Randomized Adversarial Imitation Learning

Authors: MyungJae Shin (Presenter) and Joongheon Kim
School of Computer Science and Engineering, Chung-Ang University, Seoul, Republic of Korea
Emails: mjshin.cau@gmail.com , joongheon@gmail.com
Sites: https://github.com/170928 , https://sites.google.com/site/joongheonkim/

Outline

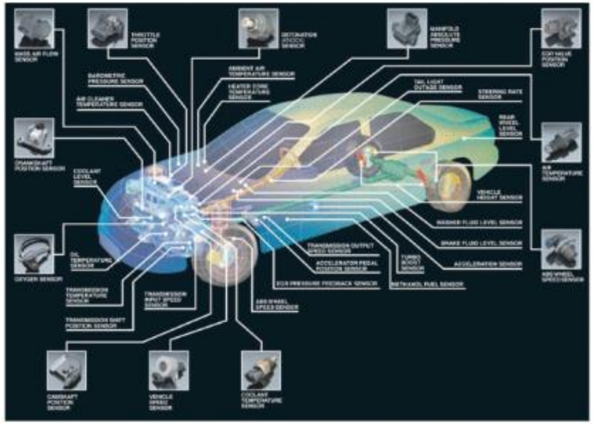
Background

Technologies

- High-performance Sensors
- Assistance System
- Deep Learning

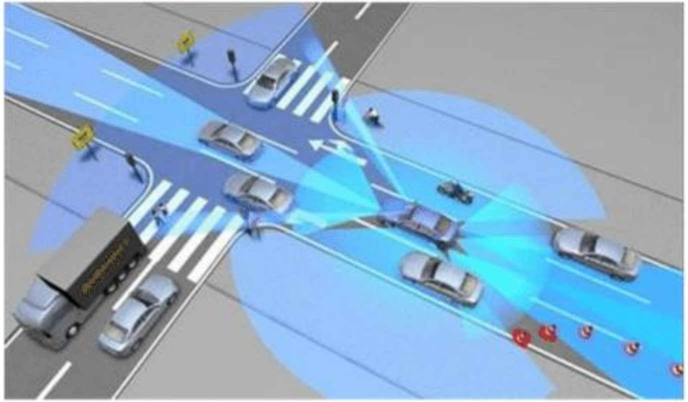
Background

Vehicle Sensors



- Various sensors for vehicle
  (e.g. LIDAR, RADAR, ⋯)
- High performance
- Sensor fusion techniques

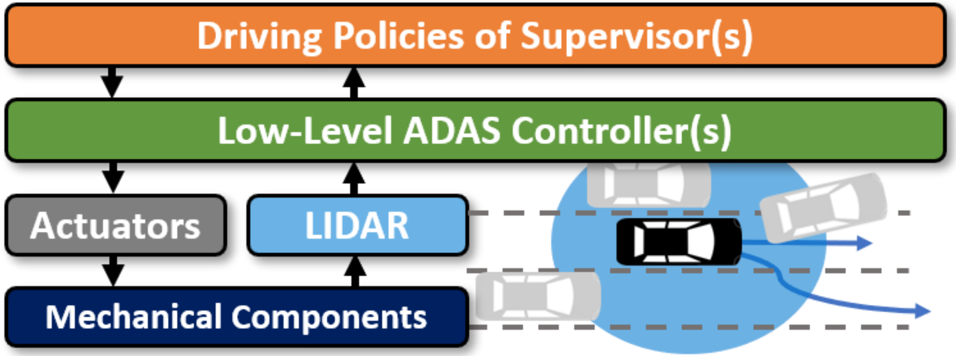
ADAS Algorithms



- Various ADAS
  (e.g. AEB, LKAS, BSD, ESC, ⋯)
- Already commercialized
- Essential function for safety

(ref) Deep Q Learning Based High Level Driving Policy Determination by Kyusik Min (IV Symposium 2018)

