_t

Reinforcement Learning

Prediction
Evaluation of Policy

Control
Infer optimal policy
(policy/value iteration)

*** Model Based** (P & R known)
Control : Policy/Value Iteration
Prediction : Policy Optimization

Model Free (P and R unknown)
Prediction : MonteCarlo(MC), TimeDifference(TD)
Control : MC control step
Q Learning

*** On policy methods**
Learning using current policy

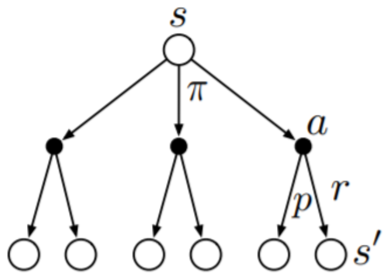
Off policy methods
Learn from observations

Action spaces
Continous (vehicle controller)
Discrete (Go, Chess)

Markovian Decisions Processes
(MDP, POMPD)

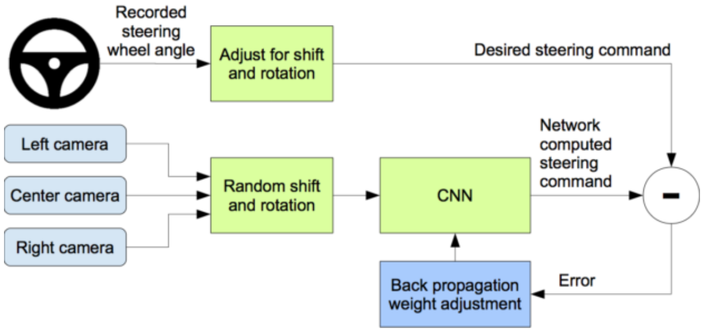
Exploration
Learning (P, R, Policy) using current policy

Exploitation
Learn Policy given
(P, R, Value function)



Backup diagram for v_π

BEHAVIORAL CLONING/ IMITATION LEARNING (IL) ≠RL



State-action map as supervised learning

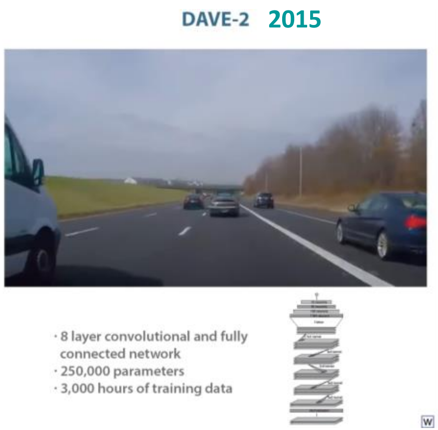
- Directly map (Inputs/states) TO (Outputs/Actions) control and ignore IID assumption, no more a sequential decision process.
- Also known as end-to-end learning since sensor streams are directly mapped to control

Issues :

- Agent mimics expert behavior at the danger of not recovering from unseen scenarios such as unseen driver behavior, vehicle orientations, adversarial behavior of agents (overfits expert)
- Poorly defined reward functions cause poor exploration
- Requires huge No. (>30M samples) of human expert samples

Improvements :

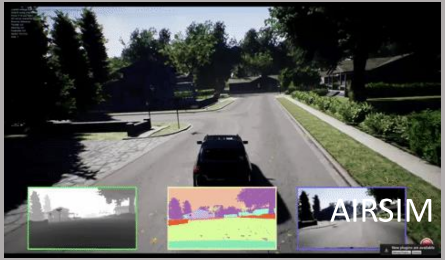
- Heuristics to improve data collection in corner cases (Dagger)
- Imitation is efficient in practice and still an alternative

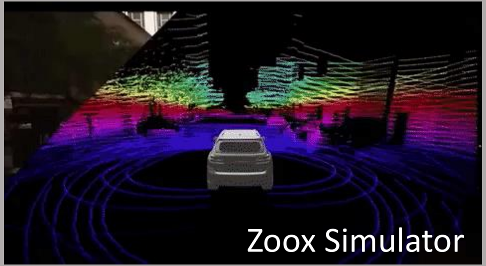


[DAGGER : A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning]()
[Hierarchical Imitation and Reinforcement Learning]()

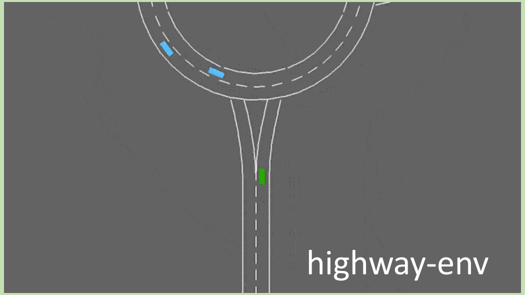
SIMULATORS : ENVIRONMENT FOR RL

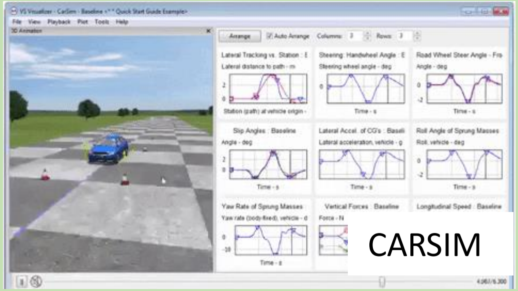
Perception Stream Simulators for End-to-End Learning



