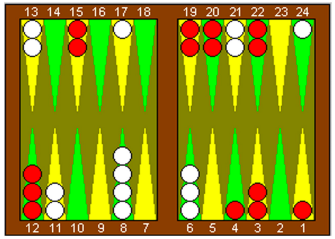
Learning from Demonstration
Applications and Challenges

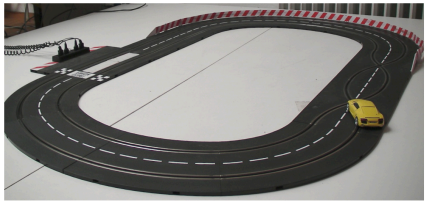
Feryal Behbahani

26 November 2018

Deep RL can learn everything?

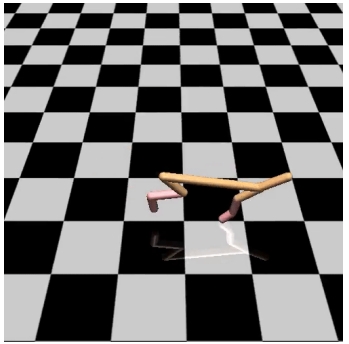


TD-Gammon, 1995

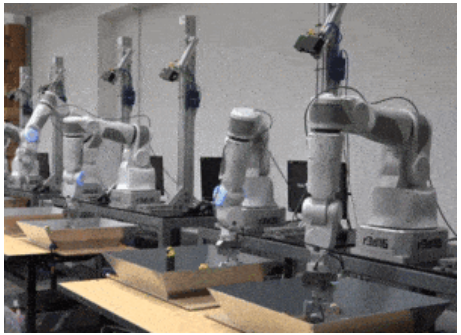


Slot car driving
Lang & Riedmiller 2012

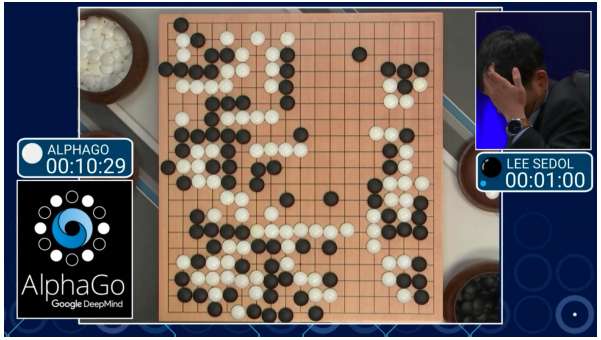
DQN, Mnih et al., 2013



TRPO, Schulman et al., 2015



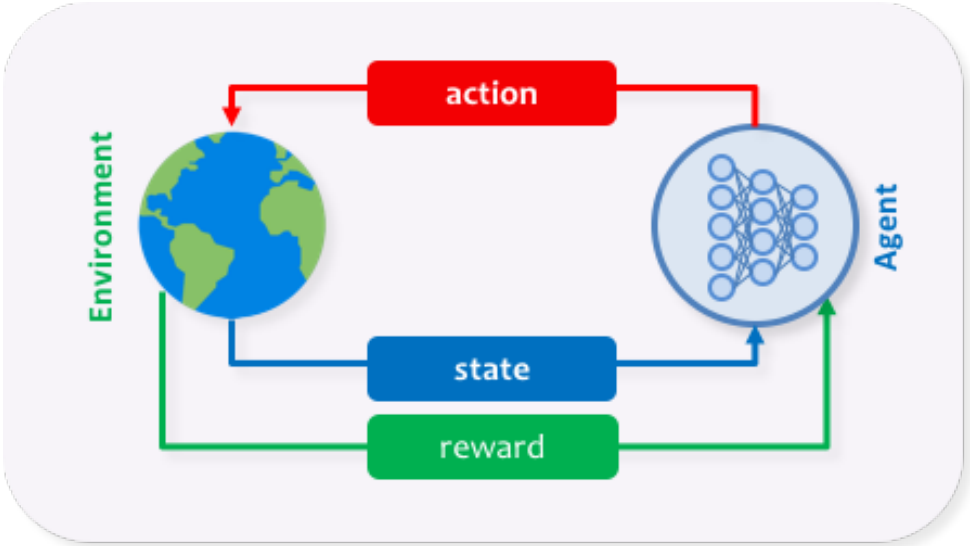
Levine et al., 2016



AlphaGo, Silver et al., 2016

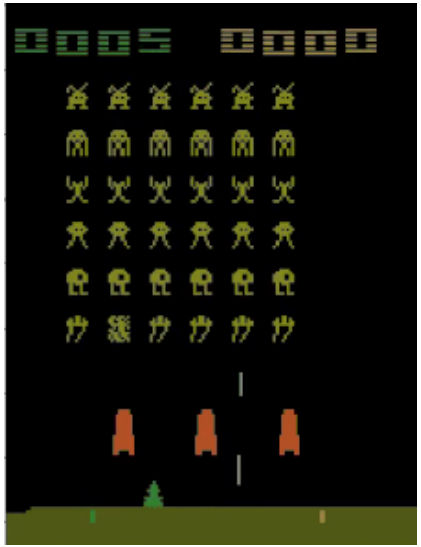
DOTA 2, OpenAI, 2018

Where do the rewards come from?



Reward

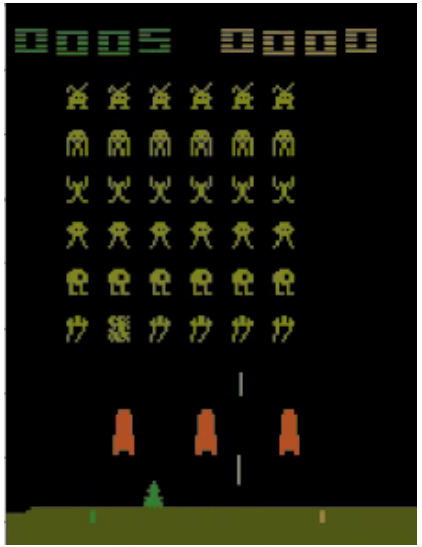
Games



Where do the rewards come from?

Reward

Games



Real World Problems!

For real world problems, there is no clear reward function, or it may vary.
It's usually easier to provide demonstrations to show what we mean!

Learning from demonstration

Many names:
Imitation Learning, Apprenticeship Learning, Programming by demonstration, …

Given: a dataset of demonstrations in the form of state-action pairs

$$\mathcal{D}_E := \left\{ \tau_i^E : \tau_i^E = \left((s_0^E, a_0^E), \ldots, (s_{T_i}^E, a_{T_i}^E) \right),\ i = 1 \ldots N \right\} \qquad \mathcal{D}_E \sim \pi_E$$

Goal: find a policy that mimics the demonstrations

$$\underset{\pi}{\operatorname{argmin}}\, \mathcal{L}(\pi, \pi_E)$$

Overview of LfD methods

Behavioural Cloning (BC)

Supervised learning of a mapping from expert states to expert's actions

ALVINN, Pomerleau, 1999
Learning to fly, Summut et al., 1992

Overview of LfD methods

Behavioural Cloning (BC)

Supervised learning of a mapping from expert states to expert's actions

ALVINN, Pomerleau, 1999
Learning to fly, Summut et al., 1992

Inverse RL (IRL)

Infers the reward function of the expert, given its behaviour

Feature matching, Abbeel and Ng, 2004
Maximum Margin IRL, Rattlif et al., 2007
Maximum Casual entropy IRL, Ziebart et al, 2008

Overview of LfD methods

Behavioural Cloning (BC)

Supervised learning of a mapping from expert states to expert's actions

ALVINN, Pomerleau, 1999
Learning to fly, Summut et al., 1992

Inverse RL (IRL)

Infers the reward function of the expert, given its behaviour

Feature matching, Abbeel and Ng, 2004
Maximum Margin IRL, Rattlif et al., 2007
Maximum Casual entropy IRL, Ziebart et al, 2008

RL + Demonstrations in memory (RLfD)

Embed the expert demonstrations into the replay memory and using off-policy learning and treat the expert behaviour as if it came from the agent.

