Imitation Learning summary

P. Alexoudi, Ch. Perchanidis, Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr
Version 2.0.1

Artificial Intelligence & Information Analysis Lab

Imitation Learning (IL)

- **Introduction to IL**
- Elements of IL
- Behavioral Cloning
- Direct Policy Learning
- Inverse Reinforcement Learning
- Challenges of IL
- IL Project in Unity
- IL in Autonomous Driving
- Cinematography Shooting

Introduction to IL

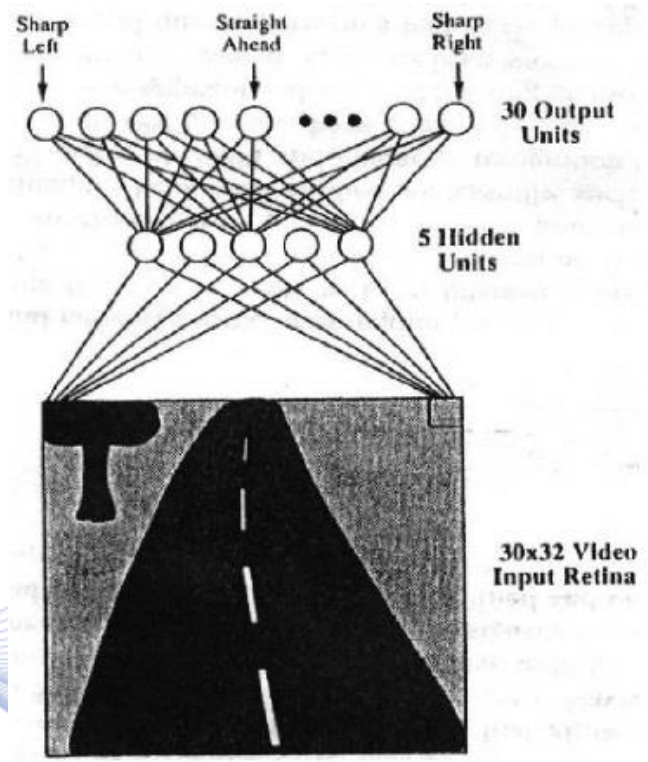
- Imitation Learning is a promising field of research with numerous imitation learning methods already being developed.
- The concept is to transfer human knowledge to an agent.
- The goal is to train a policy to mimic demonstrations. The agent tries to efficiently learn a desired behavior by imitating an expert's behavior.

Reference: [CAL-IL]

Relation with Reinforcement Learning

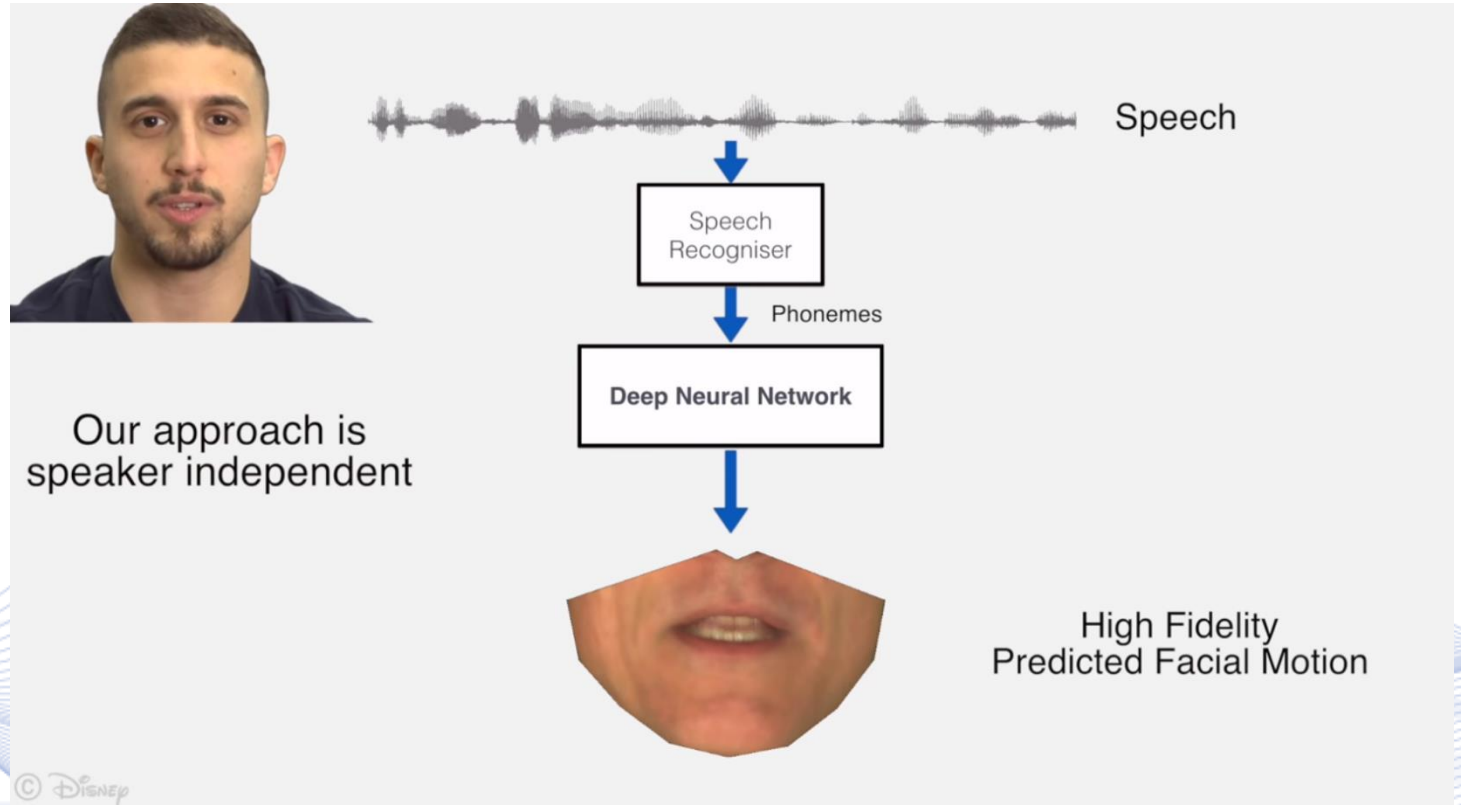
- Imitation Learning is closely related to Reinforcement Learning, which is one of the basic paradigms in Machine Learning.
- IL addresses basic problems of RL, such as sample-inefficiency, computational feasibility, speed and safety of training and reward function obscurity.
- IL is also based on the Markov Decision Process (MDP).
- Therefore knowledge of RL is necessary for IL understanding.

An Autonomous Land Vehicle In a Neural Network



Reference: [POM1989]

Deep Learning Approach for Generalized Speech Animation



Reference: [TAY2017]

Imitation Learning (IL)

- Introduction to IL
- **Elements of IL**
- Behavioral Cloning
- Direct Policy Learning
- Inverse Reinforcement Learning
- Challenges of IL
- IL Project in Unity
- IL in Autonomous Driving
- Cinematography Shooting

Elements of IL

- **Environment/Simulator**: The main element of IL is the environment, which is basically a Markov Decision Process (MDP). The environment includes:
  - State (s).
  - Action (a).

- **Policy** ($\pi_{\boldsymbol{\theta}}$): A policy is a function that ($\boldsymbol{\theta}$ is a policy parameter vector):
  - Maps states to actions: $\pi_{\boldsymbol{\theta}}(s) \rightarrow a$.
  - Or distributions over actions: $\pi_{\boldsymbol{\theta}}(s) \rightarrow P(a)$.

Elements of IL

- **Transition model**: It is the probability that taking action a in state s will lead to state s' on the next time instance:

$$P(s'|s,a).$$

- **Demonstrations/Demonstrator**: The actions of the agent are based on the expert's "optimal" policy, also known as trajectories:

$$\boldsymbol{\tau} = (s_0, a_0, s_1, a_1, \dots).$$

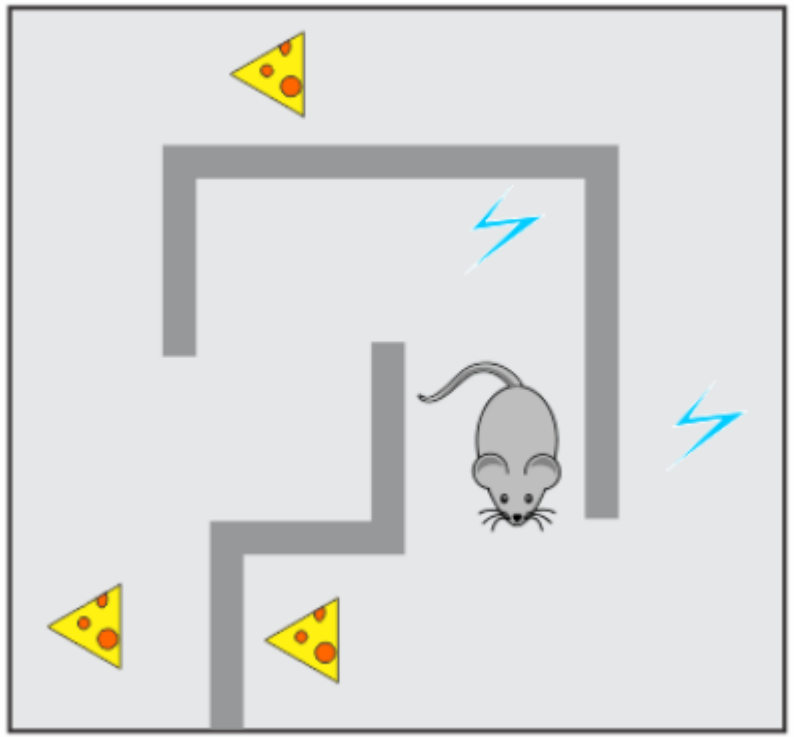
- In Imitation Learning the reward function r is not explicitly defined.

Elements of IL

- Distribution of trajectories induced by a policy: $P(\tau|\pi)$
  - Sample s_0 from P_0 (distribution over initial states), initialize $n = 1$
  - Sample action a_i from $\pi(s_{\tau-1})$
  - Sample next state s_τ from applying a_τ to $s_{\tau-1}$ (requires access to the environment)
  - Repeat from Step 2 with $n = n + 1$

- Distribution of states induced by a policy: $P(s|\pi)$
  - Let $P_\tau(s|\pi)$ denote distribution over n^{th} state
  - $P(s|\pi) = \frac{1}{N} \cdot \sum n \cdot P_\tau(s|\pi)$

Mouse in a Maze



Reference: TDS Not found??