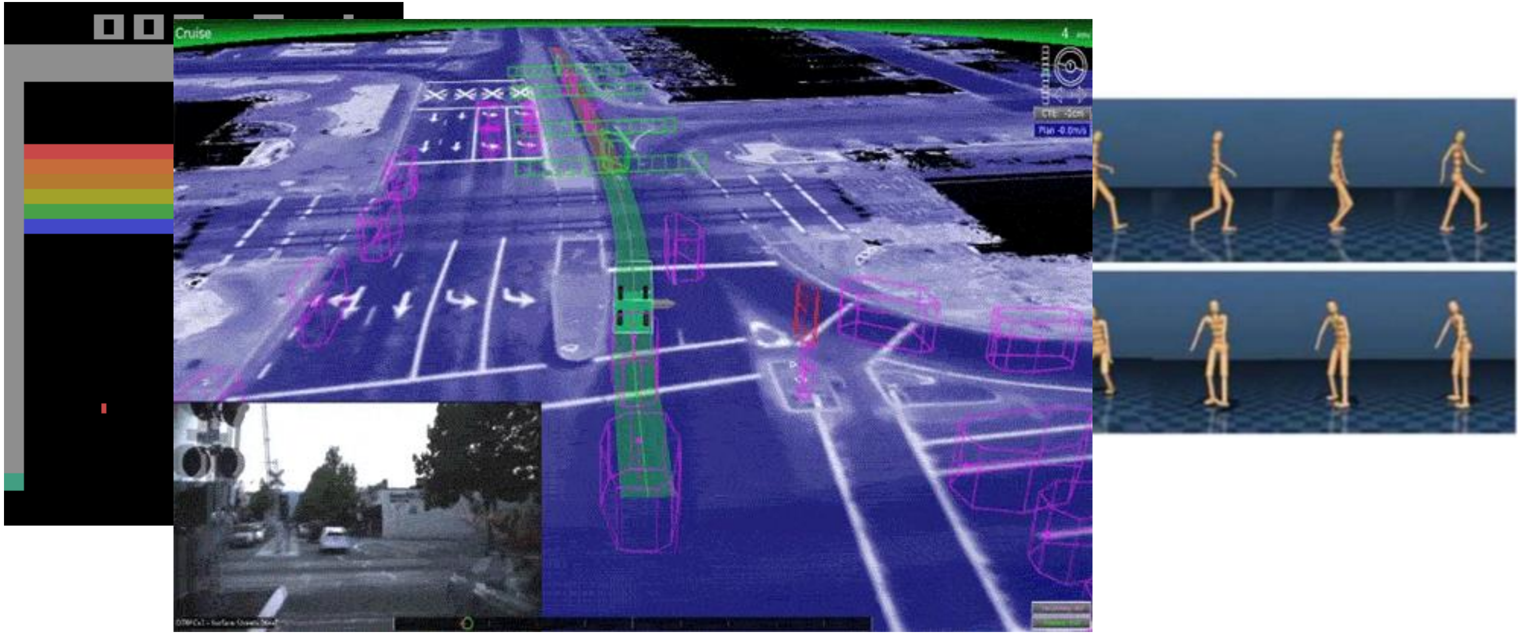
Motivation



- Many driver assistance systems are already commercialized and applied to lots of vehicles.
- Many driver assistance systems perform partial function of autonomous driving.
- Autonomous driving can be achieved by properly choosing driver assistance system in specific scenario.

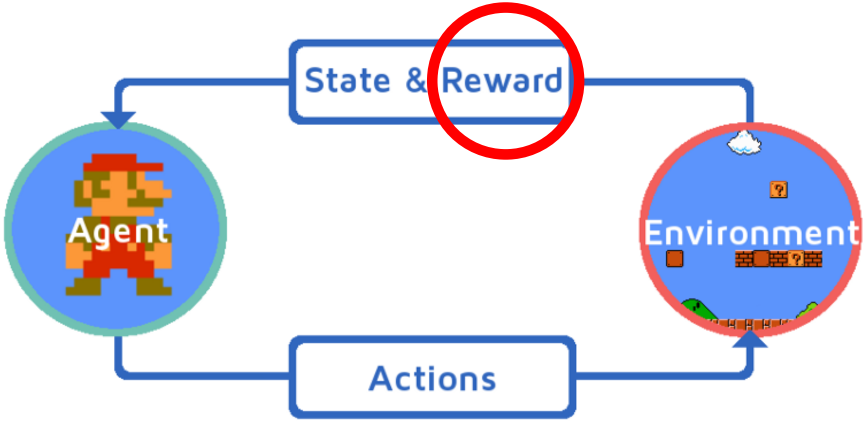
Motivation

Goal : Learn policies
High-dimensional & raw observations



Motivation

Challenge : Provide appropriate cost or reward signal.



- Sparse reward
- Mathematical definition

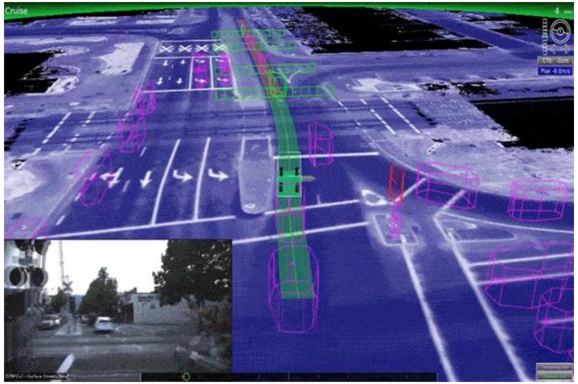
Jump is desired behavior or not?

Motivation

Input : expert behavior generated by expert π_E

$$\{(s_0^i, a_0^i, s_1^i, a_1^i, \dots)\}_{i=1}^{N} \sim \pi_E$$

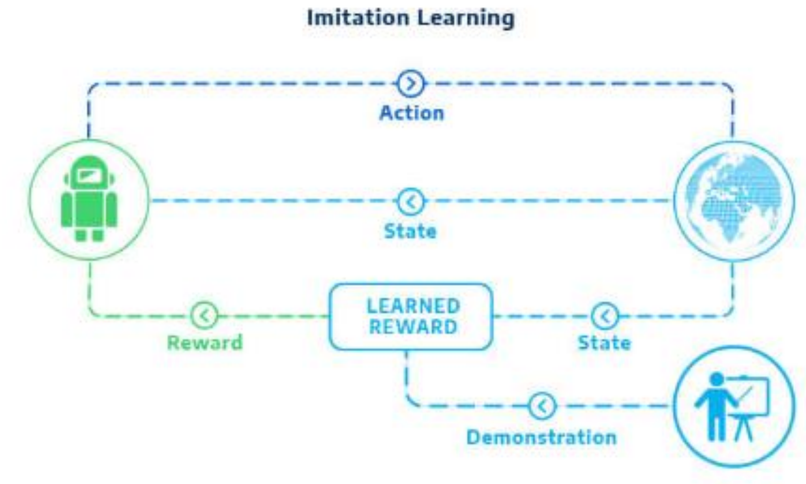
Goal : learn **cost function** or **policy**