DQNfD, Hester et al., 2017
DDPGfD, Večerík, 2017

Overview of LfD methods

Behavioural Cloning (BC)

Supervised learning of a mapping from expert states to expert's actions

ALVINN, Pomerleau, 1999
Learning to fly, Summut et al., 1992

Inverse RL (IRL)

Infers the reward function of the expert, given its behaviour

Feature matching, Abbeel and Ng, 2004
Maximum Margin IRL, Rattlif et al., 2007
Maximum Casual entropy IRL, Ziebart et al, 2008

RL + Demonstrations in memory (RLfD)

Embed the expert demonstrations into the replay memory and using off-policy learning and treat the expert behaviour as if it came from the agent.

DQNfD, Hester et al., 2017
DDPGfD, Večerík, 2017

Generative Adversarial Imitation Learning (GAIL)

Learns a policy directly by using a discriminator as a reward function similar to a GAN setup.

GAIL, Ho and Ermon, 2016
infoGAIL, Li et al., 2017
MGAIL, Baram et al., 2017

BC in a nutshell

Formulate problem as a standard supervised learning problem:
Predict expert action given expert state input.

Directly estimate the policy from expert training examples available.

$$\underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{\pi_\theta, \pi_E} \left[\| a - a_E \|^2 \right]$$

BC in a nutshell

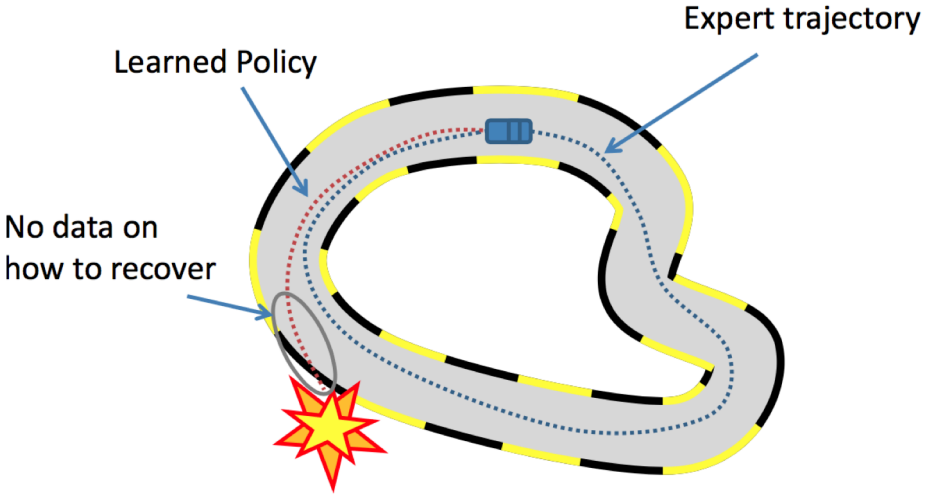
Formulate problem as a standard supervised learning problem:
Predict expert action given expert state input.

Directly estimate the policy from expert training examples available.

$$\underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{\pi_\theta, \pi_E} \left[\|a - a_E\|^2 \right]$$

Extensions:

- Data aggregation (Dagger):
  use online supervision from expert on novel states



Data distribution mismatch, also known as covariate shift

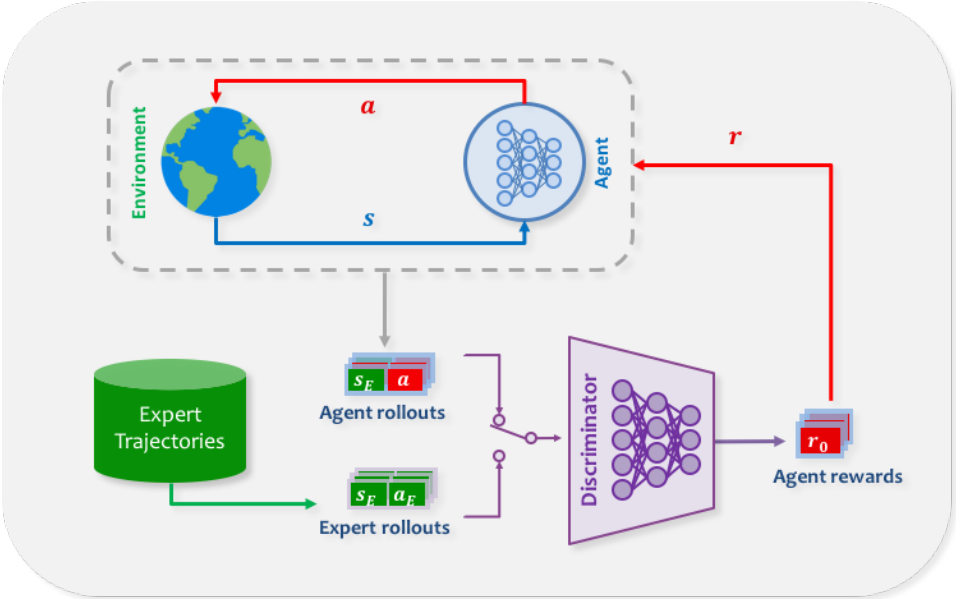
GAIL in a nutshell

Learn a deep neural network **policy** π_ϑ that cannot be ***distinguished*** from the **expert policy** π_E by the **discriminator** D_φ

Discriminator D_φ outputs probability that state-action pair is *fake / not from expert.*

<u>Adversarial game:</u>

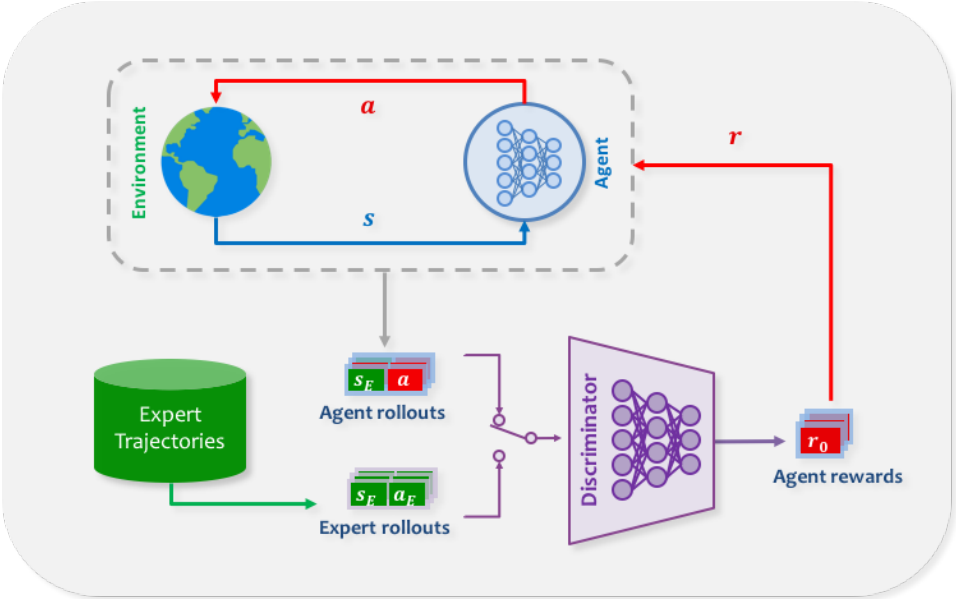
- Discriminator wants to classify between agent/expert accurately
- Agent policy wants to fool discriminator (i.e. minimise being classified as fake)



GAIL in a nutshell

Learn a deep neural network **policy** π_ϑ that cannot be ***distinguished*** from the **expert policy** π_E by the **discriminator** D_φ

Discriminator D_φ outputs probability that state-action pair is *fake / not from expert.*



Adversarial game:

- Discriminator wants to classify between agent/expert accurately
- Agent policy wants to fool discriminator (i.e. minimize being classified as fake)

Implemented as follows:

$$\min_\theta \max_\phi \mathbb{E}_{\pi_\theta}\left[\log(\underbrace{D_\phi(s,a)}_{\text{Discriminator output for agent}})\right] + \mathbb{E}_{\pi_E}\left[\log(1-\underbrace{D_\phi(s^E,a^E)}_{\text{Discriminator output for expert}})\right]$$