- **State**: Coordinates of Mouse in Maze
- **Action**: Chosen movement by Mouse
- **Policy**: Logic by which Mouse choose what to do.
  - For example, the mouse may find an action which gives a small amount of reward and instead of exploiting this action it will choose exploration in order to find a better solution.
- **Transition Model**: Maze with all possible transitions.
- **Trajectories**: Sequence of Mouse's actions

Imitation Learning (IL)

- Introduction to IL
- Elements of IL
- **Behavioral Cloning**
- Direct Policy Learning
- Inverse Reinforcement Learning
- Challenges of IL
- IL Project in Unity
- IL in Autonomous Driving
- Cinematography Shooting

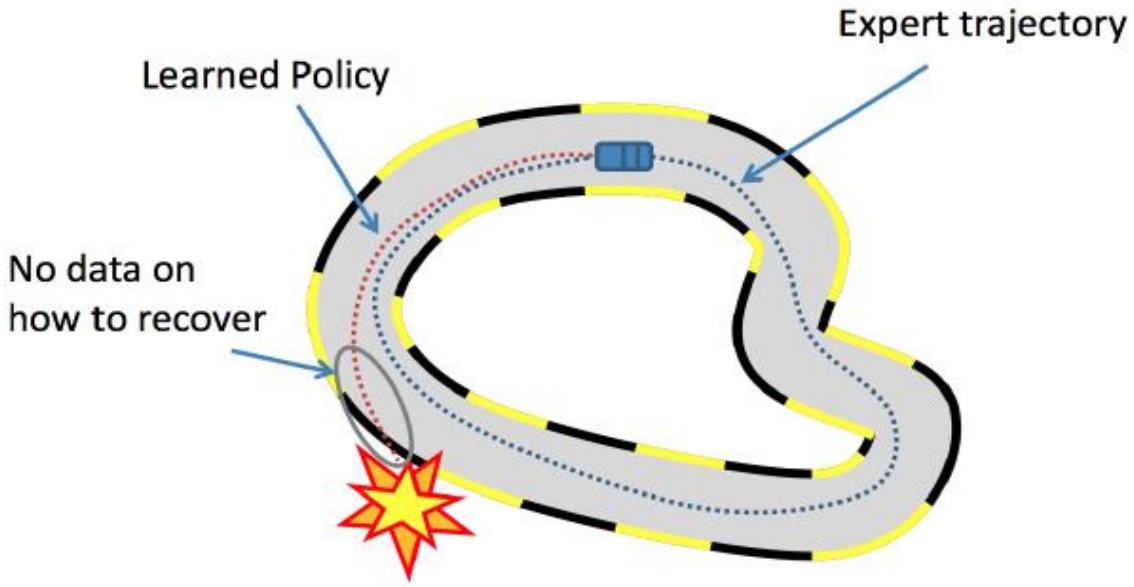
Behavioral Cloning (BC)

- The simplest form of imitation learning is Behavioral Cloning, which focuses on learning the expert's policy by using supervised learning.
- In Behavioral Cloning the agent learns to directly map states to actions without recovering the reward function.

- Algorithm's outline:
  - Record set of demonstrations ($\boldsymbol{\tau}^*$ trajectories) from expert.
  - Select a policy representation $\pi_{\boldsymbol{\theta}}$.
  - Choose an objective function J that shows the comparison between the demonstrated behaviors and the learner's policy and optimize it.
  - Return optimized policy parameter vectors $\boldsymbol{\theta}$.

Problems

- In some applications BC can work excellently, in most cases, though, it can be problematic.
- Errors add up, a mistake made by the agent can put it into a state that the expert has never visited. In such states, the behaviour is undefined and this can lead to catastrophic failures.

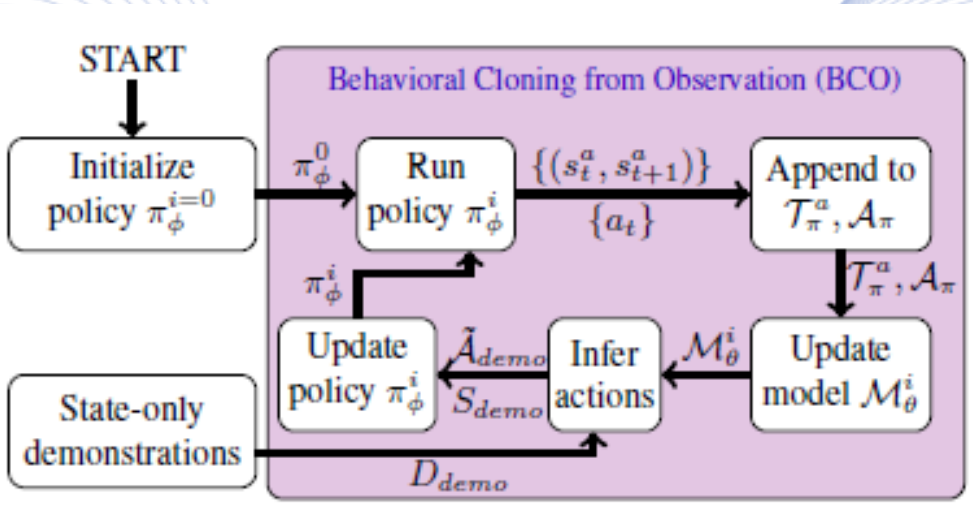


Reference: [MED-IL]

Behavioral Cloning from Observation (BCO)

- BCO combines inverse dynamics model learning with learning an imitation policy.
  - The agent learns a task-independent, inverse dynamics model, without any demonstration. Then, provided demonstrated information about the state, the model is trying to find action information that the expert did not provide.

  - BCO tries to discover a policy through behavioral cloning using the demonstration and the inferred actions.



Reference:[TOR2018]

Imitation Learning (IL)

- Introduction to IL
- Elements of IL
- Behavioral Cloning
- **Direct Policy Learning**
- Inverse Reinforcement Learning
- Challenges of IL
- IL Project in Unity
- IL in Autonomous Driving
- Cinematography Shooting

Direct Policy Learning (DPL)

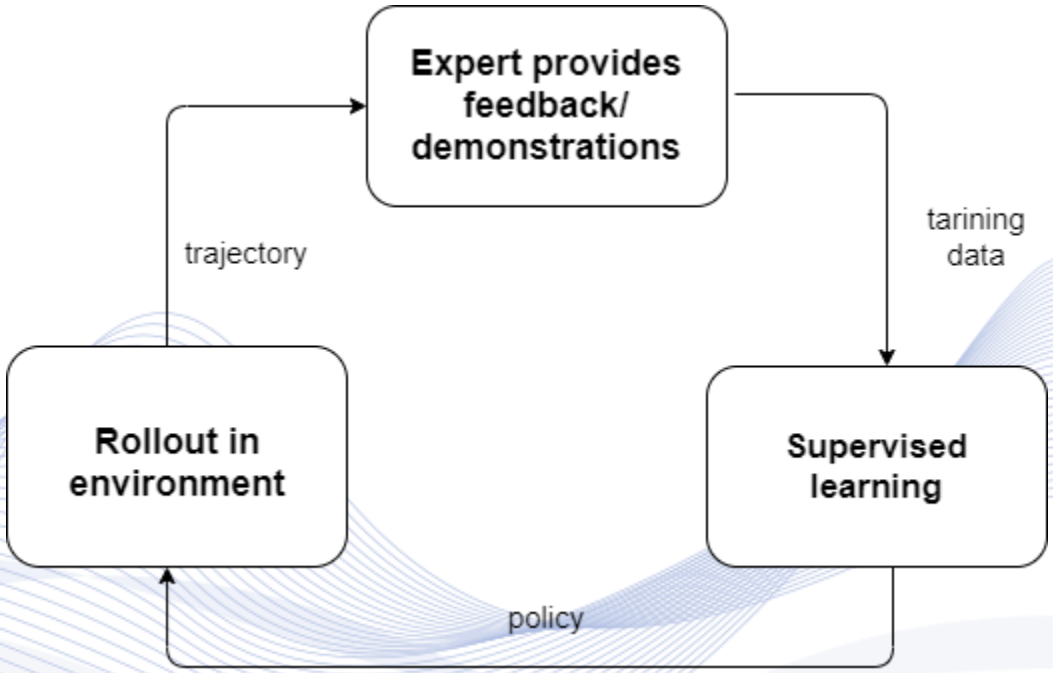
- DPL is an improved version of behavioral cloning.
- The agent collects demonstrations from the experts and applies supervised learning to learn a policy.

- It is an iterative method, an expert evaluates the agent's actions at all times.

- The agent obtains extra training data as a feedback to supervised learning.

General Algorithm

- Construct loss function $J(\pi^*(s), \pi(s))$, difference between what policies tell agent to do and what expert would have done in this state.
- Training the model:
  - Initial predictor π_0 – initial expert's demonstrations.
  - Collect trajectories $\boldsymbol{\tau}$ via rolling out π_{n-1} (Rollout policies one after another. We do this as many times as possible to collect a large enough data set).
  - Estimate state distribution P_n using $s \in \boldsymbol{\tau}$.
  - Collect interactive feedback from an interactive expert $\{(\pi^*(s)|s \in \boldsymbol{\tau}\}$.



Problem

- The algorithm works more efficiently when it uses all the previous training data during the trainings (the agent “remembers” all the mistakes). This can be achieved with the following methods:

  - **Data Aggregation:** Gathers a dataset in every iteration according to the last policy and trains the following policy under the aggregate of all accumulated datasets.

  - **Policy Aggregation:** The policy is trained based on the data of the last iteration and subsequently the policy is combined with all the preceding policies using geometric blending.

Policy Aggregation (SEARN & SMIle)

- At each iteration SEARN learns a new policy $\hat{\pi}_n$ and returns a distribution over previously learned policies:

$$\pi_n = \beta\hat{\pi}_n + (1-\beta)\pi_n, \beta \in (0,1).$$

- SMIle takes the advantages of SEARN and has less interaction with the expert and a simpler implementation:

$$\pi_n = (1-\beta)^N \pi^* + \beta \sum_{n'} (1-\beta)^{N-n'} \hat{\pi}^{*n}.$$

Imitation Learning (IL)

- Introduction to IL
- Elements of IL
- Behavioral Cloning
- Direct Policy Learning
- **Inverse Reinforcement Learning**
- Challenges of IL
- IL Project in Unity
- IL in Autonomous Driving
- Cinematography Shooting

Inverse reinforcement learning (IRL)

- In Inverse Reinforcement Learning the main idea is to learn the reward function of the environment based on the expert's demonstrations.