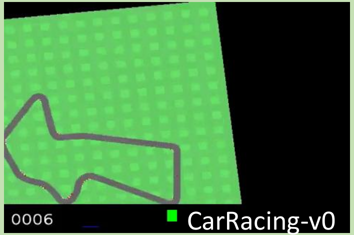


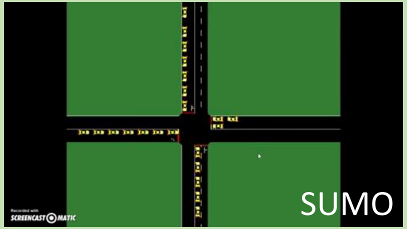
Motion planning & traffic simulators





Vehicle state, reward, damage

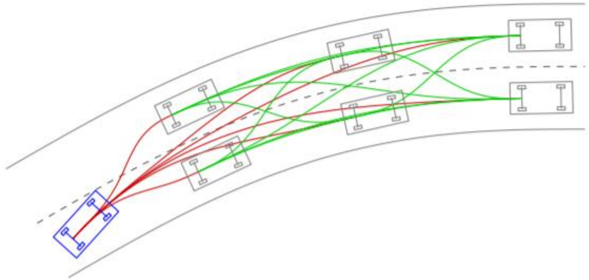




Audi partners with Israel's autonomous vehicle simulation startup Cognata

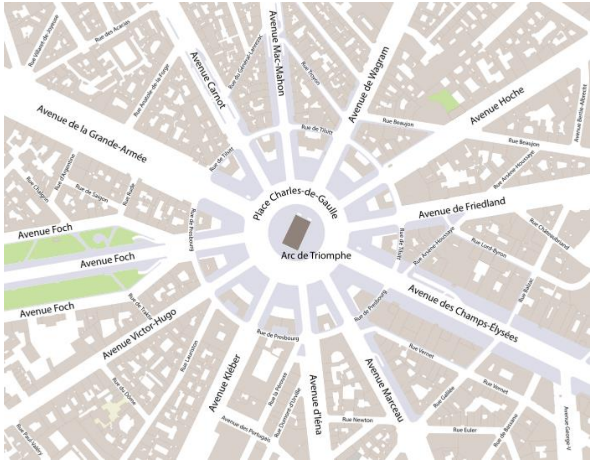
Learning vehicle controllers

- For well defined tasks : Lane following, ACC classical solutions (MPC) are good
- Tuning/choosing better controllers based on vehicle and state dynamics is where RL can be impactful
- ACC and braking assistance



Path planning and trajectory optimization

- Choose path that minimizes certain cost function
  - Lane following, Jerk minimizer
- Actor (pedestrian/vehicle) behavior prediction



Decision making in complex scenarios:

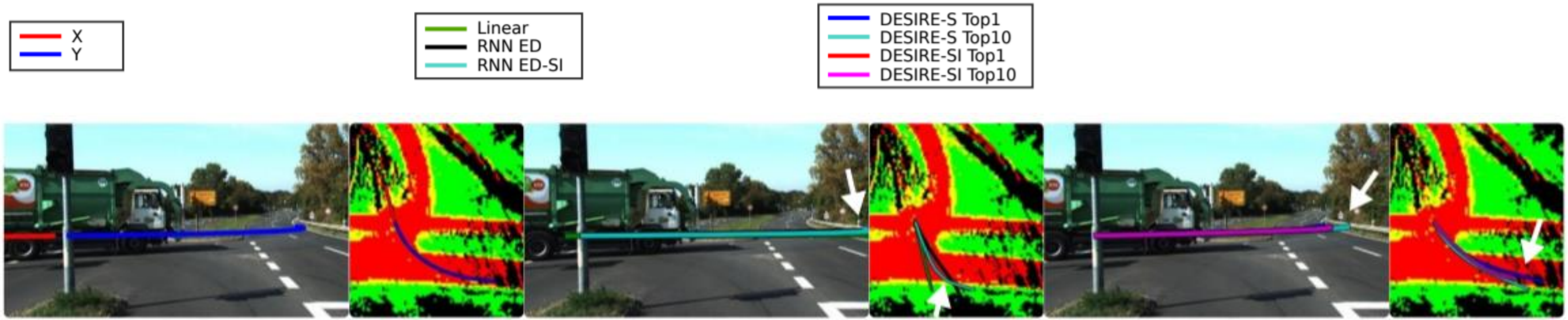
- Highways driving : Large space of obstacle configurations, translation/orientations/velocity, Rule based methods fail
- Negotiating intersections : Dynamic Path Planning
- Merge into traffic, Split out from traffic

[Planning algorithms, Steven Lavalle](#)
[Real-time motion planning](#)

Inverse RL or Inverse Optical control

- **Given** States, Action space and Roll-outs from Expert policy, Mode of the environment (State dynamics)
- **Goal** : Learn reward function, Then learn a new policy
- **Challenges** : not well defined, tough to evaluate optimal reward
- **Applications :** Predicting pedestrian, vehicles behavior on road, Basic Lane following and obstacle avoidance

it is commonly assumed that the purpose of observation is to learn policy, i.e. a direct representation of mapping from states to actions. We propose instead to recover the experts reward function and use this to generate desirable behavior. We suggest that the **reward function offers much more parsimonious description of behavior**. After all the entire field of RL is founded on the presupposition that the reward function, rather than the policy is the most succint, robust and transferable definition of the task.
Algorithms for Inverse Reinforcement Learning, Ng Russel 2000