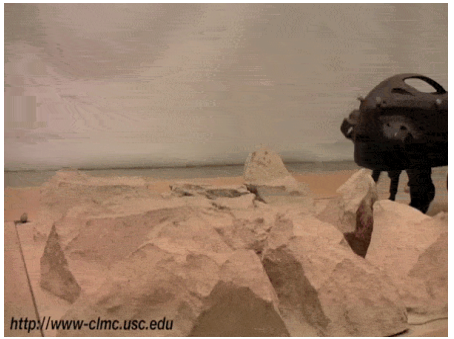
Agent can be trained using any RL algorithm, using:

$$r(s,a) = -\log(D_\phi(s,a))$$

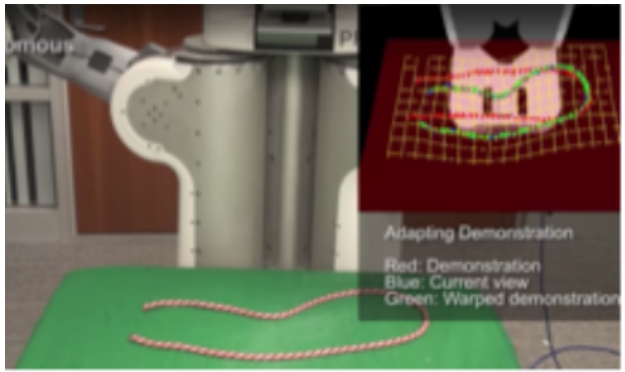
Success stories

Pomerleau et al., 1999

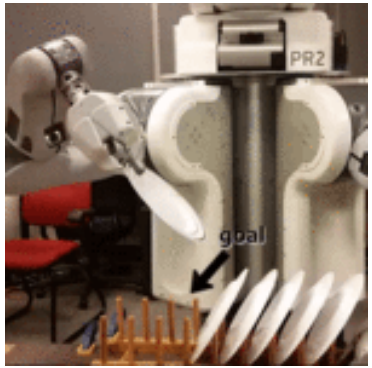
Abbeel et al., 2008



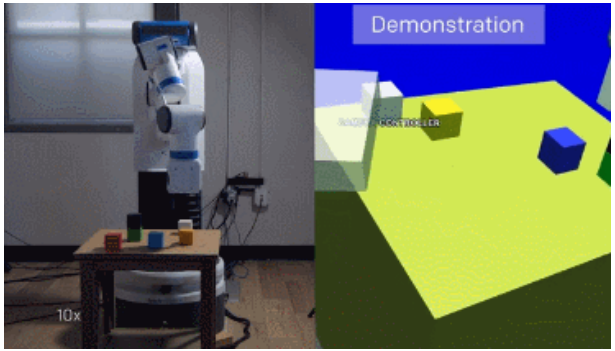
Kolter et al., 2008



Schulman et al., 2015



Finn et al., 2016



Duan et al., 2017

What if you don’t have access to demonstrator/demonstrations?!

There is a plethora of behaviour available in the wild!

Video to Behaviour (ViBe)

Use the wealth of human data around us to capture realistic human behaviour.

Multi-agent Traffic simulation

Learn **road user** policies, using available videos from **traffic cameras.**

Physical road tests are expensive and dangerous, simulation is an essential part of the training process BUT requires realistic simulator with **realistic models of road users…**

Learning from Demonstration in the Wild

Feryal Behbahani[1], Kyriacos Shiarlis[1], Xi Chen[1], Vitaly Kurin[1,2], Sudhanshu Kasewa[1,2], Ciprian Stirbu[1,2], João Gomes[1], Supratik Paul[1,2], Frans A. Oliehoek[1,3], João Messias[1], Shimon Whiteson[1,2]

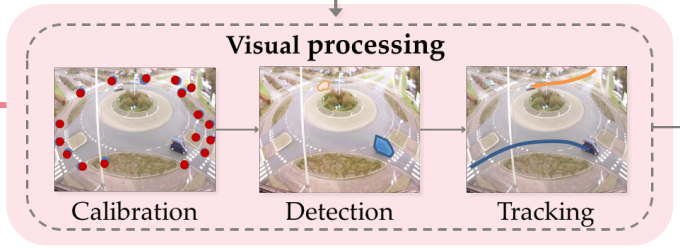
Pre-print available on arXiv

Raw videos of behaviour

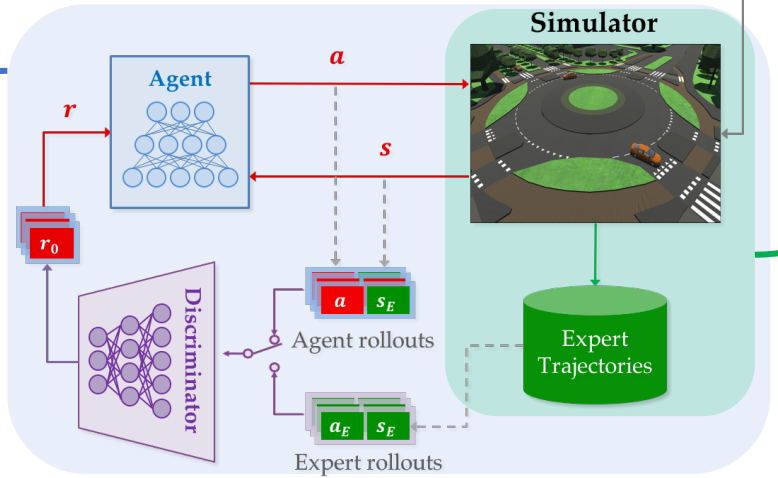
Single, monocular, uncalibrated camera
with ordinary resolution.

ViBe pipeline

First extracts demonstrations from videos



Build a simulator of the scene



Learn behaviour models using LfD

Extracting trajectories

Visual processing

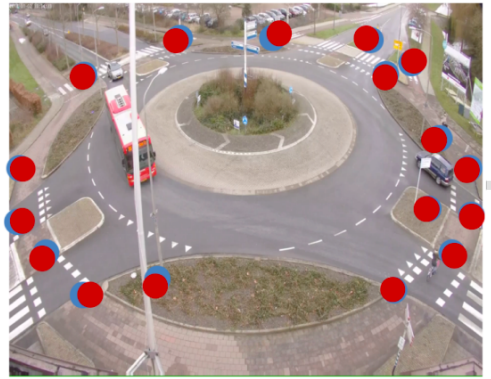
Calibration

Detection

Tracking

Extracting trajectories

Visual processing



Calibration

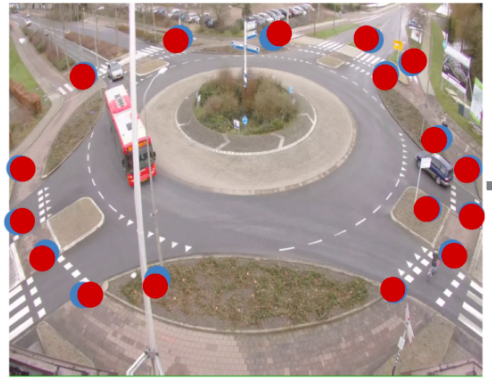
Detection

Tracking

We identify landmarks in both camera and Google Maps satellite images, and use them to estimate camera matrix and distortion parameters.

Extracting trajectories

Visual processing



Calibration

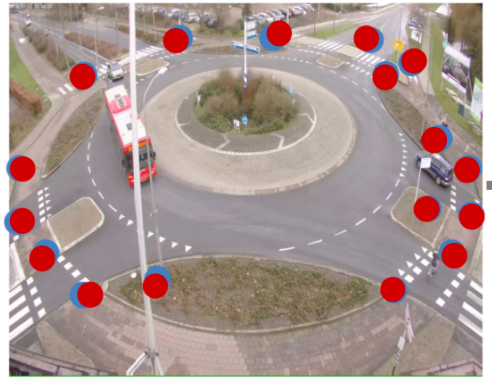
Detection

Tracking

We use Mask R-CNN (He et al., 2018) to detect the bounding boxes of the objects in the scene and map them in 3D.