- Then find the optimal policy (the one that maximizes the reward function) by using reinforcement learning.

General Algorithm

- A set of expert's demonstrations is given: $\mathcal{D} = \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_n\}$ (we assume these are optimal).
- We update the reward function parameter by optimizing the objective function, in order to find more optimal demonstrations.
- The policy parameter vectors $\boldsymbol{\theta}$ are updated using reinforcement learning methods.
- If we run this operation again and again we can obtain the policy and reward function parameters.
- Finally, the newly learned policy π is compared with the expert's policy π^*.

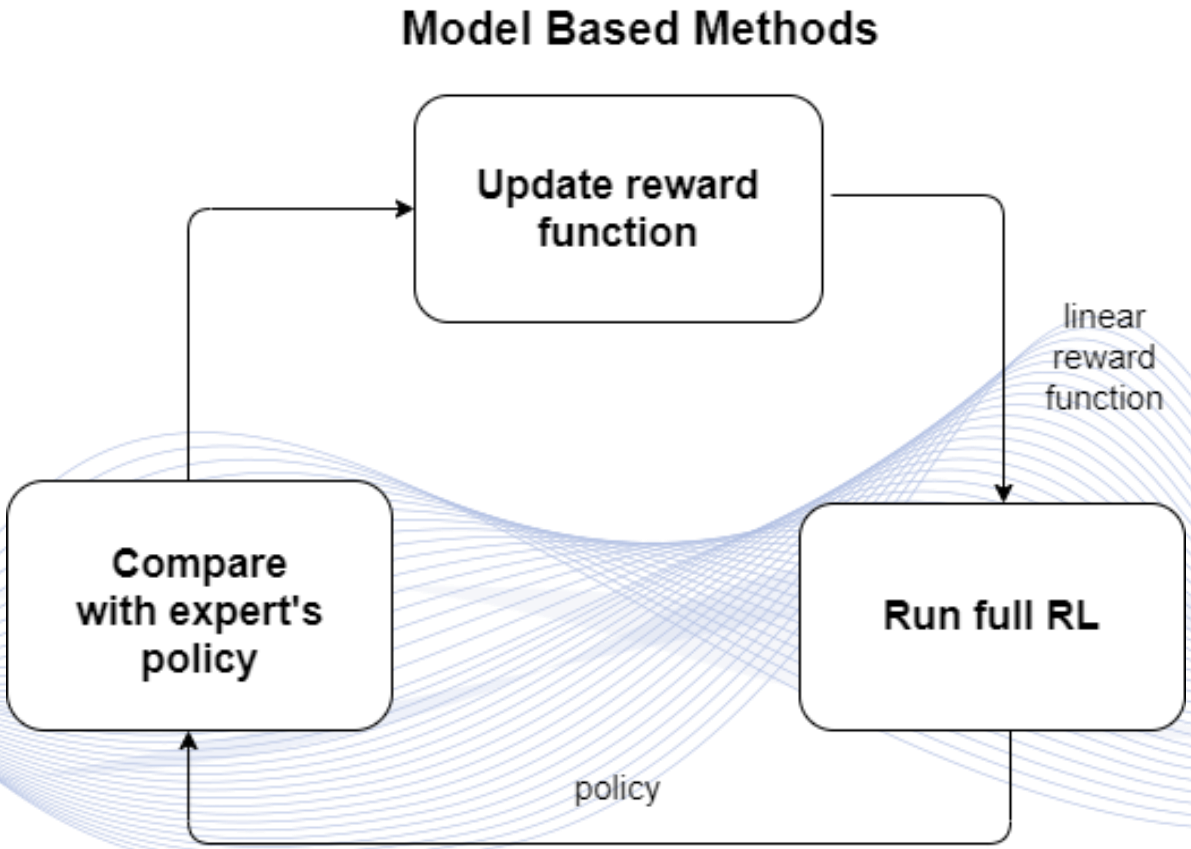
Inverse reinforcement learning (IRL)

- IRL can be distinguished into two categories:

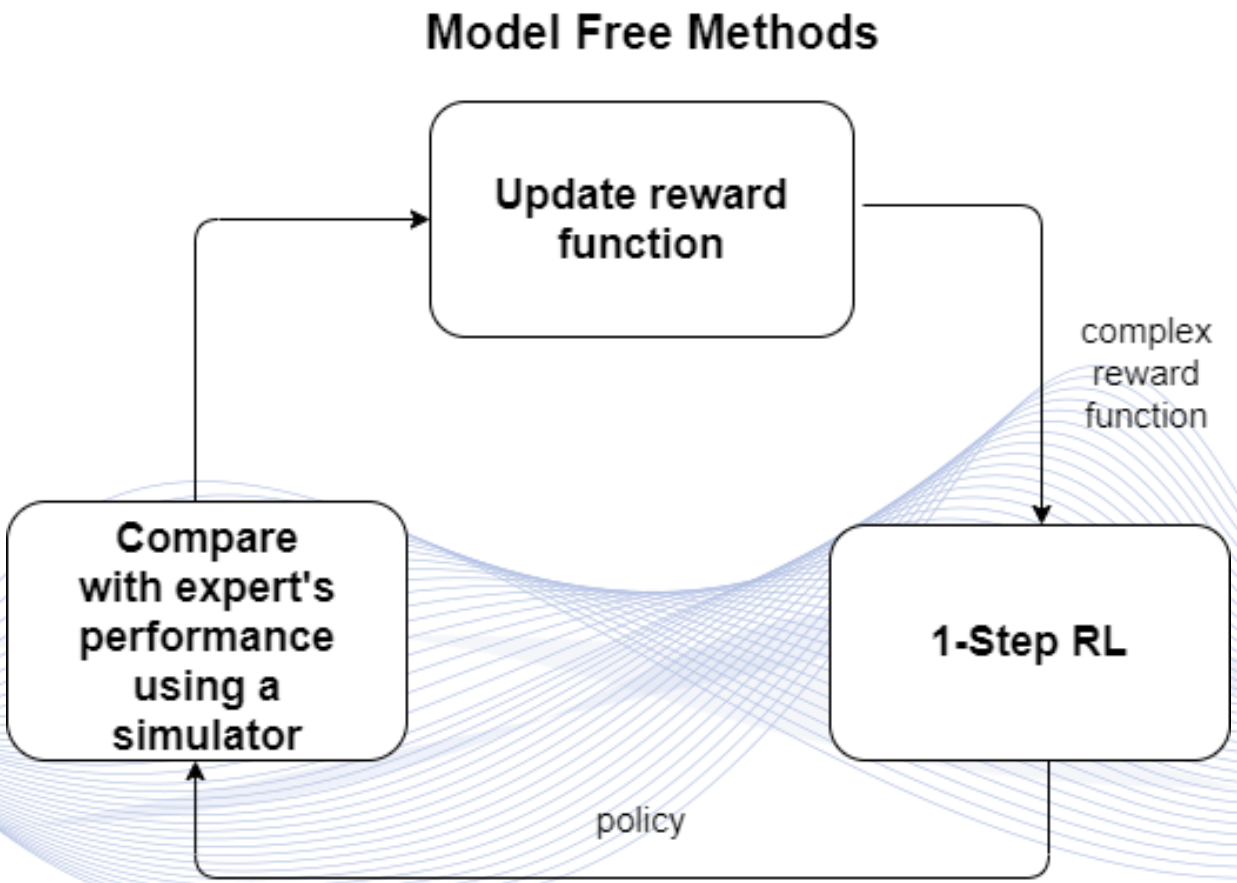
- **Model-based methods:** Require the knowledge of system dynamics in order to assess how often a state-action is visited.

- **Model-free methods**: Usually engage sampling-based methods to achieve this goal.

Model Based Algorithm

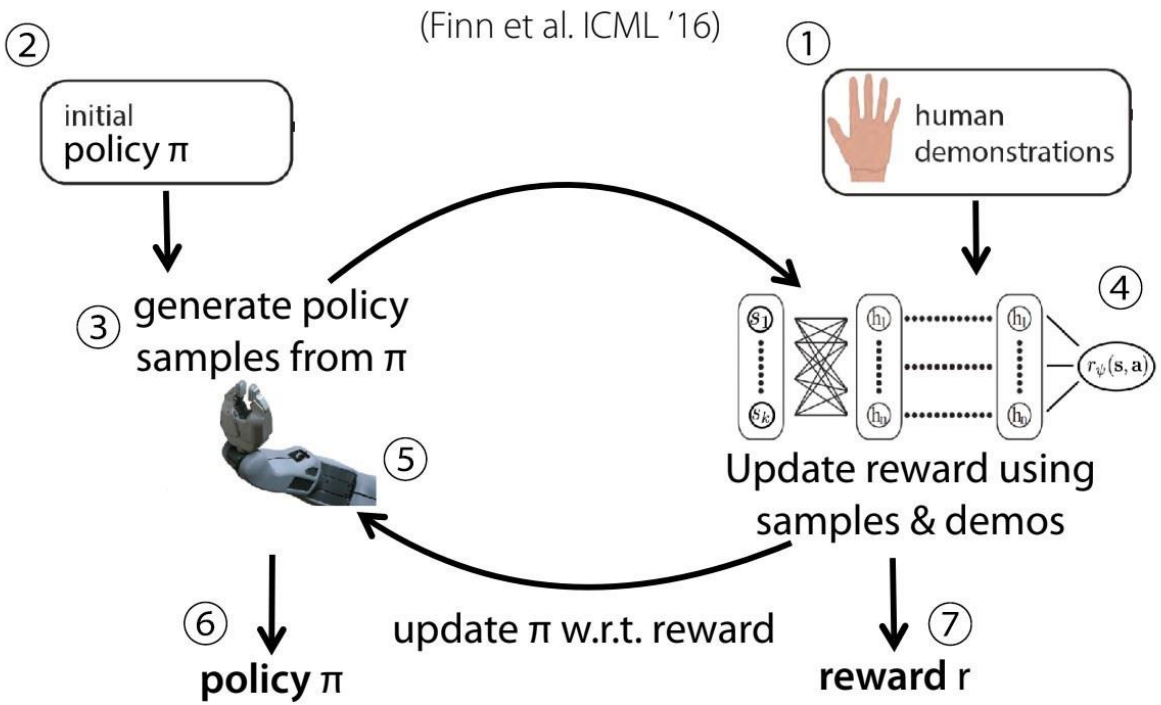


Model Free Algorithm



Robot Example

- 1. Collect a set of human demonstrations.
- 2. Initialize a controller $\pi(a|s)$ (a.k.a policy).
- 3. Run this controller on the robot and observe the sampled trajectories.
- 4. Using these samples and the human demonstrations, we calculate the reward gradient to update the reward model.
- 5. Similarly, using the Policy Gradient method, we refine the policy model using the updated reward function.
- 6. As the policy is updated, it gets better in sampling trajectories with higher rewards.
- 7. A better policy gives a better estimate for (4) and therefore the reward function is getting better also.