DESIRE: Distant Future Prediction in Dynamic Scenes with Interacting Agents

CHALLENGES IN REWARD FUNCTION DESIGN

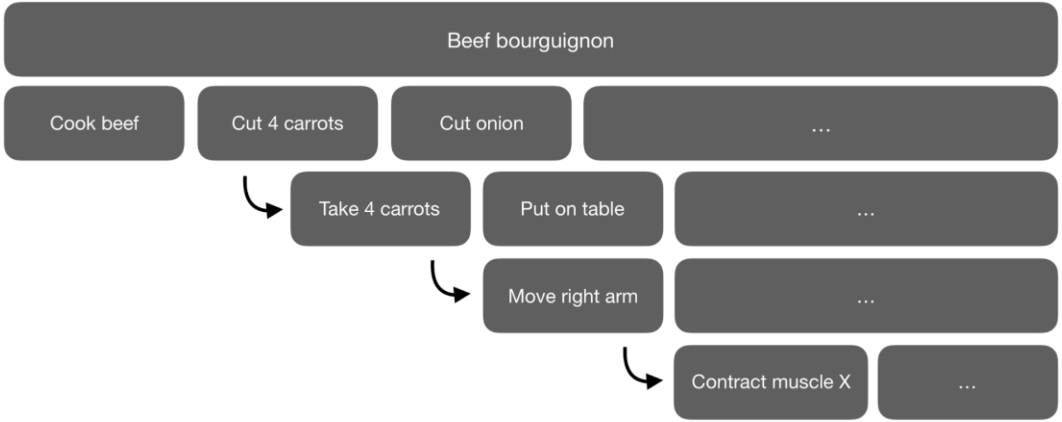
- Where do rewards come from ?
  - Simulations (low cost to high cost based on dynamics and details required)
  - Large sample complexity
  - Positive rewards without negative rewards can have dangerous consequences
  - Real World (very costly, and dangerous when agent requires to explore)
- Temporal abstraction
  - Credit assignment and Exploration-Exploitation Dilemma
  - Cobra effect : RL algorithms are blind maximizers of expected reward
- Other ways to learn a reward
  - Decompose the problem in multilple subproblems which are easier.
  - Guide the training of problems with expert supervision using imitation learning as initialization
  - Reduce the hype and embrace the inherent problems with RL : Use Domain knowledge

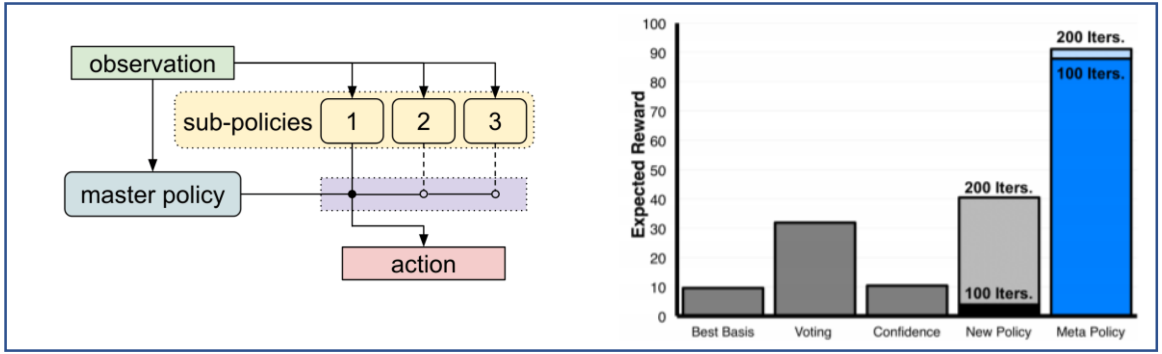
Cobra effect : The British government was concerned about the number of venomous cobra snakes in Delhi. They offered a reward for every dead cobra. Initially this was a success as large numbers of snakes were killed for the reward. Eventually, however, enterprising people began to breed cobras for the income. When the government became aware of this, the reward program was scrapped, causing the cobra breeders to set the now-worthless snakes free. As a result, the wild cobra population further increased. The apparent solution for the problem made the situation even worse.

https://www.alexirpan.com/2018/02/14/rl-hard.html

CHALLENGES IN REWARD FUNCTION DESIGN

- Hierarchy of tasks
  - Decompose the problem in multilple subproblems which are easier.
  - Combining learnt policies
- Guide training for complex problems
  - Train on principle task, then subtasks
  - expert supervision using imitation learning as initialization

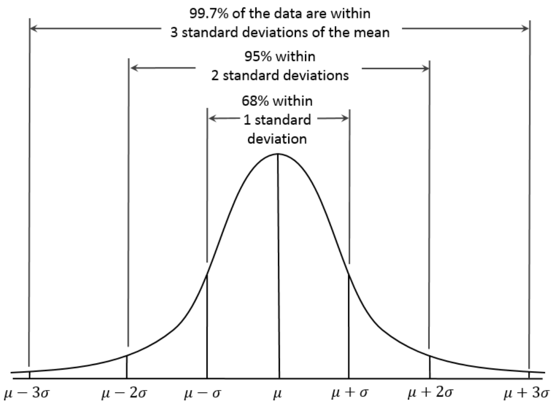




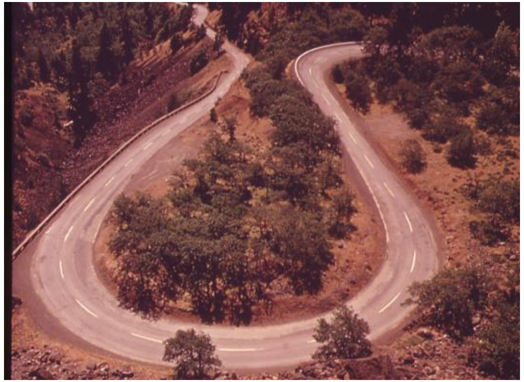
https://thegradient.pub/the-promise-of-hierarchical-reinforcement-learning/
Composing Meta-Policies for Autonomous Driving Using Hierarchical Deep Reinforcement Learning

LONG TAIL DRIVER POLICY

- Rare and adversarial scenarios are difficult to learn
  - Core issue with safe deployment of autonomous driving systems
  - Models perform well for the average case but scale poorly due to low frequency, as well as sparse rewards