Extracting trajectories

Visual processing



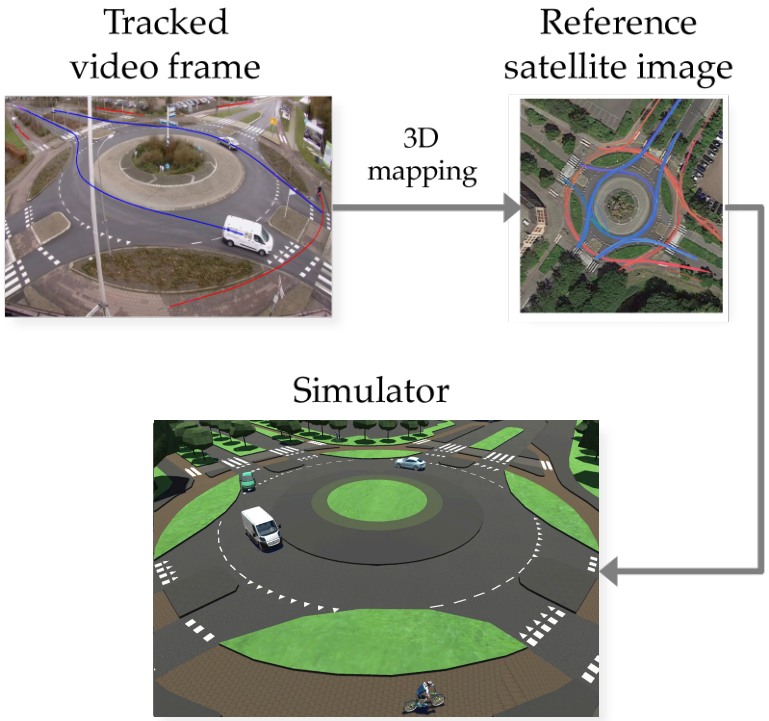
Calibration

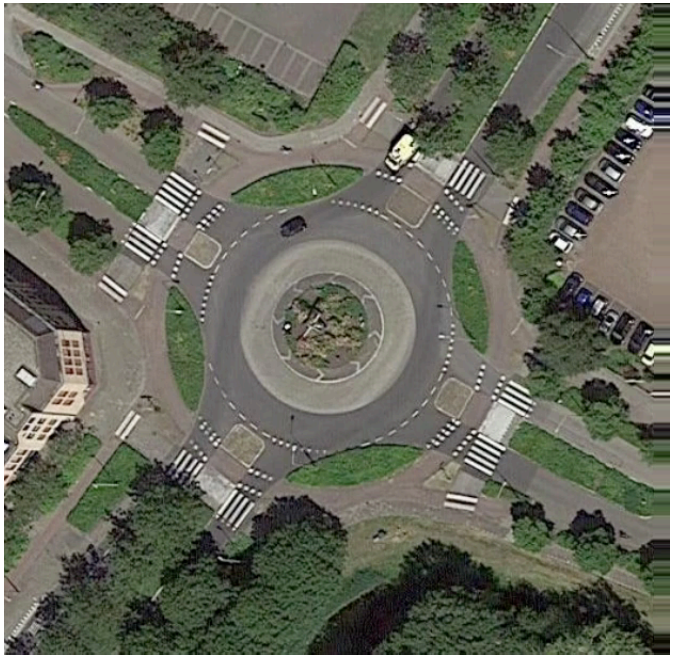
Detection

Tracking

We track the detected objects through time using image features and Kalman filter in 3D.

Results: Extracting trajectories

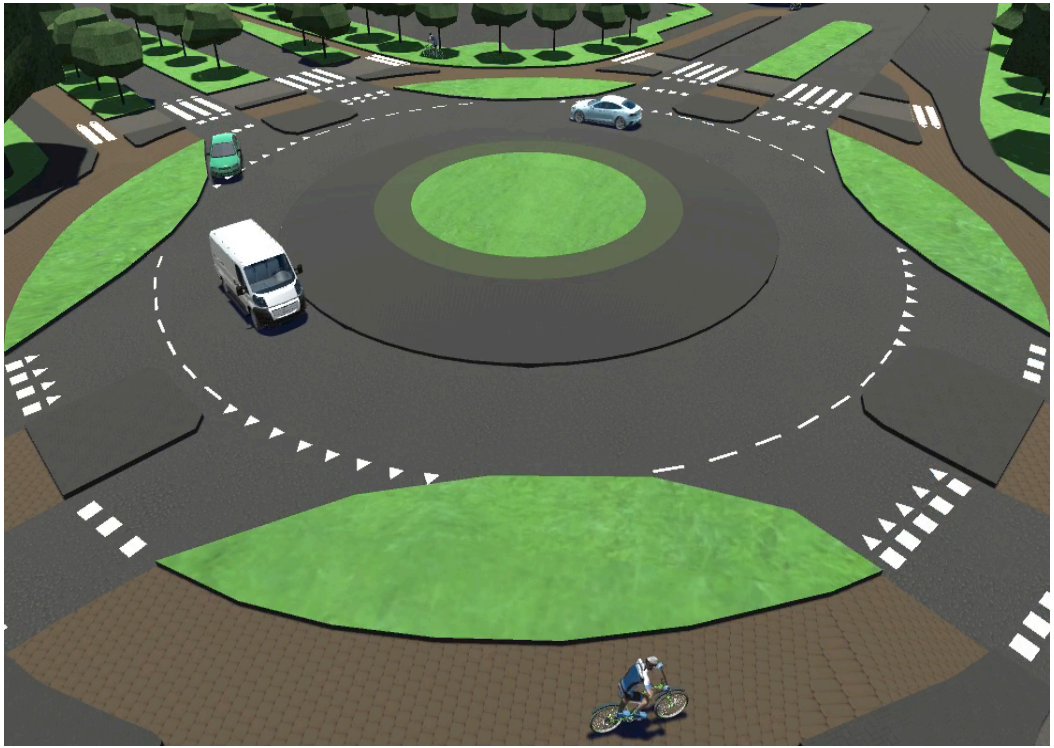




Simulator

Built simulator of scene in Unity:

- Reproduces scene accurately in 3D
- Produces observations (e.g. LIDAR, RGB observations)
- Accepts external actions from agents or humans.

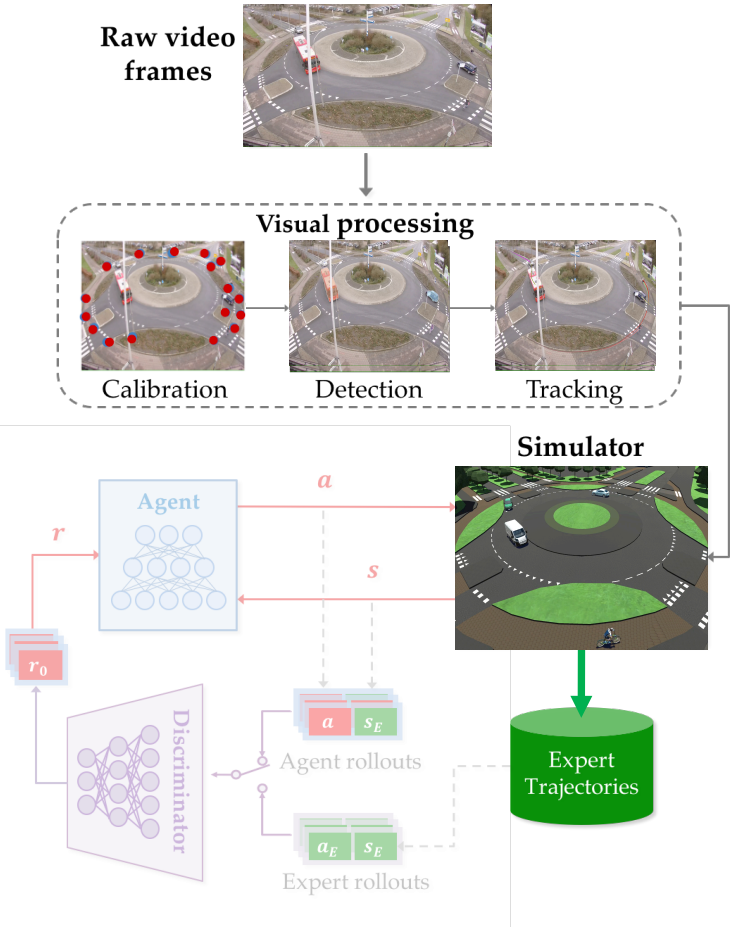


Generation of demonstrations

The simulator is used to replay all extracted trajectories and produce a dataset of expert demonstrations.

