Explainability in Reinforcement Learning:

An Application for Powertrain Control

Catherine Laflamme, PhD

Reinforcement Learning

Success Stories

- **AlphaZero (2017):**
  - General-purpose game-playing AI.
  - Chess, shogi, and Go at a superhuman level.
  - Training solely through self-play, without any prior knowledge of the games.

RL in Industry Applications

- Job Shop Scheduling
- Planning in Matrix Production
- Routing
- Energy Management Systems

RL in Industry Applications

Challenges

1. Where and when RL can be beneficial?
   - How big does my state space and action space need to be for RL to beat heuristics?
   - How does stochasticity impact results?

RL in Industry Applications

Challenges

1. Where and when RL can be beneficial?
   - How big does my state space and action space need to be for RL to beat heuristics?
   - How does stochasticity impact results?
2. Simulations
   - Creating simulations / digital twins is hard!
   - Even when you have the simulation, there is the real-world gap.

RL in Industry Applications

Challenges

1. Where and when RL can be beneficial?
   - How big does my state space and action space need to be for RL to beat heuristics?
   - How does stochasticity impact results?
2. Simulations
   - Creating simulations / digital twins is hard!
   - Even when you have the simulation, there is the real-world gap.
3. Implementing a function approximator is hard for companies to do!
   - Lack of transparency/explainability.
   - Lack of standard processes.

RL in Industry Applications

Challenges

1. Where and when RL can be beneficial?
   - How big does my state space and action space need to be for RL to beat heuristics?
   - How does stochasticity impact results?
2. Simulations
   - Creating simulations / digital twins is hard!
   - Even when you have the simulation, there is the real-world gap.
3. Implementing a function approximator is hard for companies to do!
   - Lack of transparency/explainability.
   - Lack of standard processes.

RL in Industry Applications

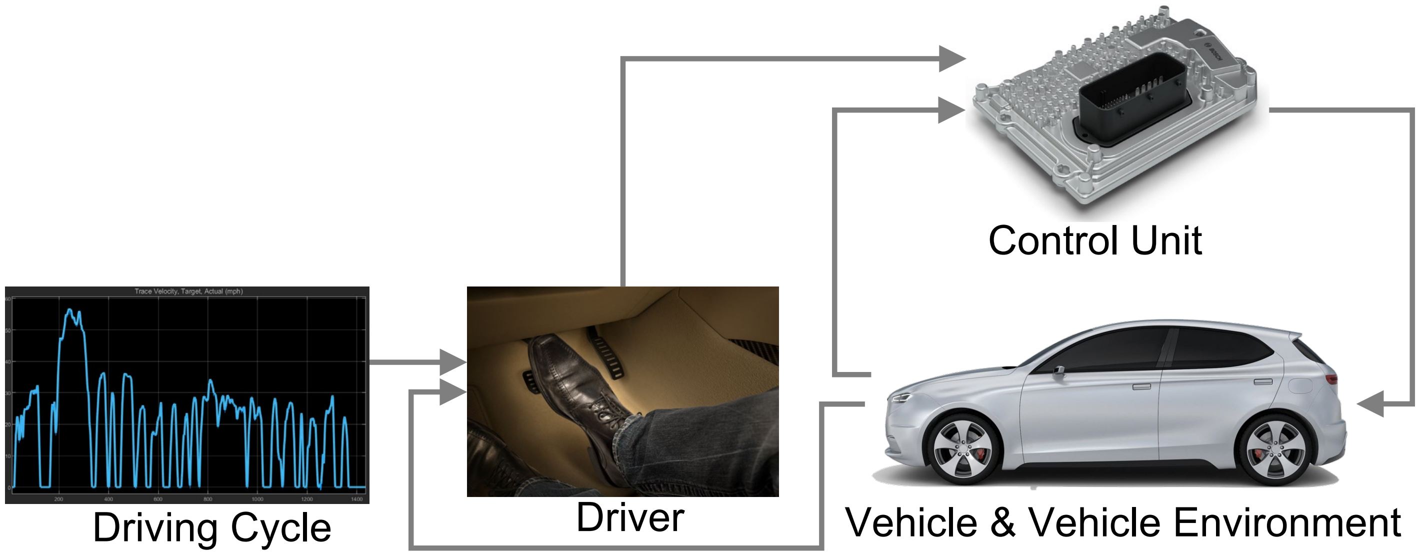
Research Question

- Can we still benefit from an RL Agent without having to implement a neural network directly into control software?