Motivation

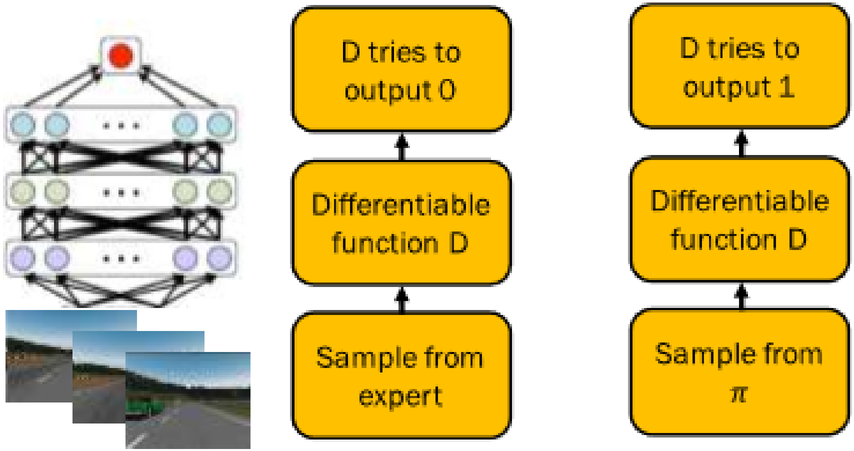
Goal

- Apply imitation learning to train supervisor of self driving car.
- Enhance safety of the self driving agent during training and testing.
- Implement an algorithm that can be easily parallelized.



Randomized Adversarial Imitation Learning (RAIL)

Generative Adversarial Imitation Learning (GAIL), *NIPS 2016*

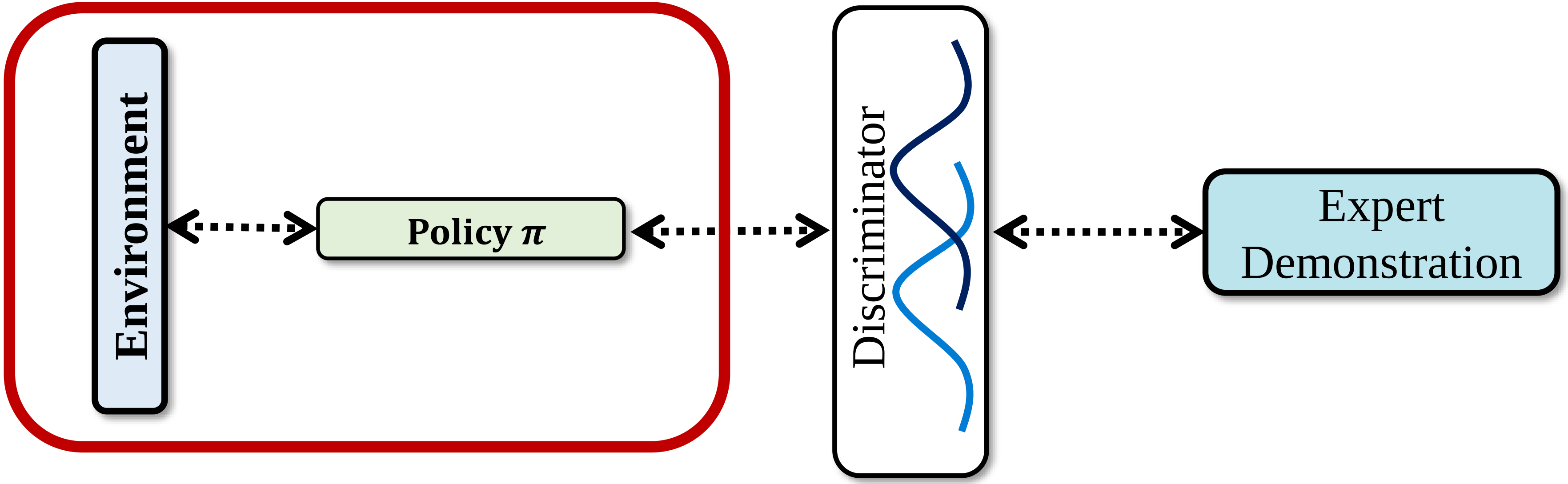


- The quality of the samples we obtain is measured by training the discriminator D.
- The policy is trained to produce behaviors that are difficult to distinguish from expert.