State contains:

- Pseudo-LiDAR readings representing of static (zebra crossings and roads) and dynamic (distance and velocity of other agents) context of the agent
- Agent's heading and velocity.
- Target exit and distance to reaching it.



Learning

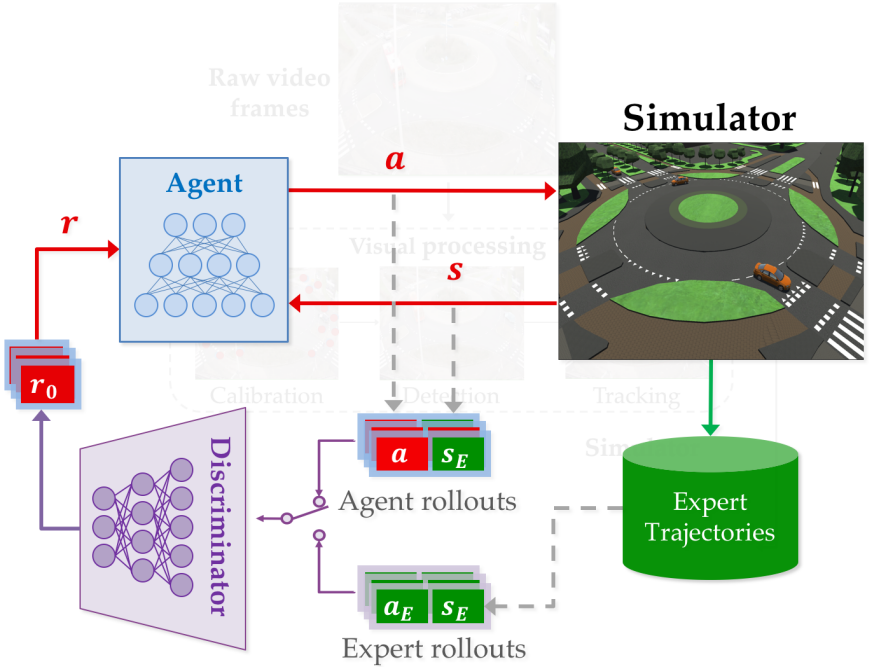
Given dataset of expert trajectories and simulator, learn a policy to mimic expert behaviour.

Use GAIL to learn agent policy.

- Agent trained with PPO (actor-critic)

Issues:

- Multi-agent situation, complicates matters
- Suffers from instabilities during training
- Sensitive to hyperparameters



$$\min_{\theta} \max_{\phi} \mathbb{E}_{\pi_\theta} \left[\log(\underbrace{D_\phi(s,a)}) \right] + \mathbb{E}_{\pi_E} \left[\log(1 - \underbrace{D_\phi(s^E, a^E)}) \right]$$

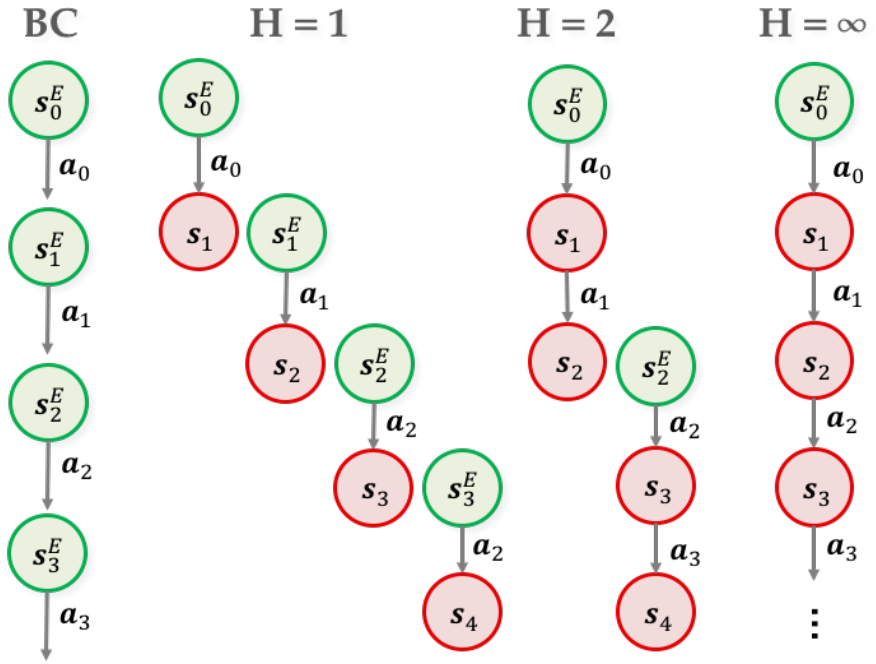
Discriminator output for agent

Discriminator output for expert

Horizon-GAIL

Solve this problem with a novel curriculum

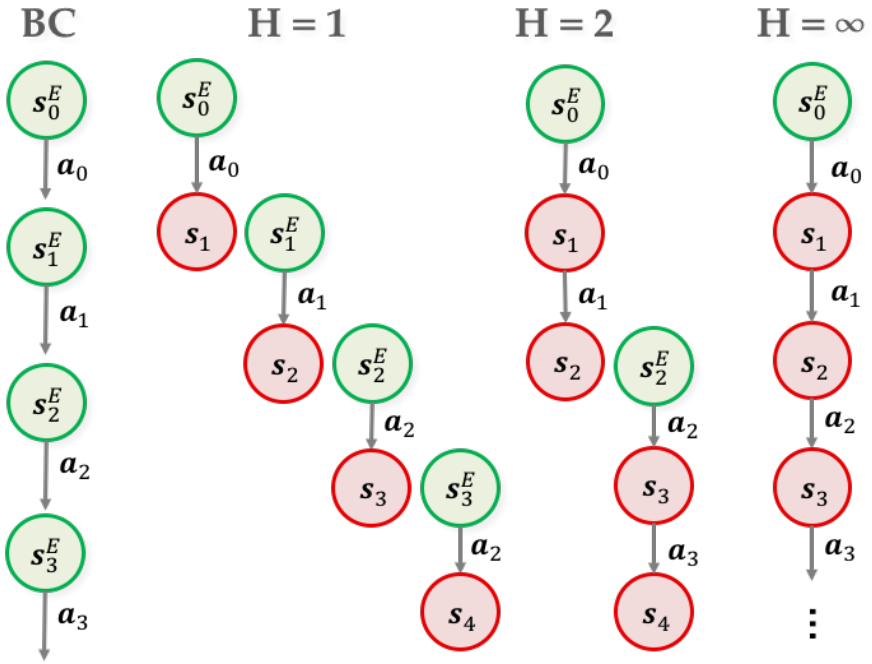
- Bootstraps learning from expert's state (akin to BC)
- Gradually increase number of timesteps that the agent can interact with the simulator, avoiding compounding errors.