Explainability in Reinforcement Learning

Powertrain Control Use Case

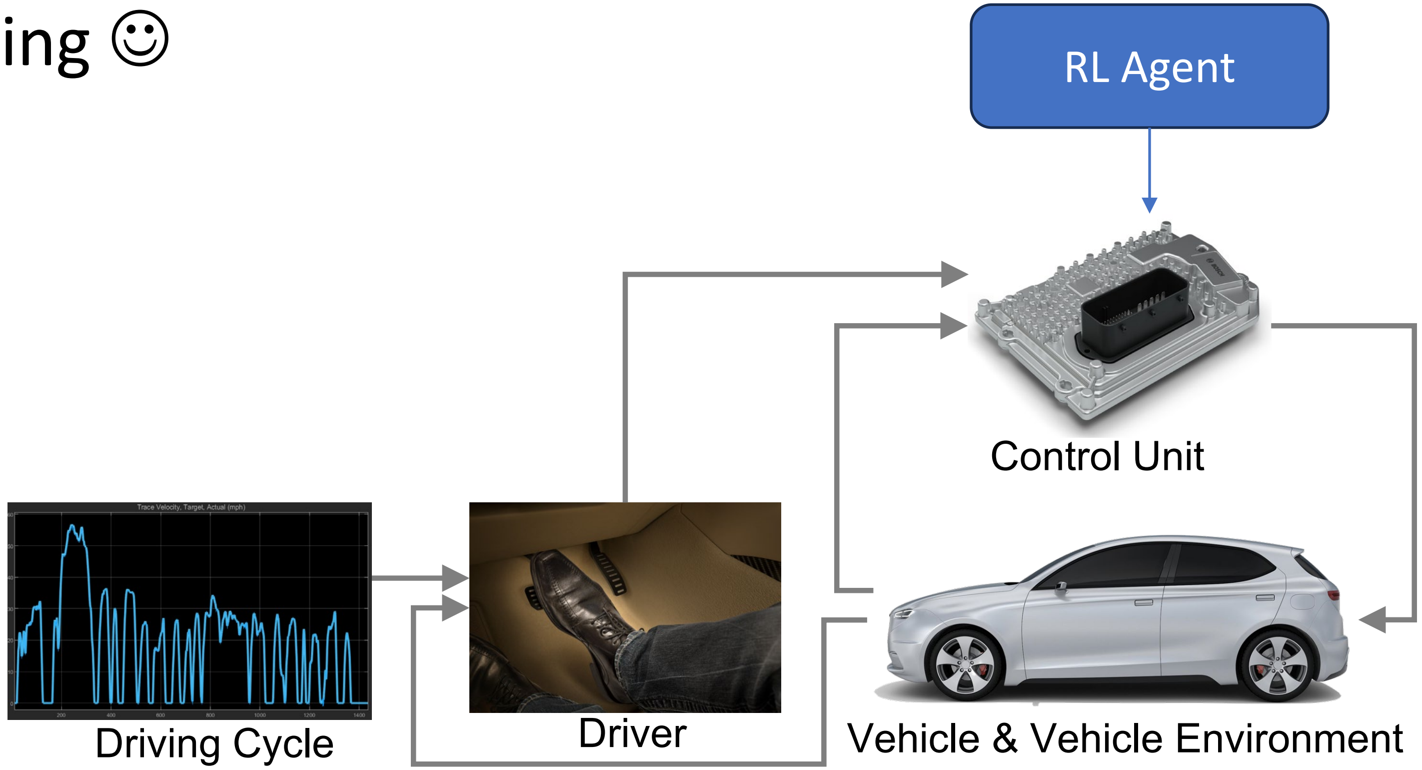
- Gear shifting logic for an automatic drive vehicle.
- Just a good, well understood use case to demonstrate feasibility.
- Start boring ☺