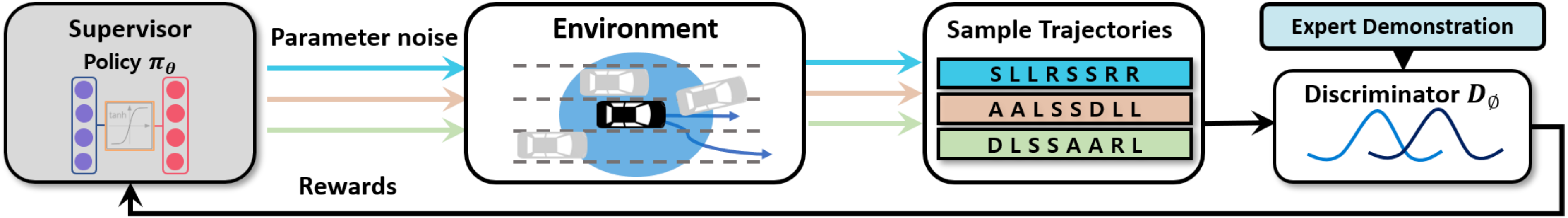
Randomized Adversarial Imitation Learning (RAIL)

Challenge

- A lot of interaction with the environment is required to optimize the policy through GAIL framework
- Hard to be parallelized.



Randomized Adversarial Imitation Learning (RAIL)



$$minimize \;\; \mathbb{E}_{\pi}[\log(D(s,a)] + \mathbb{E}_{\pi_E}[\log(1 - D(s,a)]$$

$\boldsymbol{D(s,a)}$ **: Probability between 0 and 1**

The probability that the input data sample is the expert data sample

Randomized Adversarial Imitation Learning (RAIL)

Usually in AI:

$$f'(x) = \frac{df}{dx}$$

- To update the weights of policy the gradient descent method is used.

Method of finite differences

$$f'(a) = \frac{f(a+h) - f(a)}{h}$$

- To update the weights of policy **the method of finite differences.**

Randomized Adversarial Imitation Learning (RAIL)

Policy

θ_t

$$\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix} \pm \begin{bmatrix} 0.14 & 0.37 & 0.49 \\ 0.96 & 0.25 & 0.94 \end{bmatrix}$$

δ_i

$$\begin{bmatrix} w_{1,1} + 0.14 & w_{1,2} + 0.37 & w_{1,3} + 0.49 \\ w_{2,1} + 0.96 & w_{2,2} + 0.25 & w_{2,3} + 0.94 \end{bmatrix}$$

$$\begin{bmatrix} w_{1,1} - 0.14 & w_{1,2} - 0.37 & w_{1,3} - 0.49 \\ w_{2,1} - 0.96 & w_{2,2} - 0.25 & w_{2,3} - 0.94 \end{bmatrix}$$

Gaussian Noise

$$\boldsymbol{\delta} = \{\delta_1, \delta_2, \cdots, \delta_N\}$$

random numbers or a matrix with random tiny values

Randomized Adversarial Imitation Learning (RAIL)

Positive perturbative weights

$$R_{d-pos} = \begin{bmatrix} w_{1,1} + d_{1,1} & w_{1,2} + d_{1,2} & w_{1,3} + d_{1,3} \\ w_{2,1} + d_{2,1} & w_{2,2} + d_{2,2} & w_{2,3} + d_{2,3} \end{bmatrix}$$

$$R_{e-pos} = \begin{bmatrix} w_{1,1} + e_{1,1} & w_{1,2} + e_{1,2} & w_{1,3} + e_{1,3} \\ w_{2,1} + e_{2,1} & w_{2,2} + e_{2,2} & w_{2,3} + e_{2,3} \end{bmatrix}$$

$$R_{f-pos} = \begin{bmatrix} w_{1,1} + f_{1,1} & w_{1,2} + f_{1,2} & w_{1,3} + f_{1,3} \\ w_{2,1} + f_{2,1} & w_{2,2} + f_{2,2} & w_{2,3} + f_{2,3} \end{bmatrix}$$

$$R_{g-pos} = \begin{bmatrix} w_{1,1} + g_{1,1} & w_{1,2} + g_{1,2} & w_{1,3} + g_{1,3} \\ w_{2,1} + g_{2,1} & w_{2,2} + g_{2,2} & w_{2,3} + g_{2,3} \end{bmatrix}$$

Negative perturbative weights

$$R_{d-neg} = \begin{bmatrix} w_{1,1} - d_{1,1} & w_{1,2} - d_{1,2} & w_{1,3} - d_{1,3} \\ w_{2,1} - d_{2,1} & w_{2,2} - d_{2,2} & w_{2,3} - d_{2,3} \end{bmatrix}$$

$$R_{e-neg} = \begin{bmatrix} w_{1,1} - e_{1,1} & w_{1,2} - e_{1,2} & w_{1,3} - e_{1,3} \\ w_{2,1} - e_{2,1} & w_{2,2} - e_{2,2} & w_{2,3} - e_{2,3} \end{bmatrix}$$

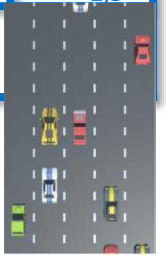
$$R_{f-neg} = \begin{bmatrix} w_{1,1} - f_{1,1} & w_{1,2} - f_{1,2} & w_{1,3} - f_{1,3} \\ w_{2,1} - f_{2,1} & w_{2,2} - f_{2,2} & w_{2,3} - f_{2,3} \end{bmatrix}$$

$$R_{g-neg} = \begin{bmatrix} w_{1,1} - g_{1,1} & w_{1,2} - g_{1,2} & w_{1,3} - g_{1,3} \\ w_{2,1} - g_{2,1} & w_{2,2} - g_{2,2} & w_{2,3} - g_{2,3} \end{bmatrix}$$