Reference: [MED-IRL]

Apprenticeship learning

- Apprenticeship learning is a variant of reinforcement learning.

- A learning agent (apprentice) observes another agent (expert), acting in a Markov Decision Process (MDP).

- The apprentice tries to achieve to learn a policy π^A that is similar to the expert's π^E, respective to an unknown reward function:

$$V(\pi^A) \geq V(\pi^E).$$

Imitation Learning (IL)

- Introduction to IL
- Elements of IL
- Behavioral Cloning
- Direct Policy Learning
- Inverse Reinforcement Learning
- **Challenges of IL**
- IL Project in Unity
- IL in Autonomous Driving
- Cinematography Shooting

Challenges of IL

- Difficulty of accurate inference:
  - IRL suffers from ambiguity in solution. There are many reward function that can create policies that understand the observed demonstration.

- Generalizability:
  - Generalizing correctly to states and actions that have not been observed in the demonstration and beginning the task at different initial states using the learned information is challenging.

Challenges of IL

- Sensitivity to Correctness of Prior Knowledge:
  - The accuracy of IRL is closely related to correct features. The sensitivity is affected by the selection of feature functions that enclose facets of the expert's actual reward function and transition function in MDP.
- Disproportionate Growth in Solution Complexity with Problem Size:
  - Computational complexity includes the time complexity in every iteration and its space complexity. Suffer from the curse of dimensionality and sample complexity.

Challenges of IL

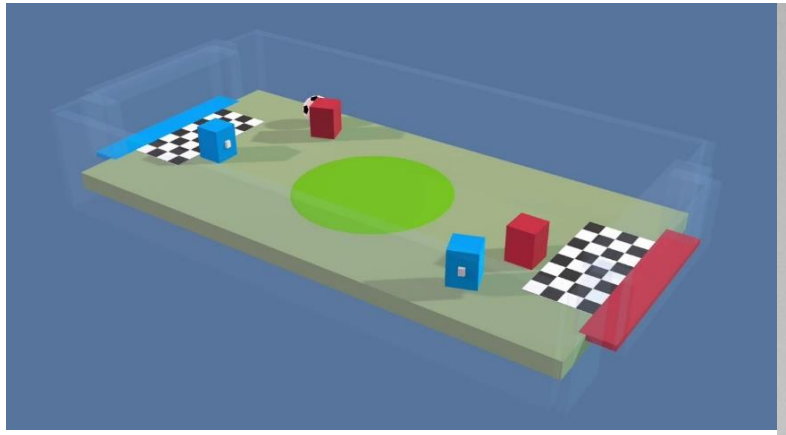
- Direct Learning of Reward or Policy Matching:

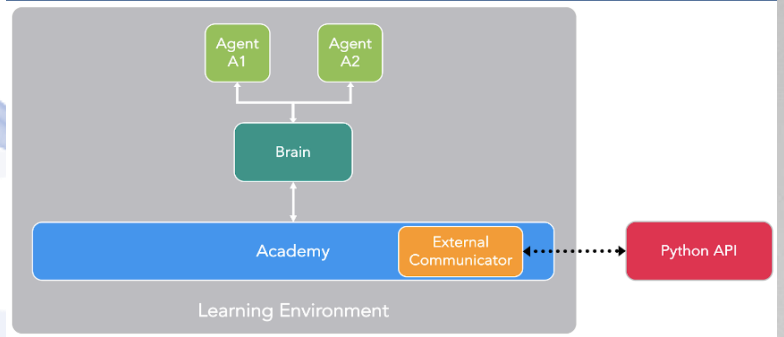
  - Direct learning is trying to immediately estimate the reward function. It may learn policies that do not completely replicate the behaviors it has observed.
  - Policy matching is learning a policy that can map the actions with the observed trajectories. Missing states in demonstration can result in not being able to match the observed policies.

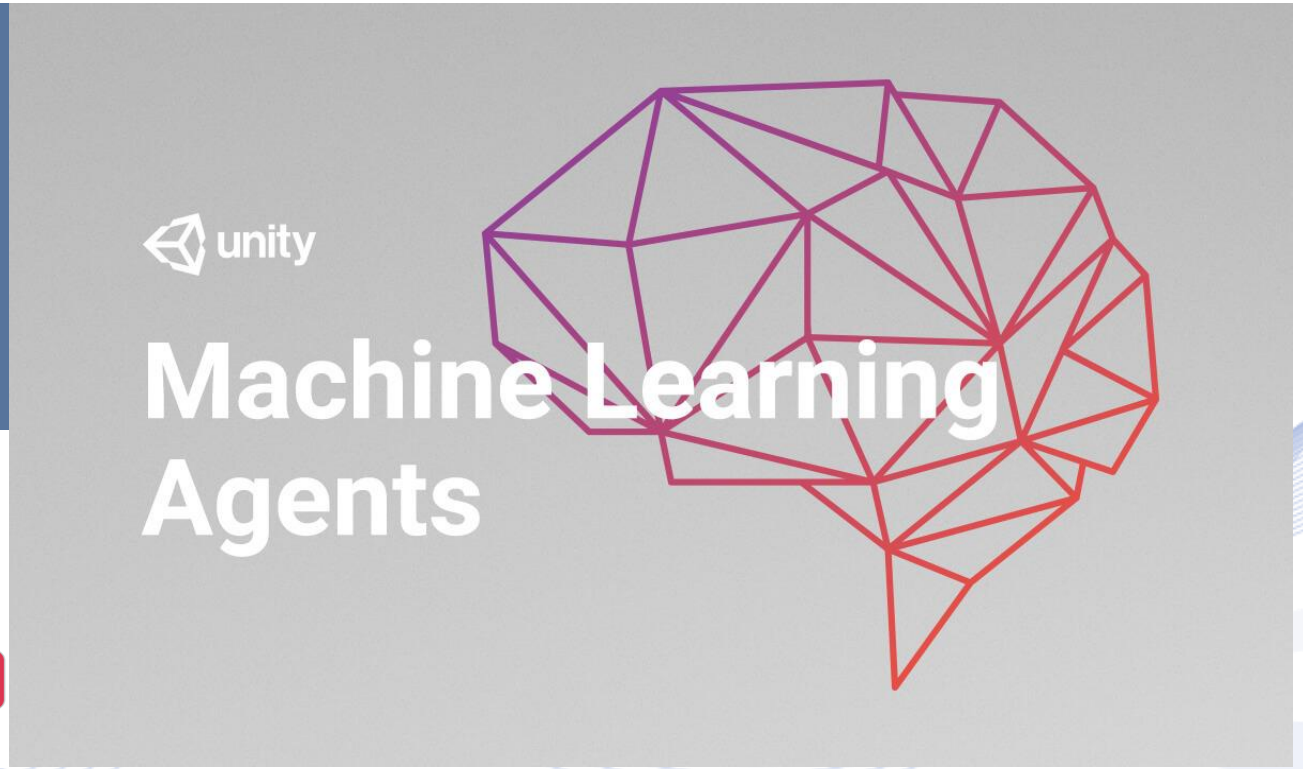
Imitation Learning (IL)

- Introduction to IL
- Elements of IL
- Behavioral Cloning
- Direct Policy Learning
- Inverse Reinforcement Learning
- Challenges of IL
- **IL Project in Unity**
- IL in Autonomous Driving
- Cinematography Shooting