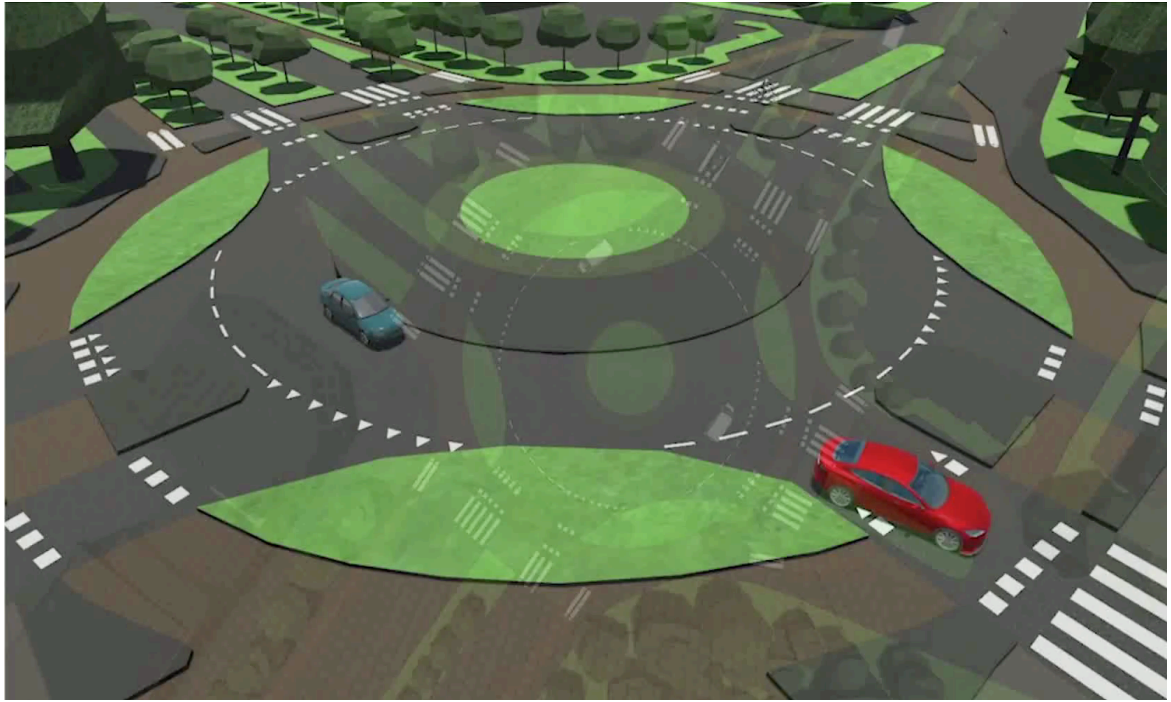
Horizon-GAIL

Solve this problem with a novel curriculum

- Bootstraps learning from expert's state (akin to BC)
- Gradually increase number of timesteps that the agent can interact with the simulator, avoiding compounding errors.
- Encourages the discriminator to learn better representations of the expert distribution early on.
- Allows agent and discriminator to jointly learn to generalise to longer sequences of behaviour.



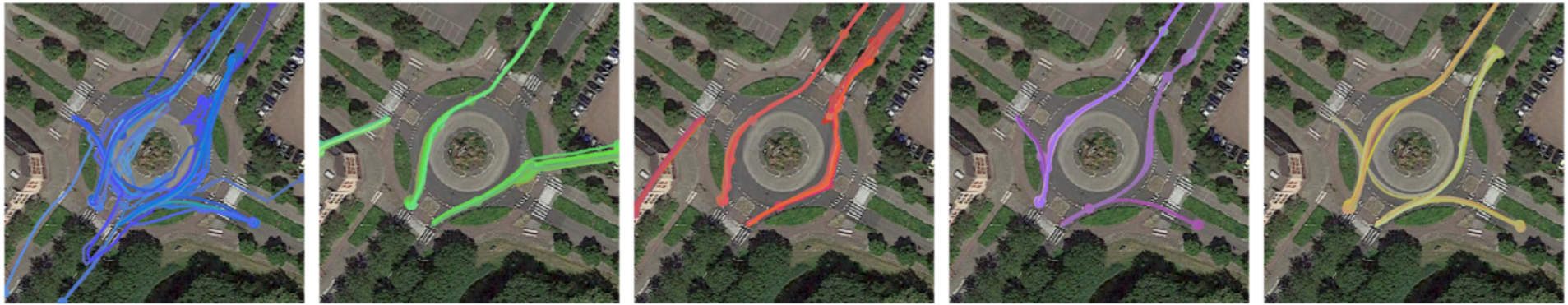
Results: videos of behaviour in simulation



Our method yields stable, plausible trajectories with fewer collisions than any other baseline methods.

Results: Comparison to other methods

Birds-eye view of trajectories taken by different agents.



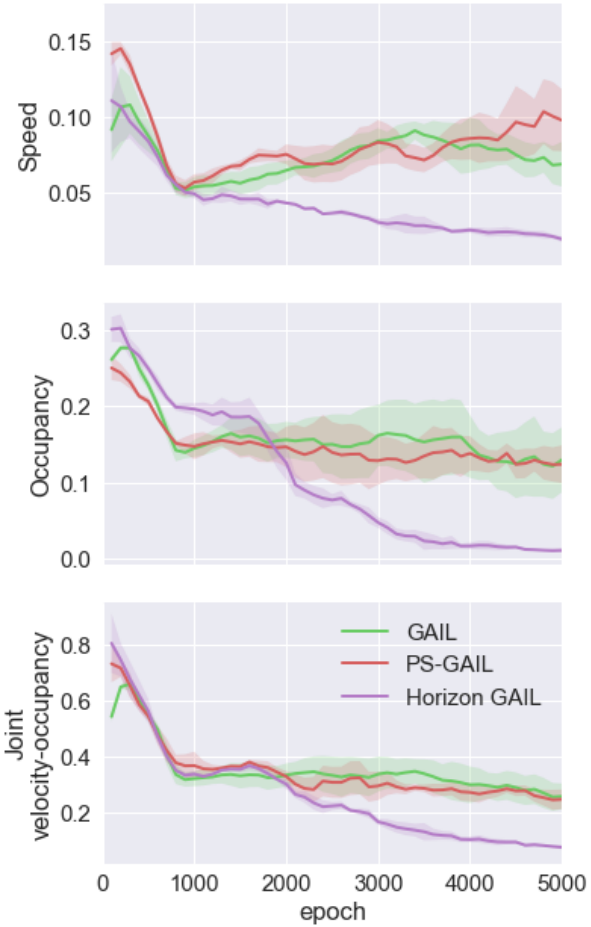
BC GAIL PS-GAIL Horizon GAIL Expert data

Results: comparison using metrics

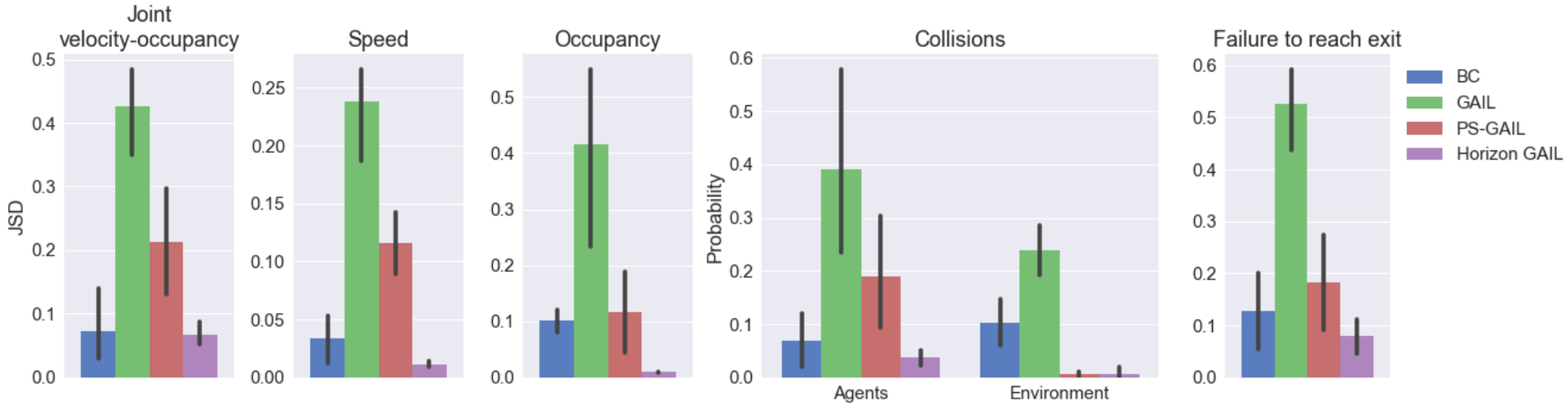
Unlike RL, evaluating LfD is not straightforward!

- Typically no single metric suffices…
- We measure speed profile, occupancy (i.e. locations in 2D space), and joint distribution of velocities and space occupancy for agent and expert.
- Measure Jensen-Shannon divergence (JSD) between expert and agent for each distribution.

Horizon-GAIL much more stable during training, and more robust to random seeds.