Randomized Adversarial Imitation Learning (RAIL)

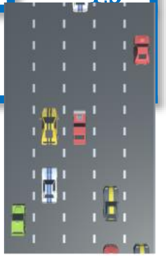
R_{d-pos}

$w_{1,1} + d_{1,1}$	$w_{1,2} + d_{1,2}$	$w_{1,3} + d_{1,3}$
$w_{2,1} + d_{2,1}$	$w_{2,2} + d_{2,2}$	



R_{e-pos}

$w_{1,1} + e_{1,1}$	$w_{1,2} + e_{1,2}$	$w_{1,3} + e_{1,3}$
$w_{2,1} + e_{2,1}$	$w_{2,2} + e_{2,2}$	



R_{f-pos}

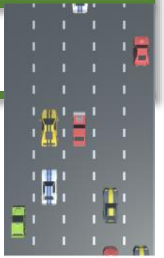
$w_{1,1} + f_{1,1}$	$w_{1,2} + f_{1,2}$	$w_{1,3} + f_{1,3}$
$w_{2,1} + f_{2,1}$	$w_{2,2} + f_{2,2}$	

R_{g-pos}

$w_{1,1} + g_{1,1}$	$w_{1,2} + g_{1,2}$	$w_{1,3} + g_{1,3}$
$w_{2,1} + g_{2,1}$	$w_{2,2} + g_{2,2}$	

R_{d-neg}

$w_{1,1} - d_{1,1}$	$w_{1,2} - d_{1,2}$	$w_{1,3} - d_{1,3}$
$w_{2,1} - d_{2,1}$	$w_{2,2} - d_{2,2}$	



R_{e-neg}

$w_{1,1} - e_{1,1}$	$w_{1,2} - e_{1,2}$	$w_{1,3} - e_{1,3}$
$w_{2,1} - e_{2,1}$	$w_{2,2} - e_{2,2}$	

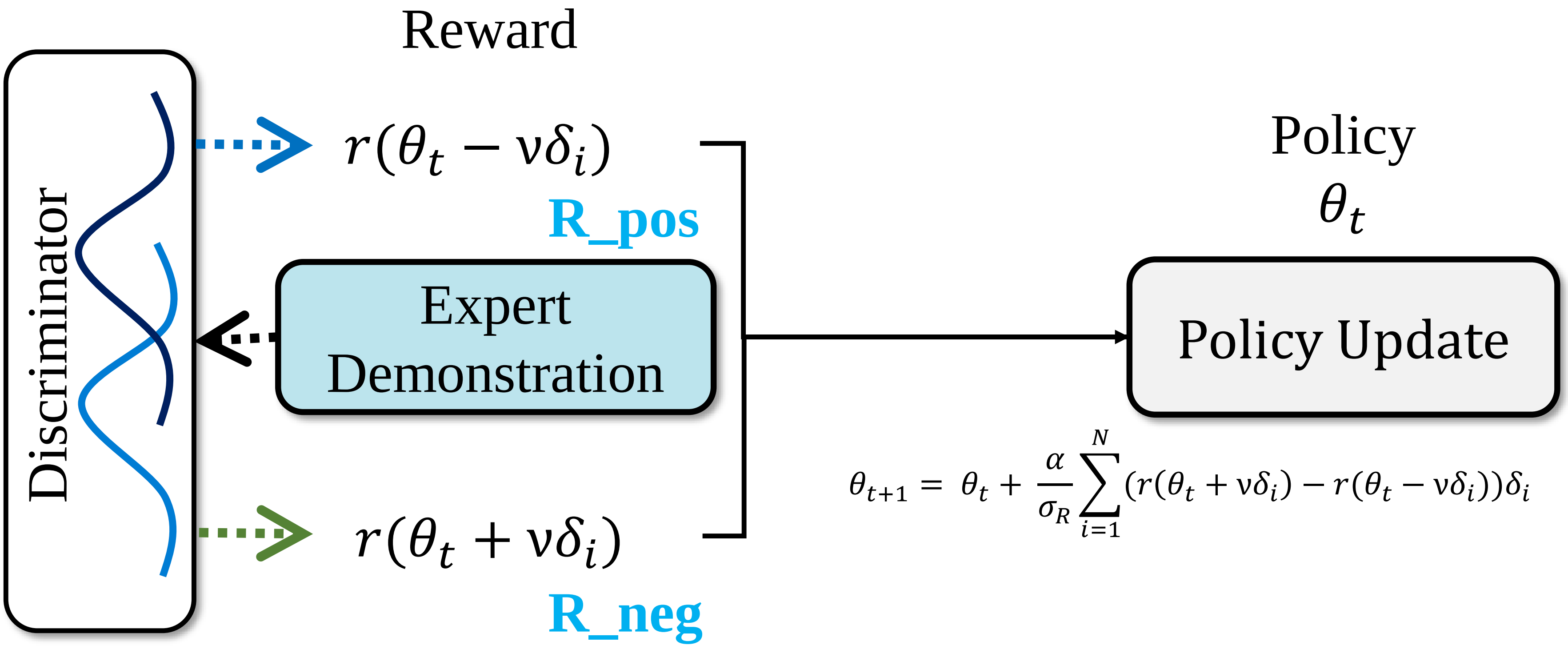
R_{f-neg}

$w_{1,1} - f_{1,1}$	$w_{1,2} - f_{1,2}$	$w_{1,3} - f_{1,3}$
$w_{2,1} - f_{2,1}$	$w_{2,2} - f_{2,2}$	