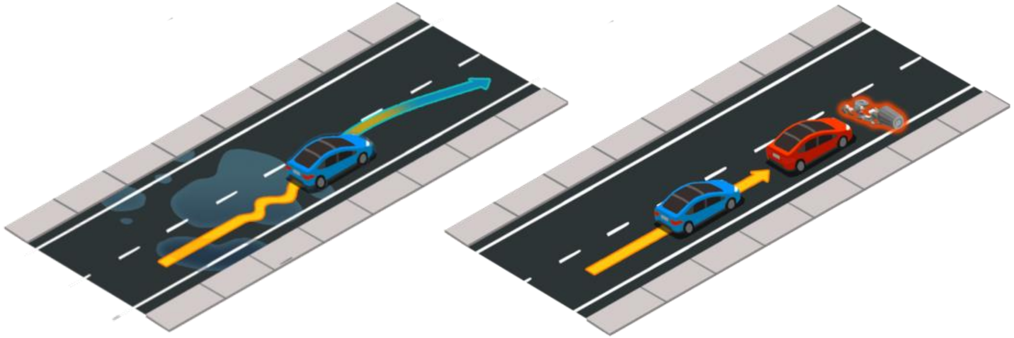
- Hairpin bends, U-Turns
  - Rare with difficult to model state space dynamics

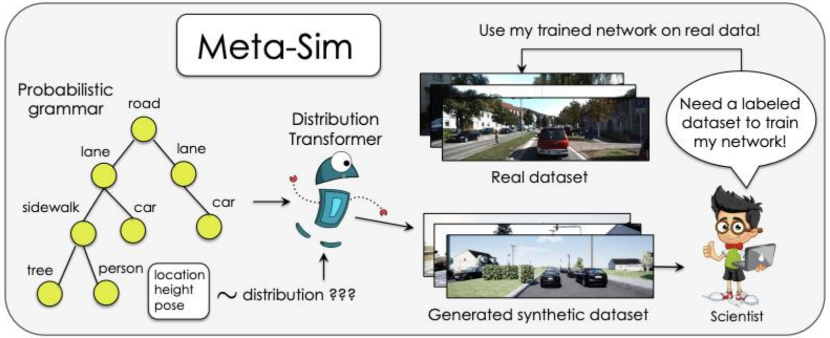


- Create simpler small sample based models blended with average case model
  - cures symptom not the disease

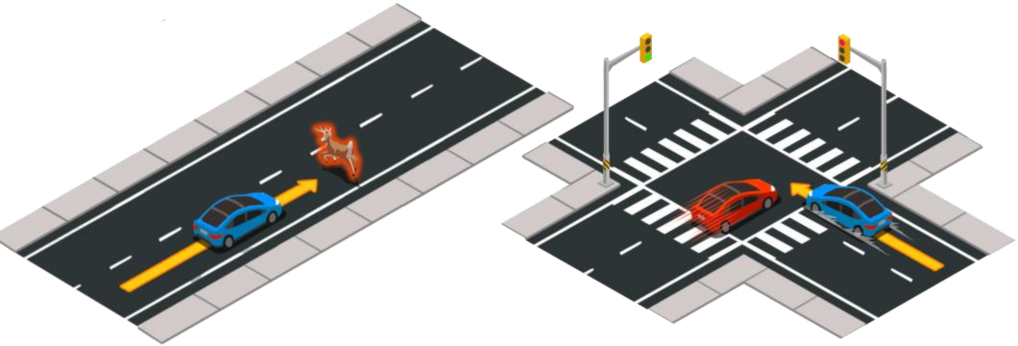
[Drago Anguelov (Waymo) - MIT Self-Driving Cars]

SCENARIO GENERATION FOR DRIVING SCENARIOS





https://nv-tlabs.github.io/meta-sim/#



Carla Challenge 2019

CHALLENGES SIMULATION-REALITY GAP

Handling domain transfer

WAYVE

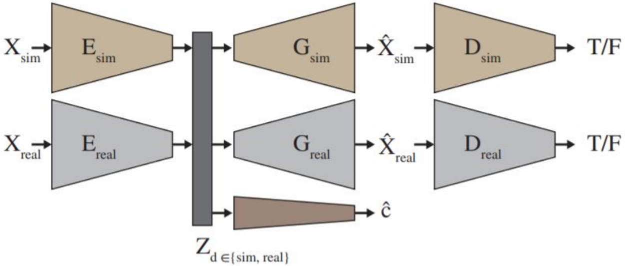
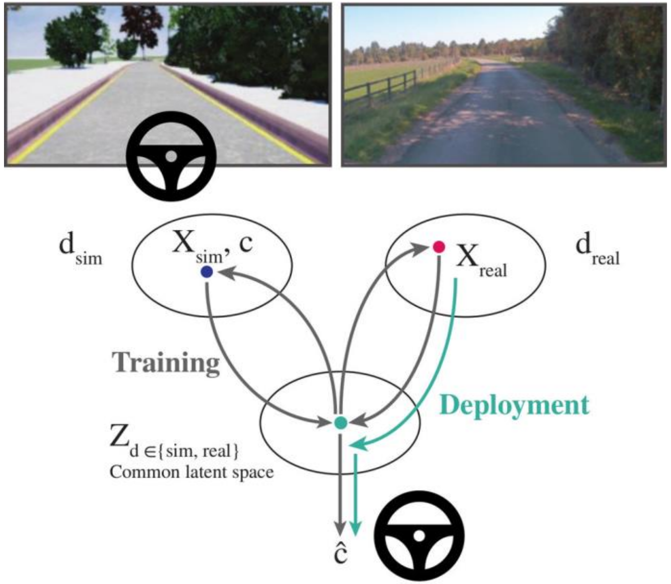


Fig. 2: Model architecture for domain-transfer from a simulated domain to real-world imagery, jointly learning control and domain translation. The encoders $E_{sim,real}$ map input images from their respective domains to a latent space Z which is used for predicting vehicle controls $\hat{c}$. This common latent space is learned through direct and cyclic losses as part of learning image-to-image translation, indicated conceptually in Figure 3 and in Section III-B.