Imitation Learning in Game Development

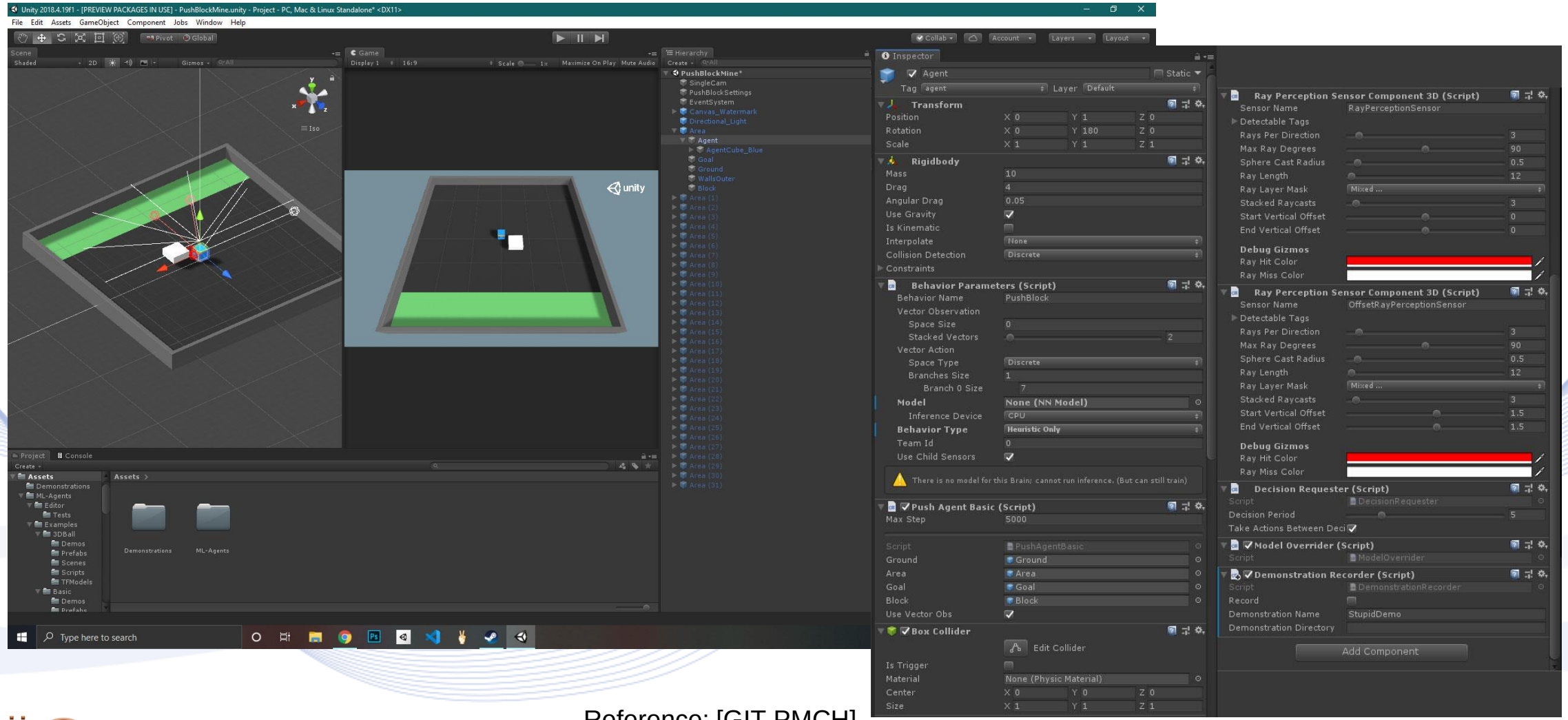






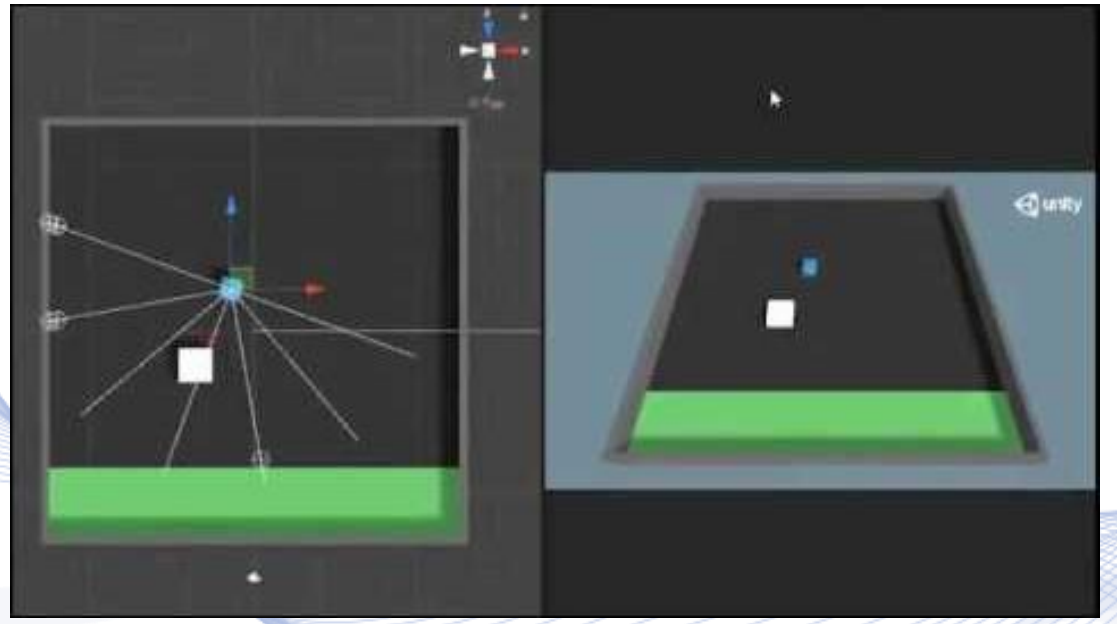
Reference: [GIT-PMCH]

Environment Setup

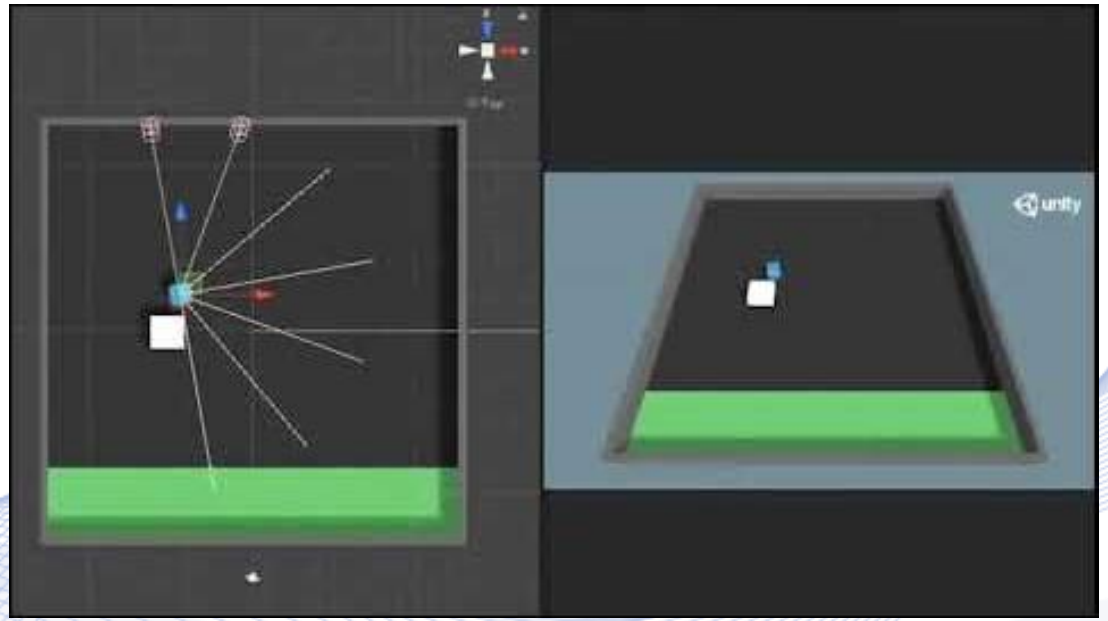


Reference: [GIT-PMCH]