R_{g-neg}

$w_{1,1} - g_{1,1}$	$w_{1,2} - g_{1,2}$	$w_{1,3} - g_{1,3}$
$w_{2,1} - g_{2,1}$	$w_{2,2} - g_{2,2}$	

Randomized Adversarial Imitation Learning (RAIL)



$$\theta_{t+1} = \theta_t + \frac{\alpha}{\sigma_R} \sum_{i=1}^{N} (r(\theta_t + v\delta_i) - r(\theta_t - v\delta_i))\delta_i$$

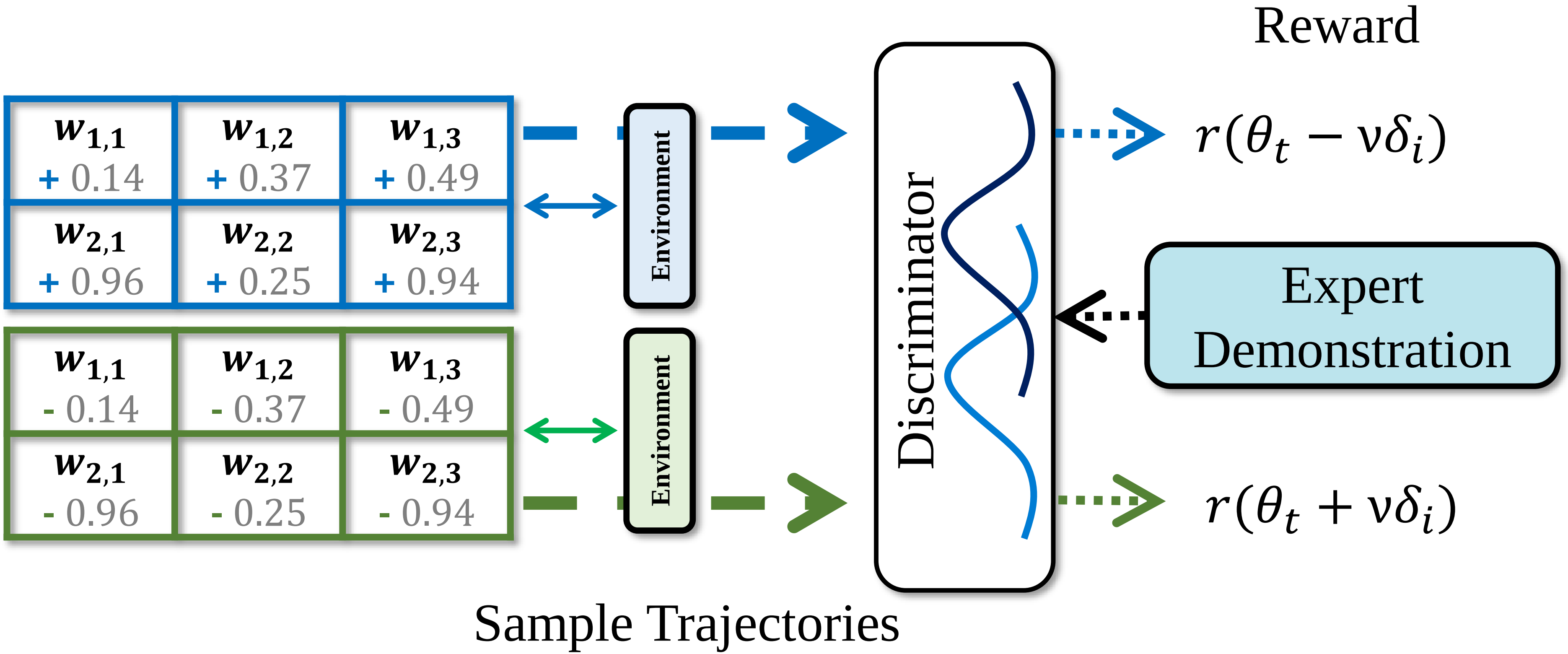
Randomized Adversarial Imitation Learning (RAIL)

$$\underset{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}}{\overset{\text{Policy}}{\theta_{t+1}}} = \underset{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}}{\overset{\text{Policy}}{\theta_{t}}} + \left[\begin{array}{l} (R_{d-pos} - R_{d-neg}) * \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,1} & d_{2,2} & d_{2,3} \end{bmatrix} + \\ (R_{e-pos} - R_{e-neg}) * \begin{bmatrix} e_{1,1} & e_{1,2} & e_{1,3} \\ e_{2,1} & e_{2,2} & e_{2,3} \end{bmatrix} + \\ \vdots \end{array} \right]$$