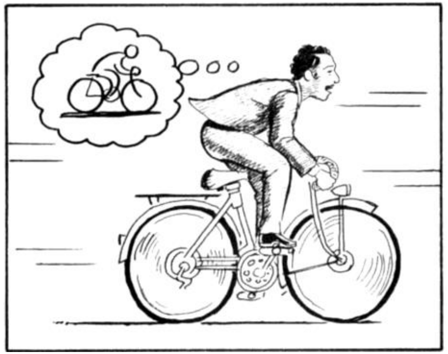


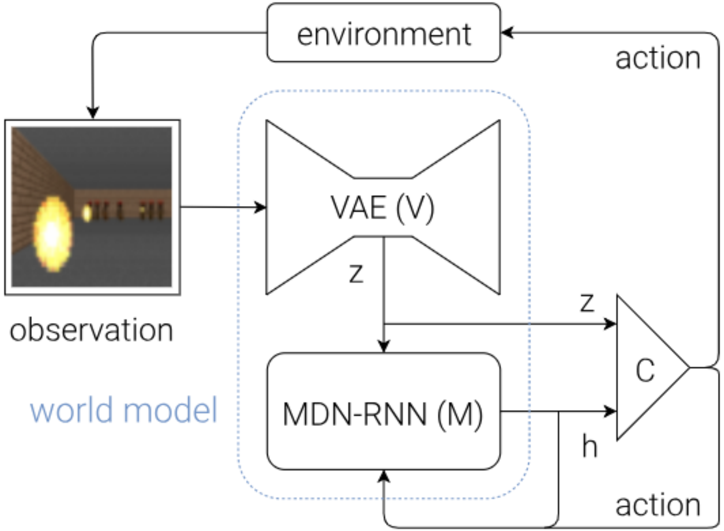
- How to create a simulated environment which both faithfully emulates the real world and allows the agent in the simulation to gain valuable real-world experience?
- Can we map Real world images to Simulation ?

nauya

CHALLENGES SIMULATION-REALITY GAP

Learning a generative model of reality



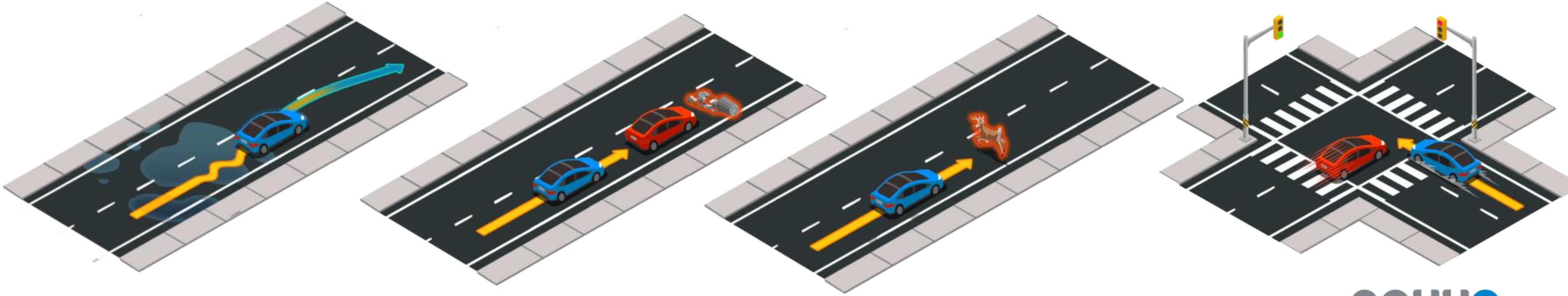


World models enable agents to construct latent space representations of the dynamics of the world, while building/learning a robust control/actuator module over this representation.

https://worldmodels.github.io/

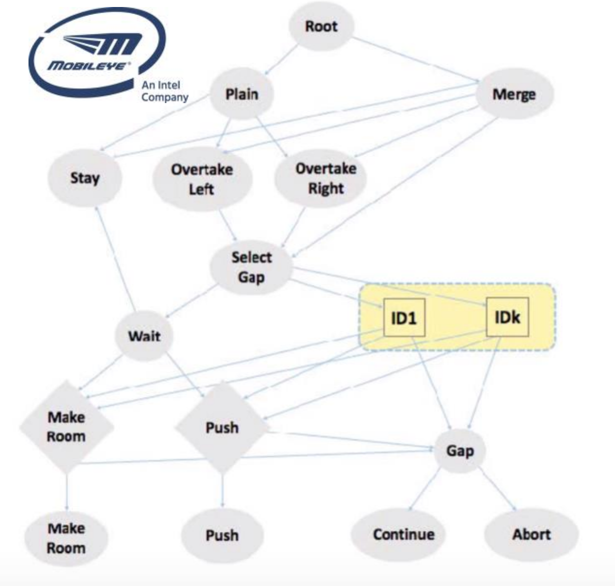
CHALLENGES: SAFETY AND REPRODUCIBILITY

- Safe policies for autonomous agent
  - **SafeDAgger** : safety policy that learns to predict the error made by a primary policy w.r.t reference policy.
  - Define a feasible set of core safe state spaces that can be increamentally grown with explorations
- Reproducible (code) on benchmark
  - variance intrinsic to the methods hyperparameters init.
  - Cross-validation for RL is not well defined as opposed to supervised learning problems
- Future standardized benchmarks
  - Evaluating autonomous vehicle control algorithms even before agent leaves for real world testing.
  - NHTSA-inspired pre-crash scenarios : Control loss without previous action, Longitudinal control after leading vehicle's brake, Crossing traffic running a red light at an intersection, and many others
  - Inspiration from the Aeronautics community on risk