Demonstrations

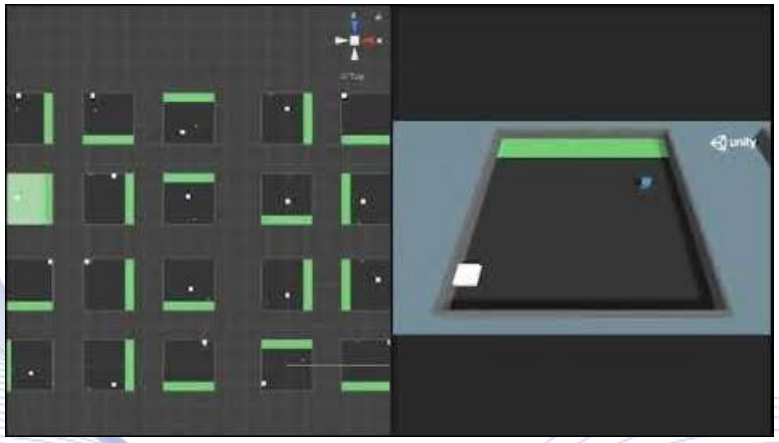


“Smart” Demonstration

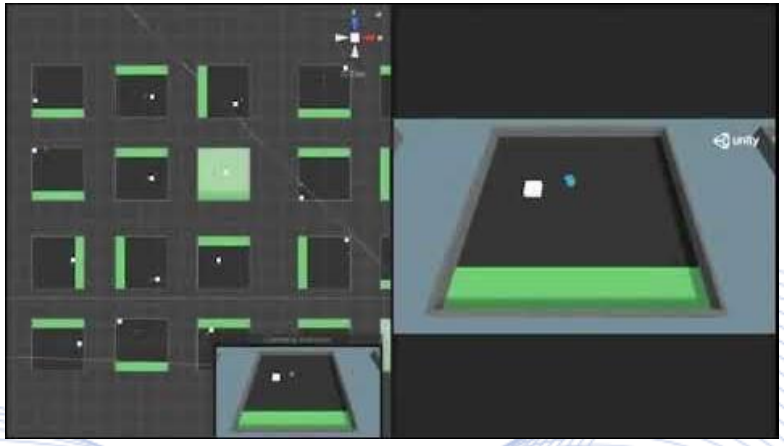


“Bad” Demonstration

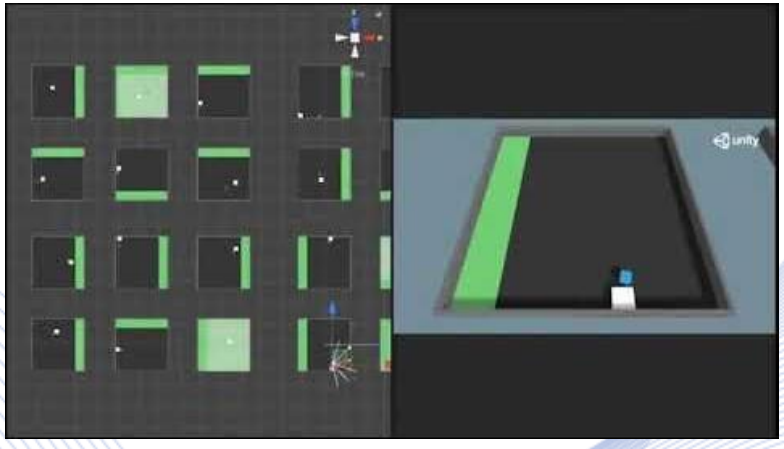
Training Process



Training of Agent with "Bad" Demonstrations

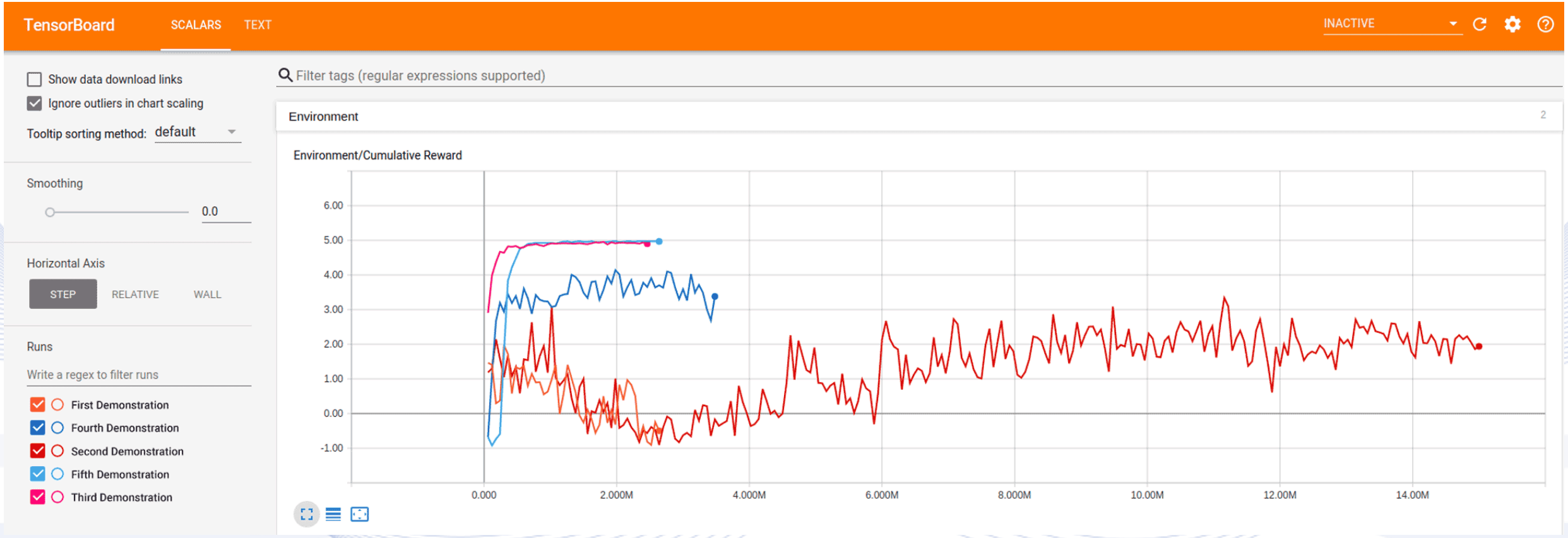


Training of Agent with "Perfect" Demonstrations



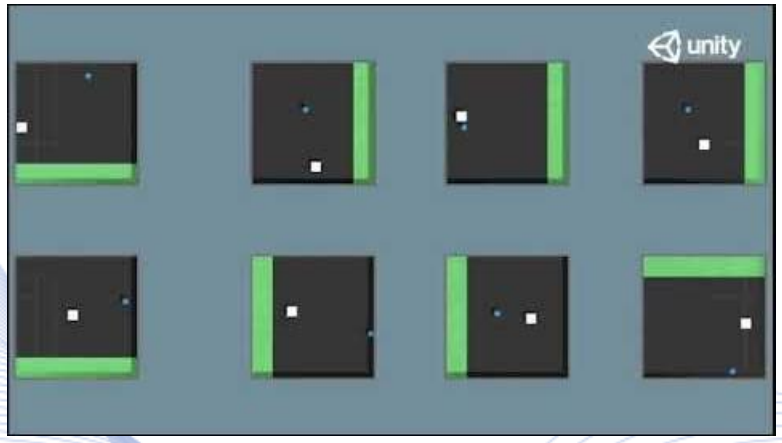
Training of Agent with simple Reinforcement Learning

Results

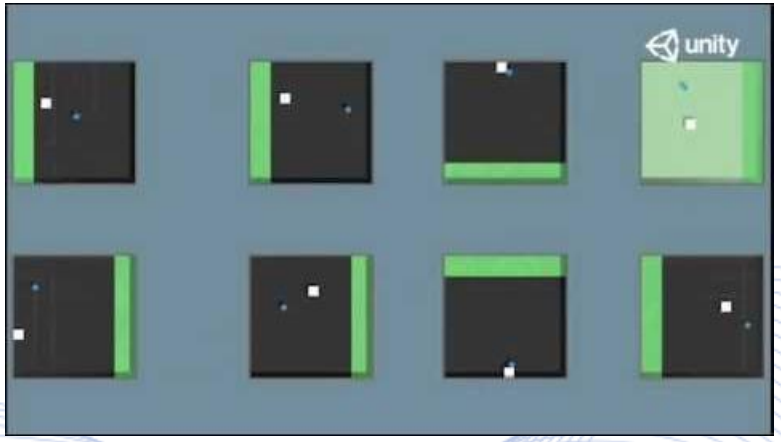


Reference: [GIT-PMCH]

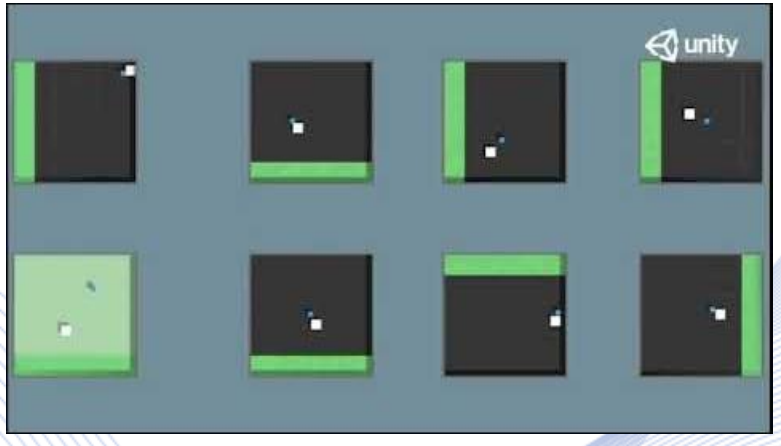
Agent's Performance



Gameplay of Agent that was trained on "Bad" Demonstrations



Gameplay of Agent that was trained on "Perfect" Demonstrations



Gameplay of Agent that was trained with simple Reinforcement Learning

Imitation Learning (IL)

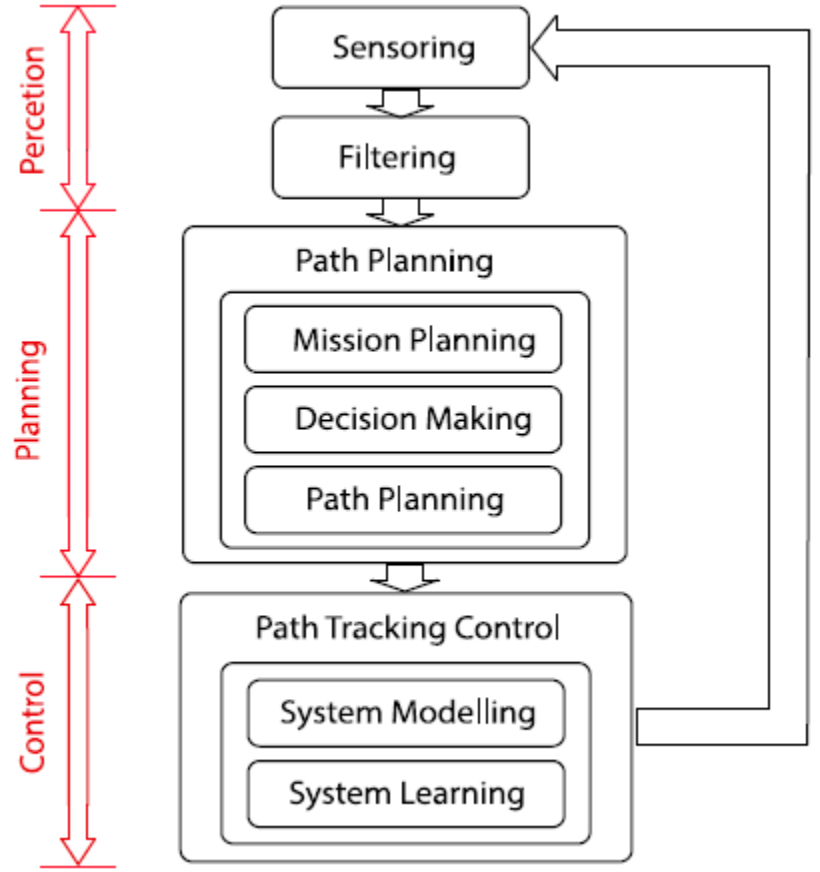
- Introduction to IL
- Elements of IL
- Behavioral Cloning
- Direct Policy Learning
- Inverse Reinforcement Learning
- Challenges of IL
- IL Project in Unity
- **IL in Autonomous Driving**
- Cinematography Shooting

IL in Autonomous Driving

- The exploration of end-to-end learning for autonomous-driving vehicles has started in the late 1980s.
- ALVINN was one of the first attempts, which learned to steer angles using camera and laser measurements in a neural network.
- Significant progress has been witnessed in autonomous driving ever since.

IL in Autonomous Driving

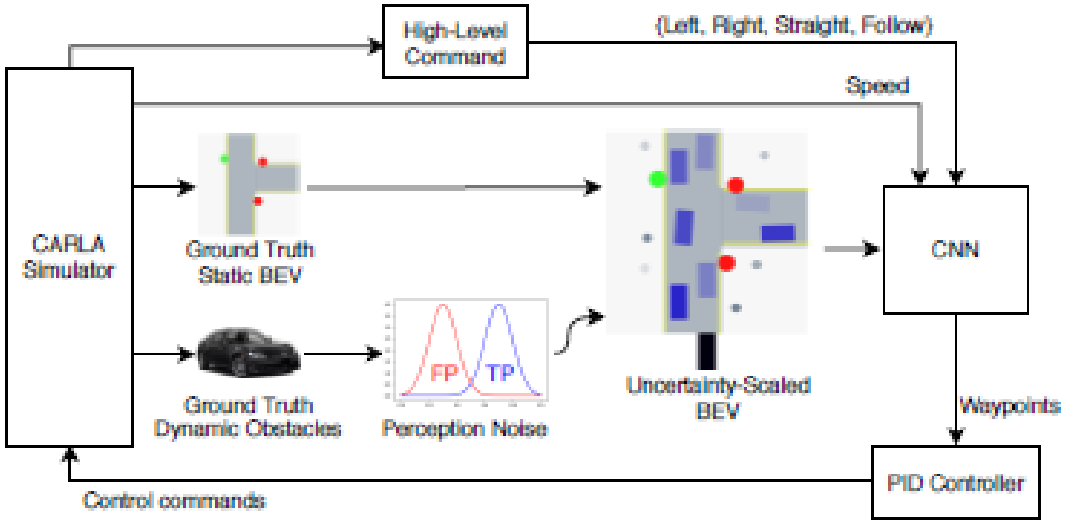
- A fully autonomous vehicle consists of three levels:
  - Perception:
    - Sensing system: Information about the state and environment of the vehicle are provided through the sensing system using sensors.
    - Filtering system: Then the filtering system must remove the noise from signals from the sensing system and supply estimations of unmeasurable states.