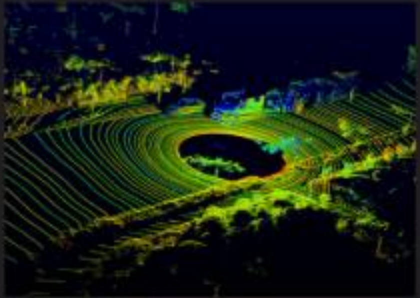
Randomized Adversarial Imitation Learning (RAIL)

$$\overset{\substack{\text{Policy} \\ \theta_{t+1}}}{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}} = \overset{\substack{\text{Policy} \\ \theta_{t}}}{\begin{bmatrix} w_{1,1} & w_{1,2} & w_{1,3} \\ w_{2,1} & w_{2,2} & w_{2,3} \end{bmatrix}} + \left[\begin{array}{l} (R_{d-pos} * \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,1} & d_{2,2} & d_{2,3} \end{bmatrix} \\ \qquad - R_{d-neg} * \begin{bmatrix} d_{1,1} & d_{1,2} & d_{1,3} \\ d_{2,1} & d_{2,2} & d_{2,3} \end{bmatrix}) \quad + \\ \qquad\qquad \vdots \end{array} \right]$$

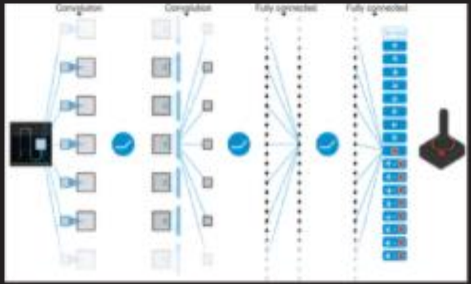
Randomized Adversarial Imitation Learning (RAIL)

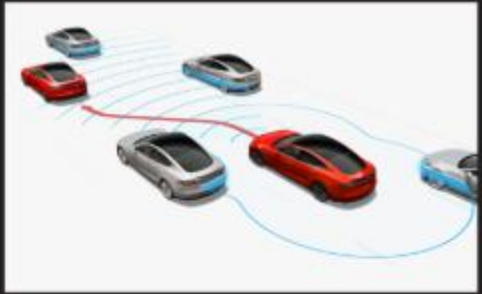


Randomized Adversarial Imitation Learning (RAIL)