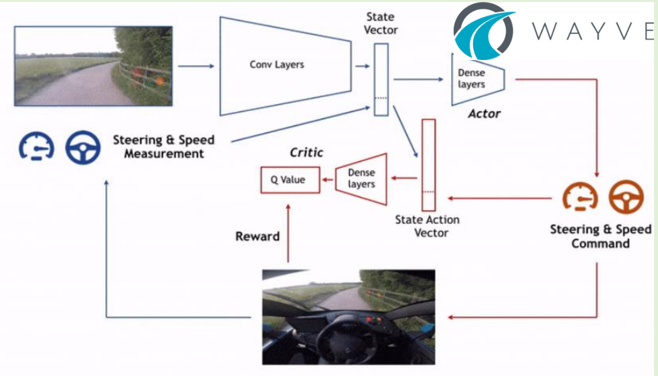


Carla Challenge 2019

DRLAD : MODERN DAY DEPLOYMENTS

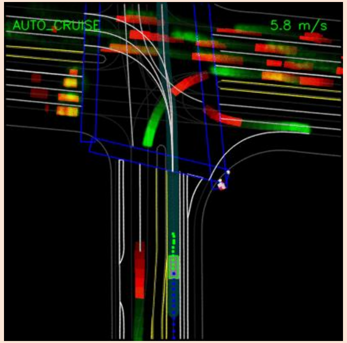


Options Graph :



Agent maximizes the reward of distance travelled before intervention by a safety driver.

Recovering from a Trajectory Perturbation



Future Trajectory Prediction on Logged Data

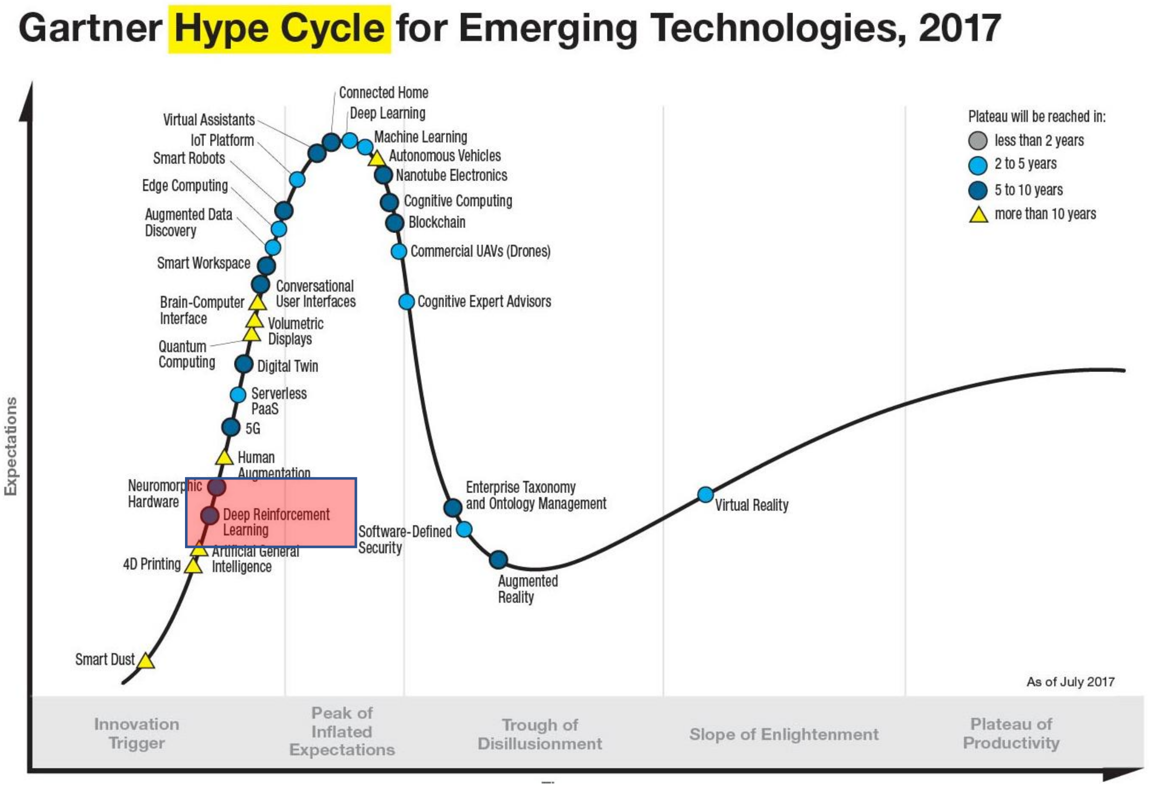
WAYMO

Robust Imitation learning using perturbations and simulated expert variations and aumented imitation loss function
https://sites.google.com/view/waymo-learn-to-drive/