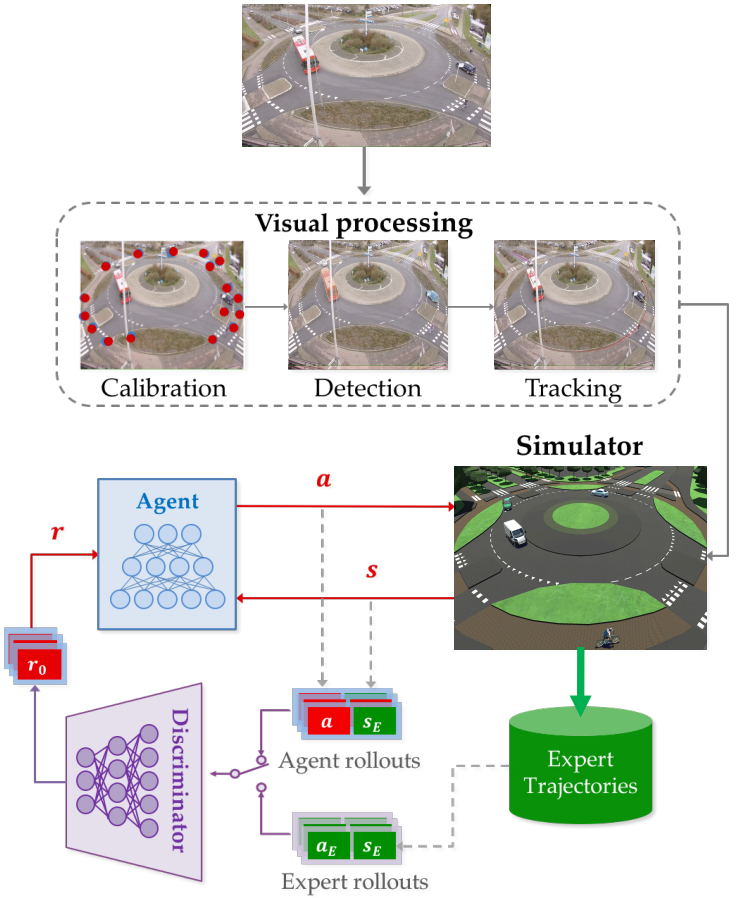
Results: comparison using metrics



Performance of all models for 4 independent, 4000 timestep multi-agent simulation after 5000 epochs of training

Recap

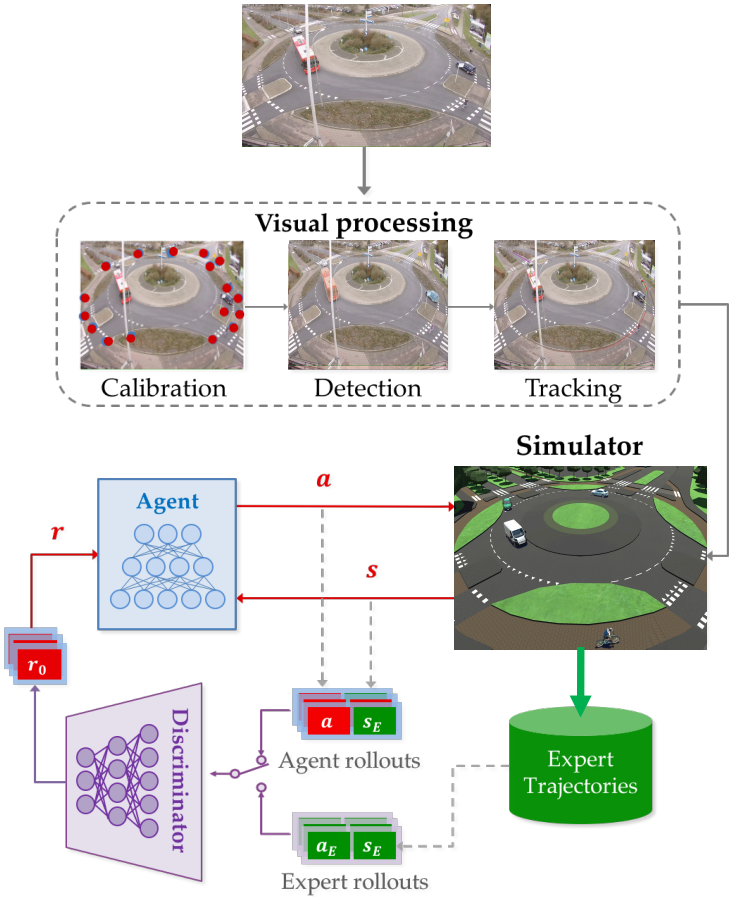
- ViBe allows to extract robust driving behaviours from raw videos.
- Building a real-world simulator can help assess the safety of autonomous driving vehicles and of course more realistic animation and game-play!



Recap

- ViBe allows to extract robust driving behaviours from raw videos.
- Building a real-world simulator can help assess the safety of autonomous driving vehicles and of course more realistic animation and game-play!
- Concurrent work in learning acrobatics (Peng et al., 2018)

Cartwheel A



Challenges and future work

Diverse behaviour?

If demonstrations come from different experts, how to capture the multi-modality?

- Cluster trajectories a-priori, learn independent models
- Provide conditioning information to policy and discriminator

Learn trajectory embeddings using VAEs (Wang et. al, 2017)
Conditional GANs: **InfoGAIL** (Li et. al, 2017)

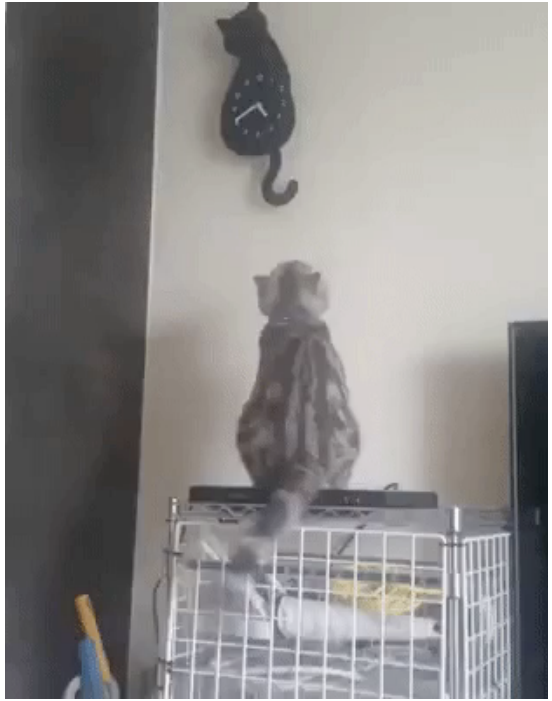
Challenges and future work

Third-person imitation?

Learn some invariant feature map, feed that to your discriminator (Stadie et al., 2017)

Learn how to transform demonstrations into learner's perspective (Liu et al., 2017)

Work still needed to extend this to real-world data.



Challenges and future work

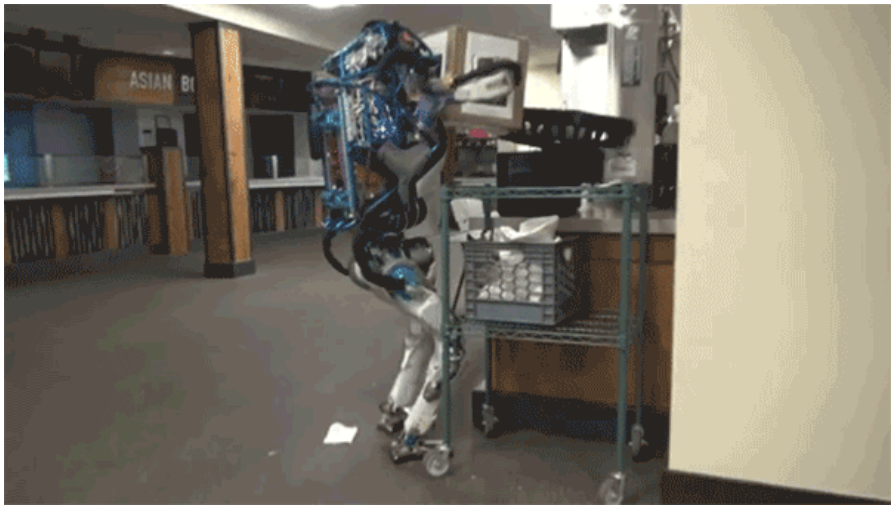
Prevent undesirable/unsafe behaviour?