Reference: [YOU2019]

IL Framework for Autonomous Driving

- The way-points are then used by a PID controller to compute the control signals for the steering and throttle of the vehicle.
- The goal is to learn a policy that fulfills the standard behavioral cloning target, while remaining invariant to the noise in the input features.

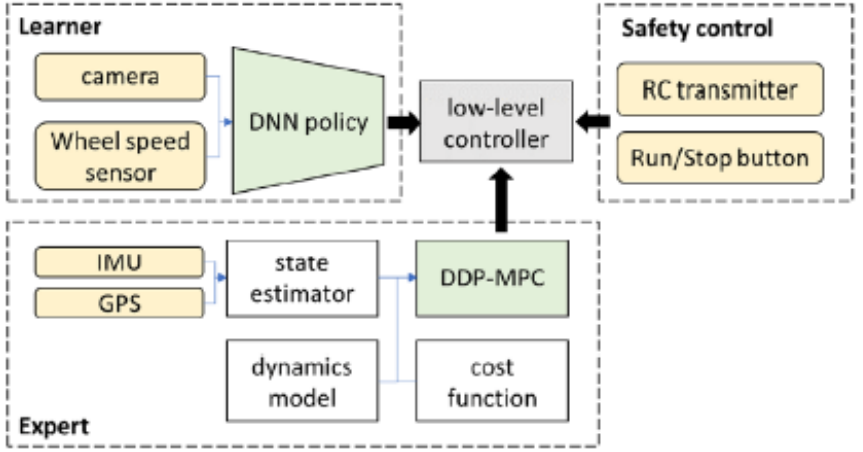


Reference: [BUR2020]

IL Framework for Autonomous Driving

- Imitation learning for agile autonomous driving:
  - The system learns DNN driving policies and is comprised of three high level controllers:
    - Expert
    - Learner
    - Safety control module

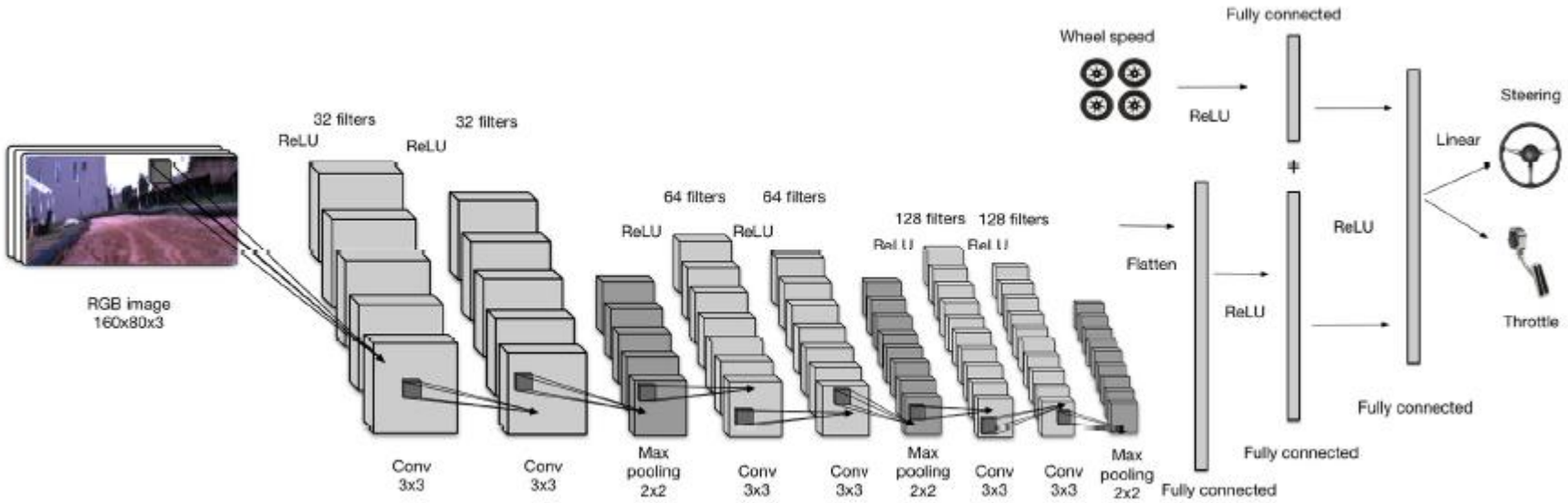
  and a low level controller for the steering and throttle commands.



Reference: [PAN2020]

IL Framework for Autonomous Driving

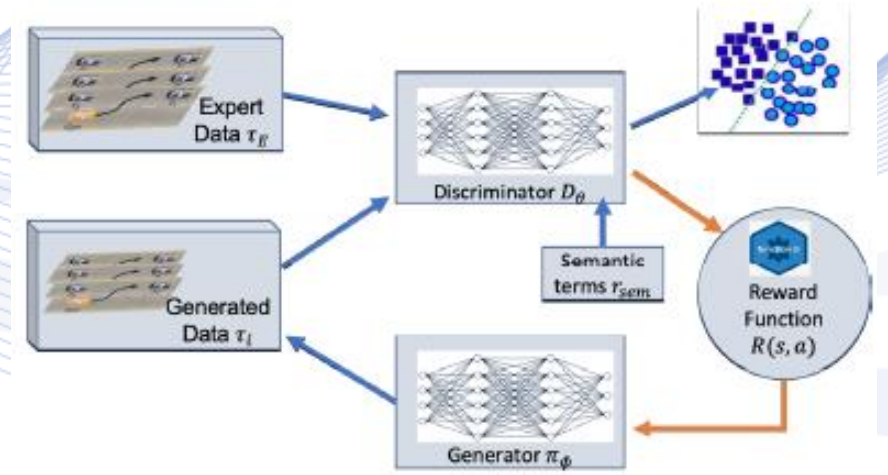
- The learner's control policy is parametrized by a DNN network.



Reference: [PAN2020]

IL Framework for Autonomous Driving

- Augmented AIRL learning framework:
  - The data provided from the expert and the generator are fed into discriminator.
  - The weights of the discriminator are comprised of the reward function for policy update.
  - In order to create new samples the last policy and reward function are used for this purpose.



Reference: [WAN2019]

Challenges

- Bootstrapping RL with imitation:
  - High quality demonstrations are hard to collect.
  - Approaches: DQfD, combination of RL and LfD (Learning from Demonstrations).

- Exploration issues with imitation:
  - There is a possibility where the agent can't receive the expert's demonstrations or these demonstrations do not cover the state space, resulting in learning bad policies.
  - Approaches: DAgger, SEARN, SMIle, Hierarchical imitation.

Imitation Learning(IL)

- Introduction to IL
- Elements of IL
- Behavioral Cloning
- Direct Policy Learning
- Inverse Reinforcement Learning
- Challenges of IL
- IL Project in Unity
- IL in Autonomous Driving
- **Cinematography Shooting**

Cinematography Shooting

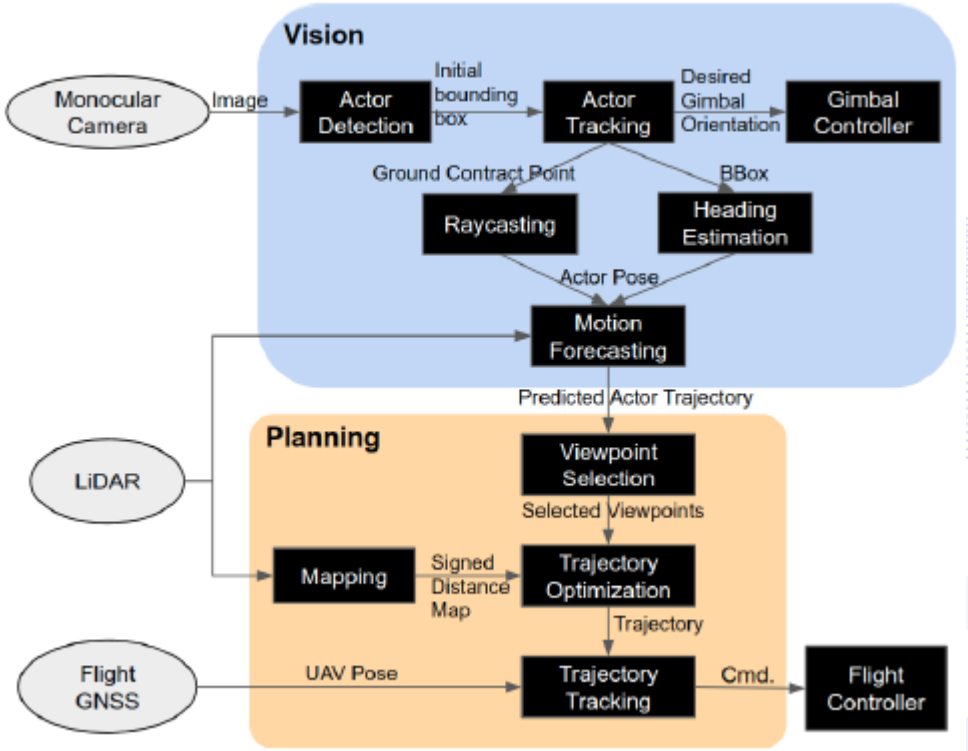
- Another application of imitation learning that gains attention is cinematography shooting.

- Autonomous cinematography systems, like Unmanned Aerial Vehicles (UAVs, or “drones”), try to mimic a cameraman’s operations to learn video shooting skills.

- The use of drones in cinematography shooting can be demanding as it is necessary to simultaneously operate as a controller and analyze dynamic scenarios.