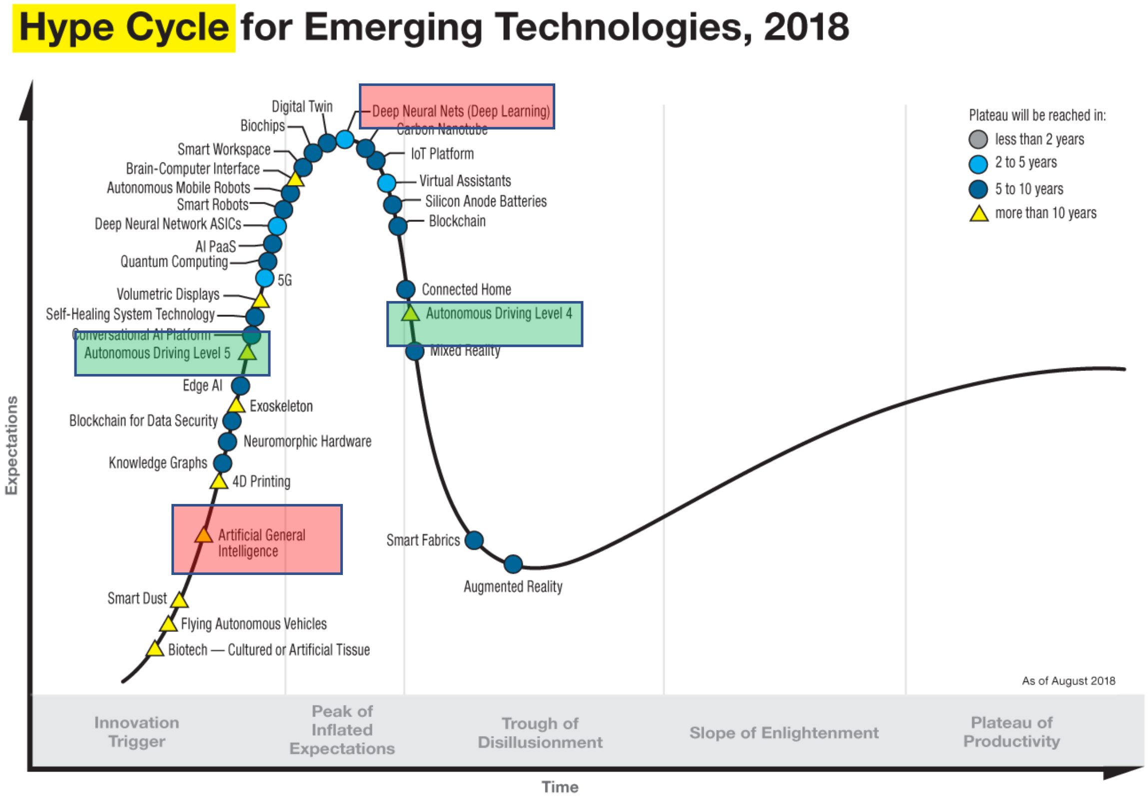
CONCLUSION

- How to design rewards ?
- How should the problem be decomposed to simplify learning an policy?
- How to train in different levels of simulations (efficiently)?
- How to handle long tail cases, especially risk intensive cases
- How to intelligently perform domain change from simulation to reality ?
- Can we use imitation to solve the problem before switching to Reinforcement Learning?
- How can we learning in a multi-agent setup to scale up learning?

WHERE IS THE HYPE ON DEEP RL



WHERE IS THE HYPE ON DEEP RL



Hypes are highly non-stationary

LECTURES AND SOURCES

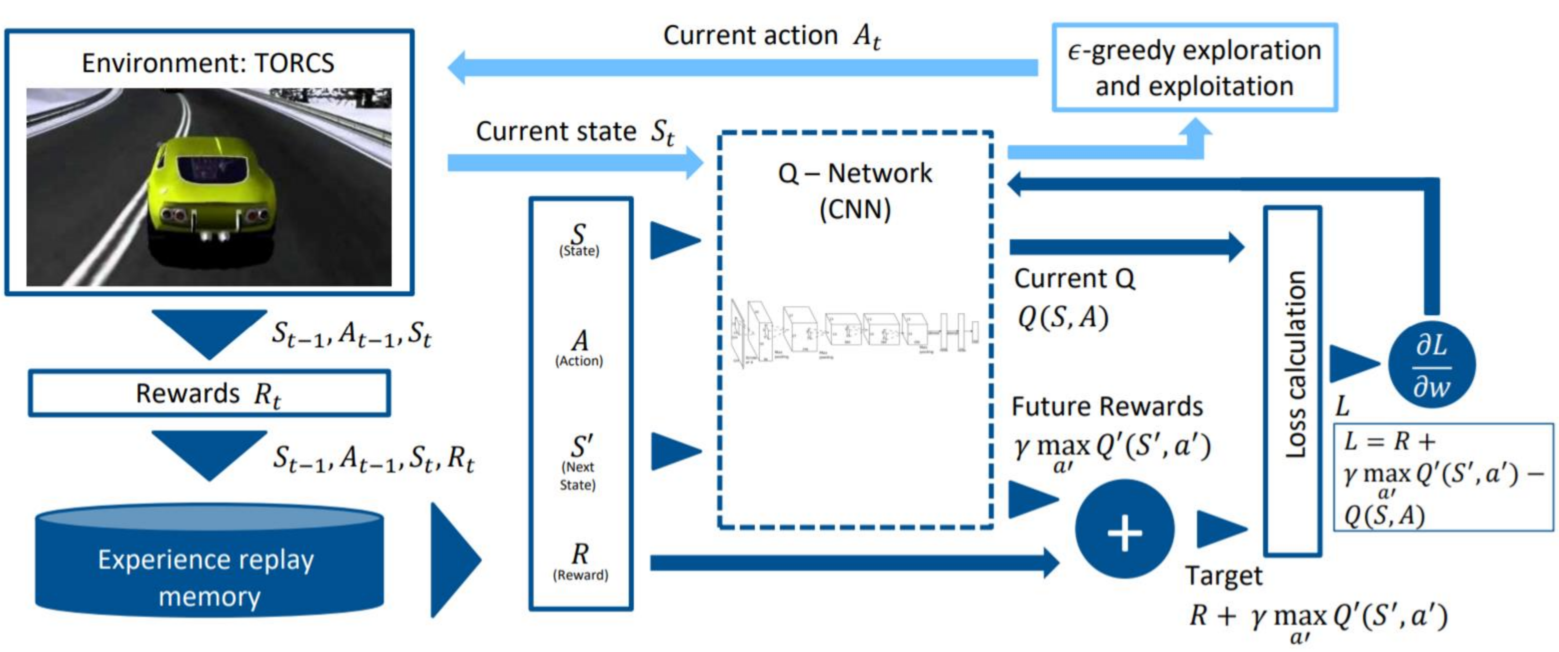
- Reinforcement Learning: An Introduction Sutton 2018 [book]
- David Silver’s RL Course 2015 [link]
- Berkeley Deep Reinforcement Learning [Course]
- Deep RL Bootcamp lectures Berkeley [Course]
- Reinforcement learning and optimal control : D P Bertsekas 2019 [book]