Error-free policies virtually impossible to learn from demonstrations alone.
How to fix undesirable behaviour (e.g. going off-road)?

Engineer on top of learned policy?

- Limit / override agent actions
  - Can corrupt state in recurrent models
- LfD + hand engineer rewards
  - Can destroy ‘human-like’ behavior
  - Greedy for unseen states

Lacotte et al., 2018. **Risk-Sensitive GAIL**



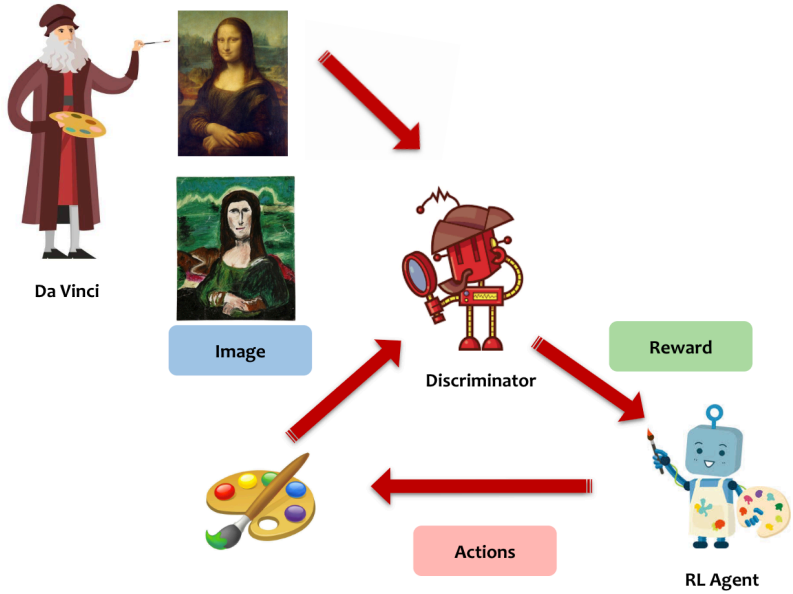
Challenges and future work

How to evaluate learning?

Eyeballing and cherry-picking are not the best approaches (Lucic et al., 2017. **Are GANs Created Equal?**)

Use real-life metrics specific to your domain (e.g. collision frewuency in traffic domain, Kuefler et al., 2017)

Analytical approaches to evaluate model quality: (Odena et al., 2018. **Is Generator Conditioning Causally Related to GAN Performance?)**



Summary

- If we want to capture interesting and complex behaviours, we can’t rely solely on hard-coding reward functions
- We can leverage the plethora of data available in the wild
- Our work offers one of the first attempts at doing so in the context of traffic scenes, but could hopefully be useful in many different settings where there is abundance of video data

Thanks for your attention!

and to many collaborators:

Kyriacos Shiarlis, Xi Chen, Vitaly Kurin, Sudhanshu Kasewa, Ciprian Stirbu, Joao Gomes, Supratik Paul, Jakob Howard, Paul Mougin, Omar Makhlouf, Frans A. Oliehoek, Kirsty Lloyd-Jukes, Joao Messias, and Shimon Whiteson

and brilliant interns:

Rishabh Agarwal (now at Google Brain)
Daniel Marta (now at Delft University)

We're hiring!


Feryal Behbahani
feryal@latentlogic.com


feryal.github.io
www.LatentLogic.com


@feryalmp
@latent_logic

RL recap

state $s \in S$

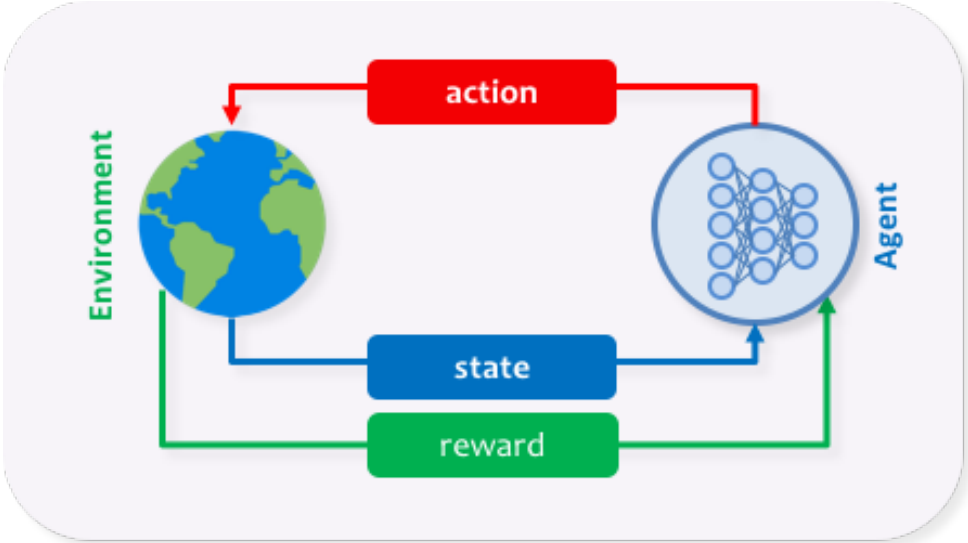
action $a \in A$

reward $r \in \mathbb{R}$

discount factor $\gamma \in [0, 1]$

policy $\pi : S \to A$

value $V : S \to \mathbb{R}$



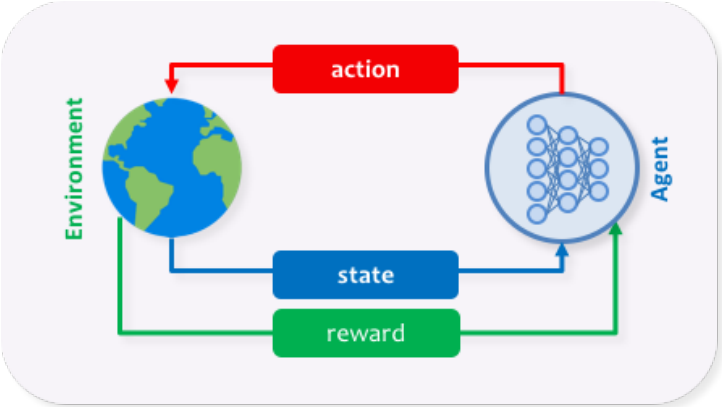
Agent interacts with the environment in order to maximise expected rewards!

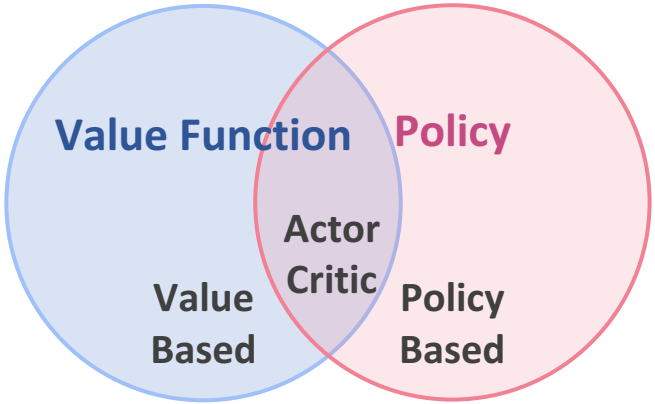
- Decisions are sequential; agent determines what it sees (non i.i.d data)
- Feedback is usually delayed
- No supervisor only reward function

RL recap

How to train an agent to maximise rewards?

- **Value based:**
  - Estimate optimal **Value function**
  - Implicit policy greedily uses it (Q-Learning, DQN)
- **Policy based:**
  - Estimate directly for the optimal **policy**
  - Policy-gradient methods (REINFORCE, TRPO, DDPG)
- **Actor-Critic:**
  - Estimate both Value function and optimal policy
  - Value function used to reduce variance of policy gradient estimator (A3C/IMPALA, ACKTR)





Challenges and future work

How to simulate other agents?

When context is important, e.g. multi-agent tasks, you need to realistically simulate everything that affects your agent!

How?

- Classic AI methods (e.g. A*)
- Original data
  - not closed-loop…
- Single reactive behaviour
  - chicken and egg problem
- Train all agents simultaneously
  - Need to handle co-training peculiarities