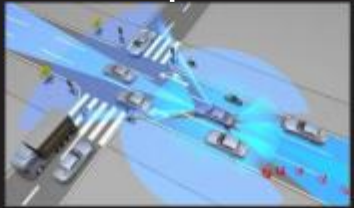


Sensor Data



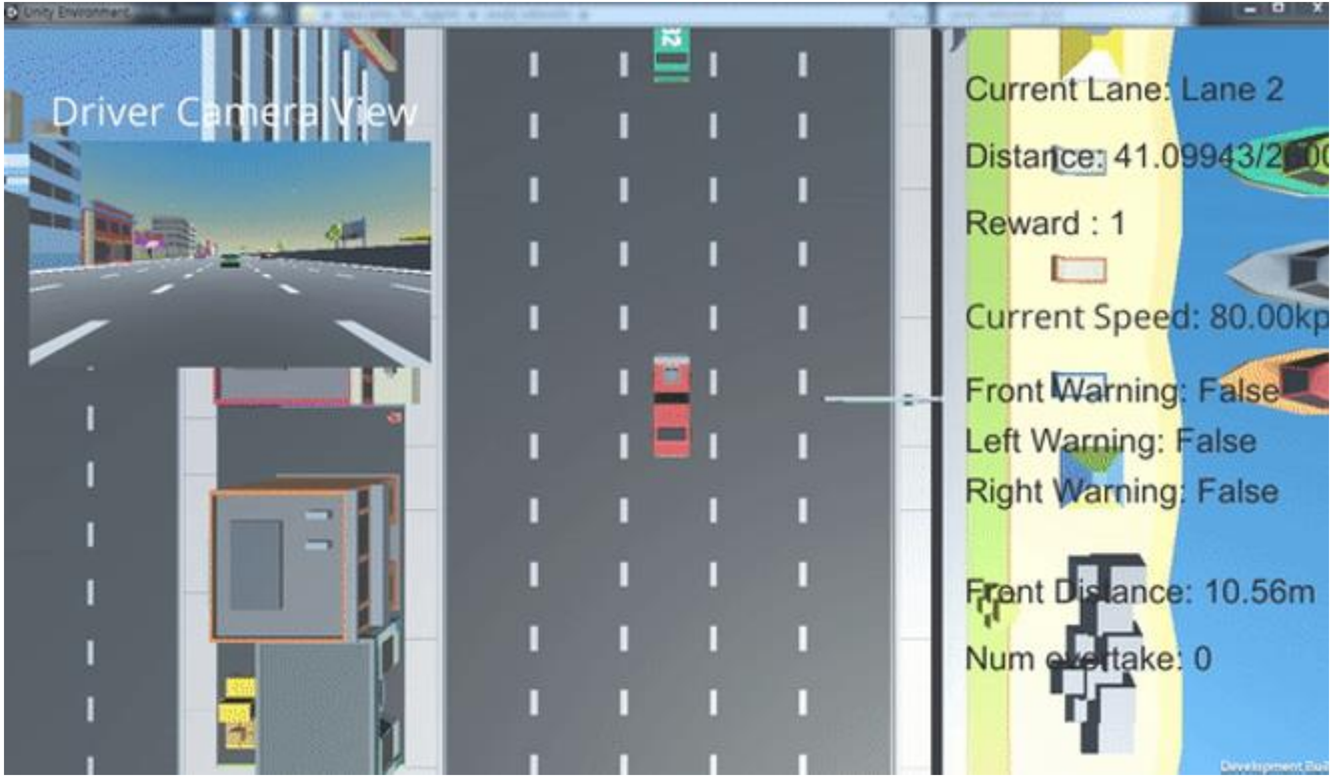


Vehicle Control

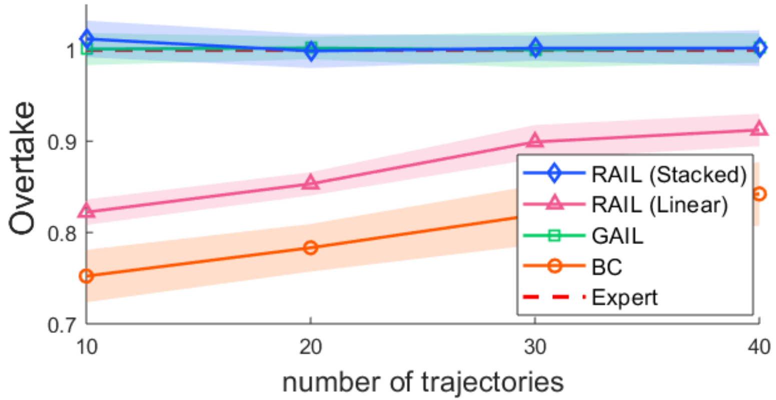


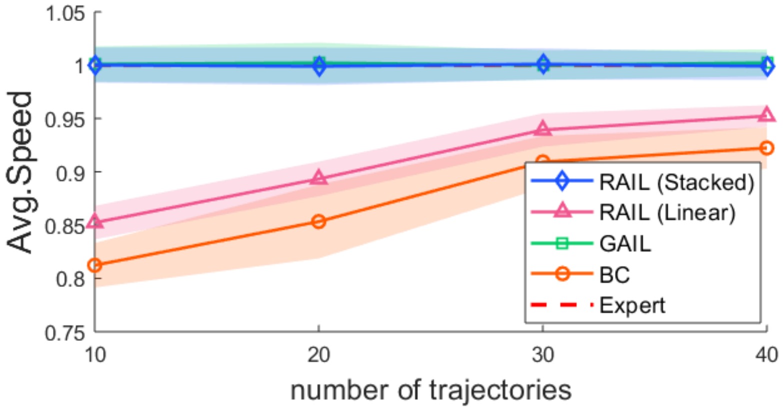
ADAS

Randomized Adversarial Imitation Learning (RAIL)



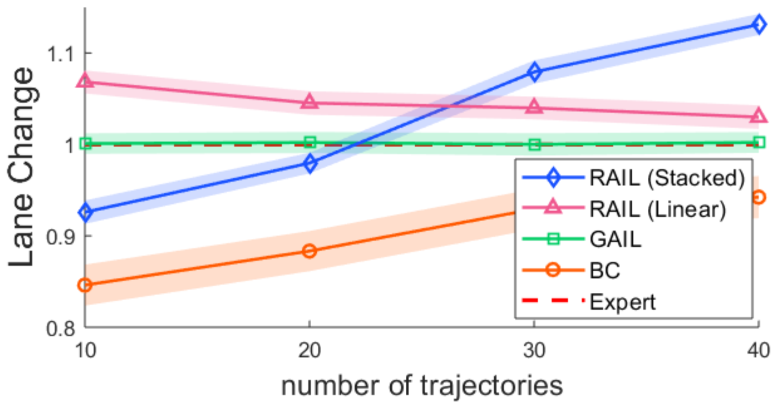
Experiments

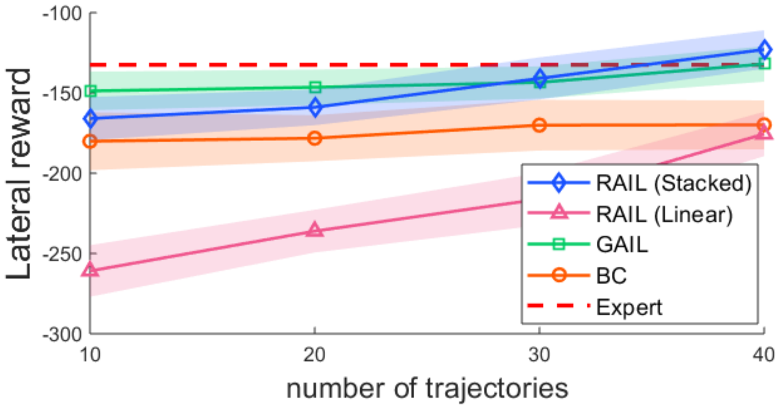




Experiments

Average	RAIL (Stacked)	RAIL (Linear)	Expert
Speed [km/h]	70.38	65.00	68.83
# Overtake	45.04	40.03	44.48
# Lane change	15.01	13.05	14.04
Longitudinal	2719.38	2495.57	2642.11
Lateral	-122.98	-175.6	-132.52





Summary

- In autonomous driving system, RAIL is able to train the shallow network supervisor.
- RAIL is easy to parallel processing because only the constant reward values need to be shared between processors.
- RAIL is an algorithm that increases reproducibility with fewer hyperparameters.

Contact

- MyungJae Shin : mjshin.cau@gmail.com
- Joongheon Kim : joongheon@gmail.com

Thank You