Cinematography Shooting Frameworks

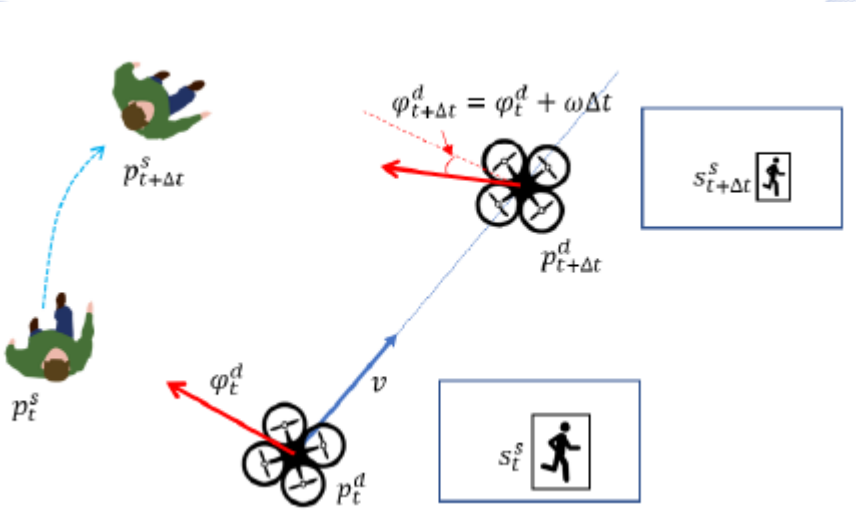
- Autonomous Cinematography using Unmanned Aerial Vehicles:
  - System architecture:
    - The vision subsystem controls the camera orientation using image-space feedback and forecasts the actor's future poses in the world frame.
    - The planning subsystem uses the forecasted actor motion, current UAV pose, and truncated signed distance map to generate flight trajectories (using IRL methods).



Reference: [ZHA2018]

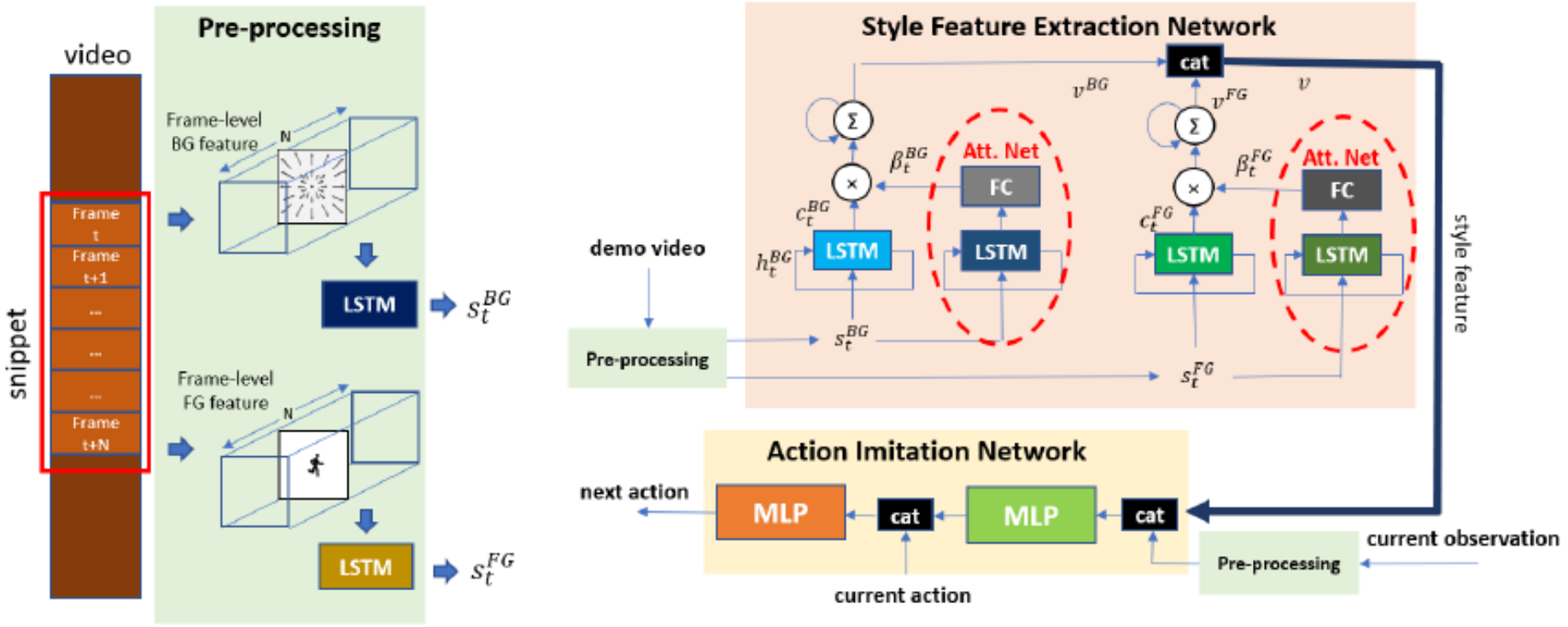
Cinematography Shooting Frameworks

- One-Shot Imitation Filming of Human Motion Videos:
  - Pre-processing: a video is constituted of a series of fragments of foreground (FG) and background (BG).
  - The style feature extraction network maps a variable-length demo video into a fixed-length style feature.
  - The action imitation network forecasts the next camera motion from the current observation, current action and style feature.



Reference: [HUA2019]

Cinematography Shooting Frameworks



One-shot imitation filming framework.

Reference: [HUA2019]

Current Challenges

- Legal Safety restrictions:
  - Maximum flight altitude.
  - Minimum distance from human crowd.
  - Constant sight between pilot and vehicle.
- Energy consumption restrictions:
  - Limit UAV continuous flight.
- Finite bandwidth in the wireless communication channel.

Bibliography

[POM1989] ALVINN, Dean A. Pomerleau et al., 1989

[TAY2017] A Deep Learning Approach for Generalized Speech Animation, Sarah Taylor, Taehwan Kim and Yisong Yue et al., 2017

[ROS2011] A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning, Stéphane Ross et al., 2011

[ROS2010] Efficient Reductions for Imitation Learning, Stéphane Ross and J. Andrew Bagnell, 2010

[ABB2004] Apprenticeship Learning via Inverse Reinforcement Learning, Pieter Abbeel and Andrew Y. Ng, 2004

[SYE2007] A Game-Theoretic Approach to Apprenticeship Learning, Umar Syed and Robert E. Schapire, 2007

[ABB2017] Guided Cost Learning: Deep Inverse Optimal Control via Policy Optimization, Sergey Levine, Pieter Abbeel and Chelsea Finn et al., 2016

[ZIE2008] Maximum Entropy Inverse Reinforcement Learning, Brian D. Ziebart et al., 2008

Bibliography

[YOU2019] You, Changxi, et al. "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning." Robotics and Autonomous Systems 114 (2019): 1-18.

[TOR2018] Torabi, Faraz, Garrett Warnell, and Peter Stone. "Behavioral cloning from observation." arXiv preprint arXiv:1805.01954 (2018).

[BUR2020] Bühler, Andreas, et al. "Driving through ghosts: Behavioral cloning with false positives." arXiv preprint arXiv:2008.12969 (2020).

[PAN2020] Pan, Yunpeng, et al. "Imitation learning for agile autonomous driving." The International Journal of Robotics Research 39.2-3 (2020): 286-302.

[WAN2019] Wang, Pin, et al. "Human-like Decision Making for Autonomous Driving via Adversarial Inverse Reinforcement Learning." arXiv (2019): arXiv-1911.

[ZHA2018]Zhang, Yanfu, et al. "Autonomous Cinematography using Unmanned Aerial Vehicles.“2018

[HUA2019] Huang, Chong, et al. "One-Shot Imitation Filming of Human Motion Videos." arXiv preprint arXiv:1912.10609 (2019).

Bibliography

[MAD2019] Mademlis, Ioannis, et al. "Autonomous uav cinematography: a tutorial and a formalized shot-type taxonomy." ACM Computing Surveys (CSUR) 52.5 (2019): 1-33.

[OSA2018] Osa, Takayuki, et al. "An algorithmic perspective on imitation learning." arXiv preprint arXiv:1811.06711 (2018).

[ARO2018] Arora, Saurabh, and Prashant Doshi. "A survey of inverse reinforcement learning: Challenges, methods and progress." arXiv preprint arXiv:1806.06877 (2018).

[HO2016] Ho, Jonathan, and Stefano Ermon. "Generative adversarial imitation learning." Advances in neural information processing systems. 2016.

[DAU2009] Daumé, Hal, John Langford, and Daniel Marcu. "Search-based structured prediction." *Machine learning* 75.3 (2009): 297-325.

[ROS2010] Ross, Stéphane, and Drew Bagnell. "Efficient reductions for imitation learning." Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010.

Bibliography

[SHA2008] Shalev-Shwartz, Shai, and Sham M. Kakade. "Mind the duality gap: Logarithmic regret algorithms for online optimization." Advances in Neural Information Processing Systems 21 (2008): 1457-1464.

[WUL2015] Wulfmeier, Markus, Peter Ondruska, and Ingmar Posner. "Maximum entropy deep inverse reinforcement learning." arXiv preprint arXiv:1507.04888 (2015).

[LE2016] Le, Hoang M., et al. "Smooth imitation learning for online sequence prediction." arXiv preprint arXiv:1606.00968 (2016).

[HUS2017] Hussein, Ahmed, et al. "Imitation learning: A survey of learning methods." ACM Computing Surveys (CSUR) 50.2 (2017): 1-35.

References

[CAL-IL]https://drive.google.com/file/d/12QdNmMll-bGlSWnm8pmD_TawuRN7xagX/view

[MED-IL] https://medium.com/@SmartLabAI/a-brief-overview-of-imitation-learning-8a8a75c44a9c

[MED-IRL]https://jonathan-hui.medium.com/rl-inverse-reinforcement-learning-56c739acfb5a

[GIT-PMCH] https://github.com/pmchrist/ml-agents

Bibliography

[PIT2021] I. Pitas, “Computer vision”, Createspace/Amazon, in press.
[PIT2017] I. Pitas, “Digital video processing and analysis ” , China Machine Press, 2017 (in Chinese).
[PIT2013] I. Pitas, “Digital Video and Television ” , Createspace/Amazon, 2013.
[NIK2000] N. Nikolaidis and I. Pitas, 3D Image Processing Algorithms, J. Wiley, 2000.
[PIT2000] I. Pitas, “Digital Image Processing Algorithms and Applications”, J. Wiley, 2000.

Q & A

Thank you very much for your attention!

More material in
http://icarus.csd.auth.gr/cvml-web-lecture-series/

Contact: Prof. I. Pitas
pitas@csd.auth.gr