REFERENCES

- World Models Ha, Schimdhuber NeurIPS 2018 .
- Jianyu Chen, Zining Wang, and Masayoshi Tomizuka. "Deep Hierarchical Reinforcement Learning for Autonomous Driving with Distinct Behaviors". 2018 IEEE Intelligent Vehicles Symposium (IV).
- Peter Henderson et al. "Deep Reinforcement Learning That Matters". In: (AAAI-18), 2018.
- Andrew Y Ng, Stuart J Russell, et al. "Algorithms for inverse reinforcement learning
- Daniel Chi Kit Ngai and Nelson Hon Ching Yung. "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers"
- Stephane Ross and Drew Bagnell. "Efficient reductions for imitation learning". In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010
- Learning to Drive using Inverse Reinforcement Learning and Deep Q-Networks, Sahand Sharifzadeh et al.
- Shai Shalev-Shwartz, Shaked Shammah, and Amnon Shashua. Safe, multi-agent, reinforcement learning for autonomous driving 2016.
- Learning to Drive in a Day, Alex Kendall et al. 2018 Wayve
- ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst
- StreetLearn, Deepmind google
- A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research [pdf]
- Meta-Sim: Learning to Generate Synthetic Datasets [link][pdf] Sanja Fidler et al.
- Deep Reinforcement Learning in the Enterprise: Bridging the Gap from Games to Industry 2017 [link]

DEEP Q LEARNING

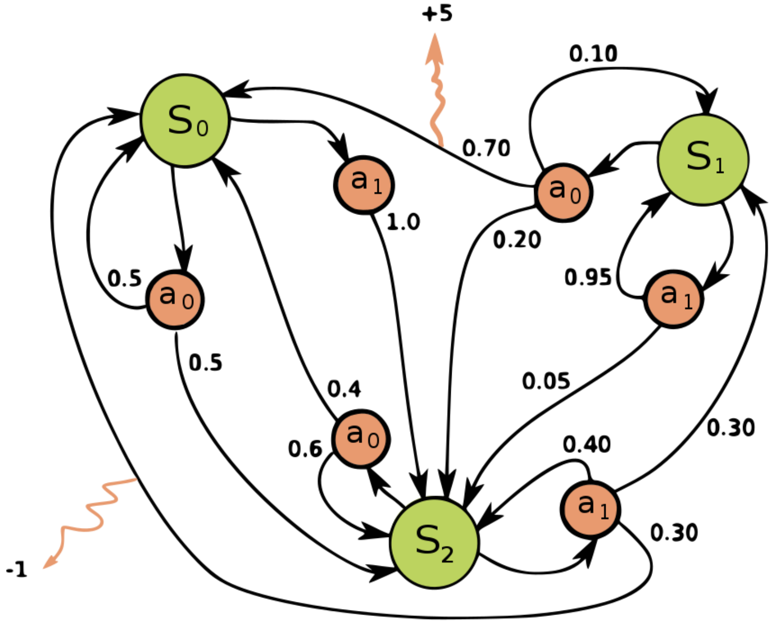
Autonomous driving agent in TORCS



MARKOV DECISION PROCESS

Markovian assumption on reward structure

s_{t+1} is conditionally independant of the past states/actions give s_t, a_t



MDP

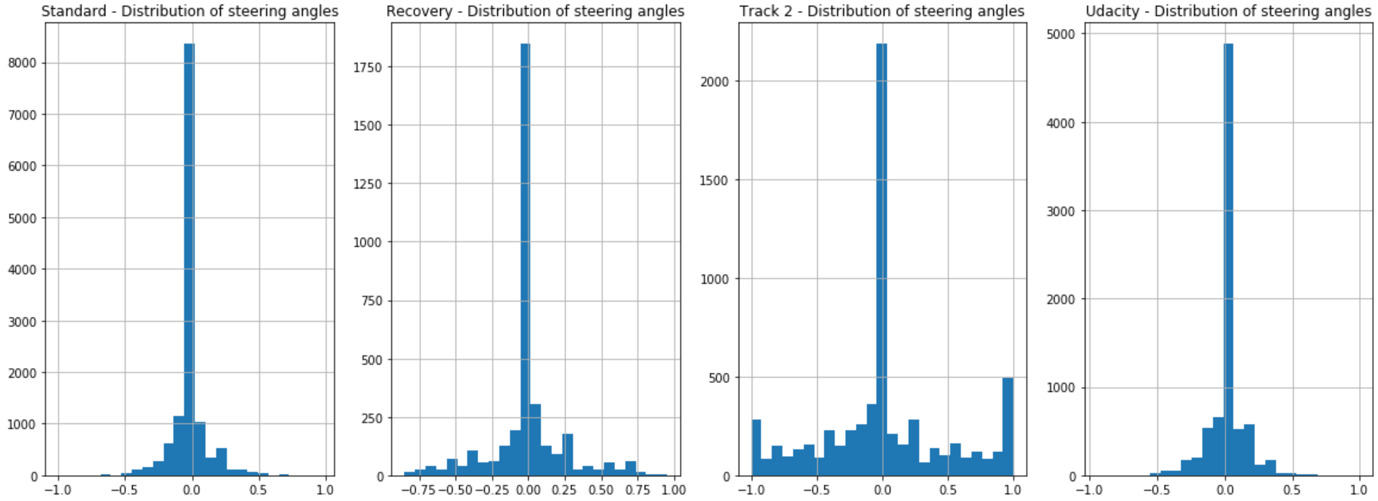
$\mathcal{S}$: Set of States(discrete/continous)

$\mathcal{A}$: Set of Actions

$\mathcal{P}(s, s')$: Transition probabilities

$r(s, a)$: Reward at state given an action

CHALLENGES IMITATION LEARNING



[Source](#) improving diversity of steering angle

- Learning from Demonstrations (LfD)/Imitation/Behavioral Cloning demonstrations are hard to collect
  - Measure the divergence between your expert and the current policy
  - Give priority in a replay buffer
  - Iteratively collect samples (DAgger)
  - Hierarchical Imitation reduce sample complexity by data aggregation by organizing the action spaces in a hierarchy