Explainability in Reinforcement Learning

Powertrain Control Use Case

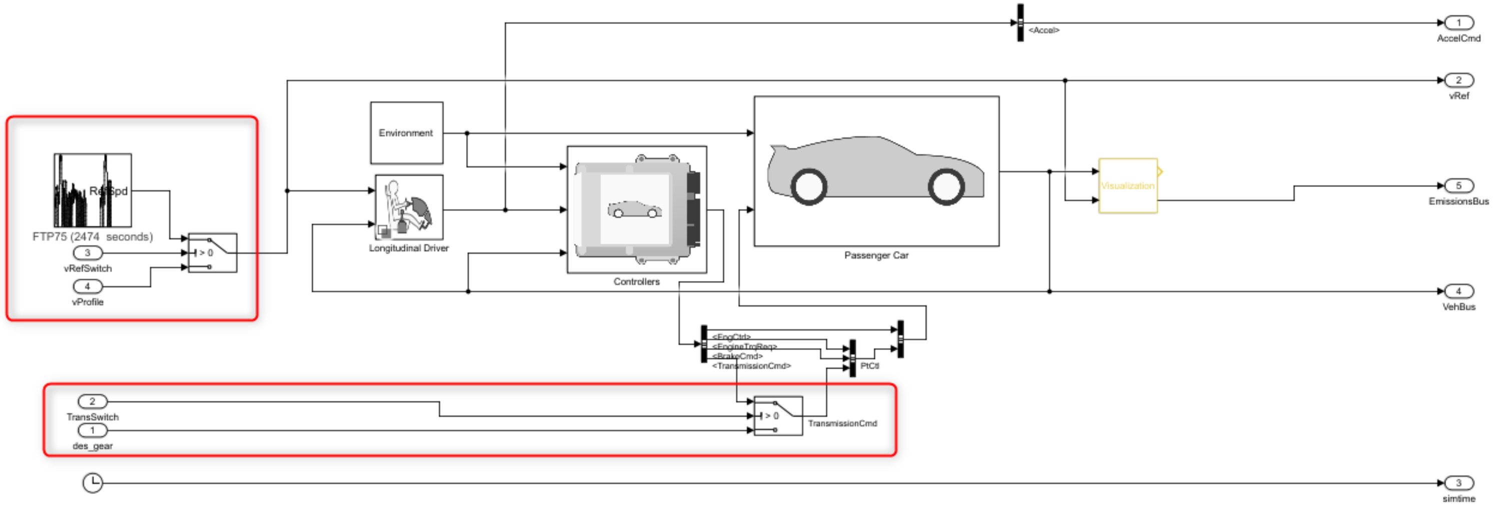
- Gear shifting logic for an automatic drive vehicle.
- Just a good, well understood use case to demonstrate feasibility.
- Start boring ☺



Explainability in Reinforcement Learning

Matlab: Conventional Vehicle Reference Application

- Full vehicle model with internal combustion engine, transmission, powertrain control algorithms.
- Used for powertrain matching analysis and component selection, control and diagnostic algorithm design, and hardware-in-the-loop testing

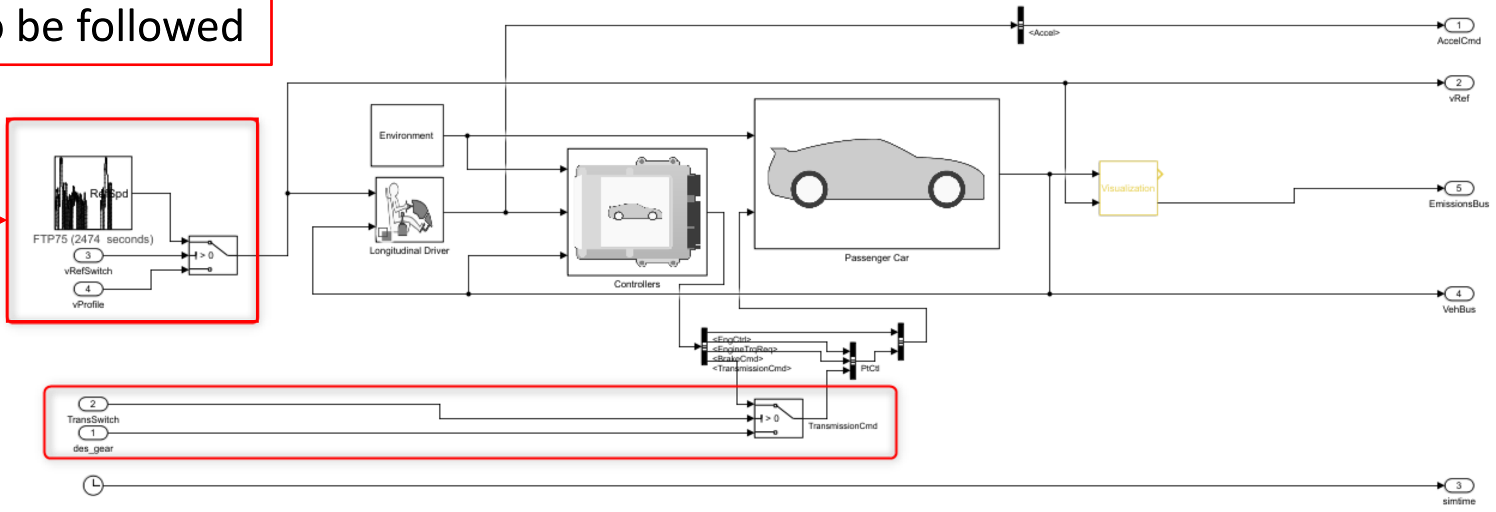


https://www.mathworks.com/help/autoblks/ug/conventional-vehicle-reference-application.html

Explainability in Reinforcement Learning

Matlab: Conventional Vehicle Reference Application

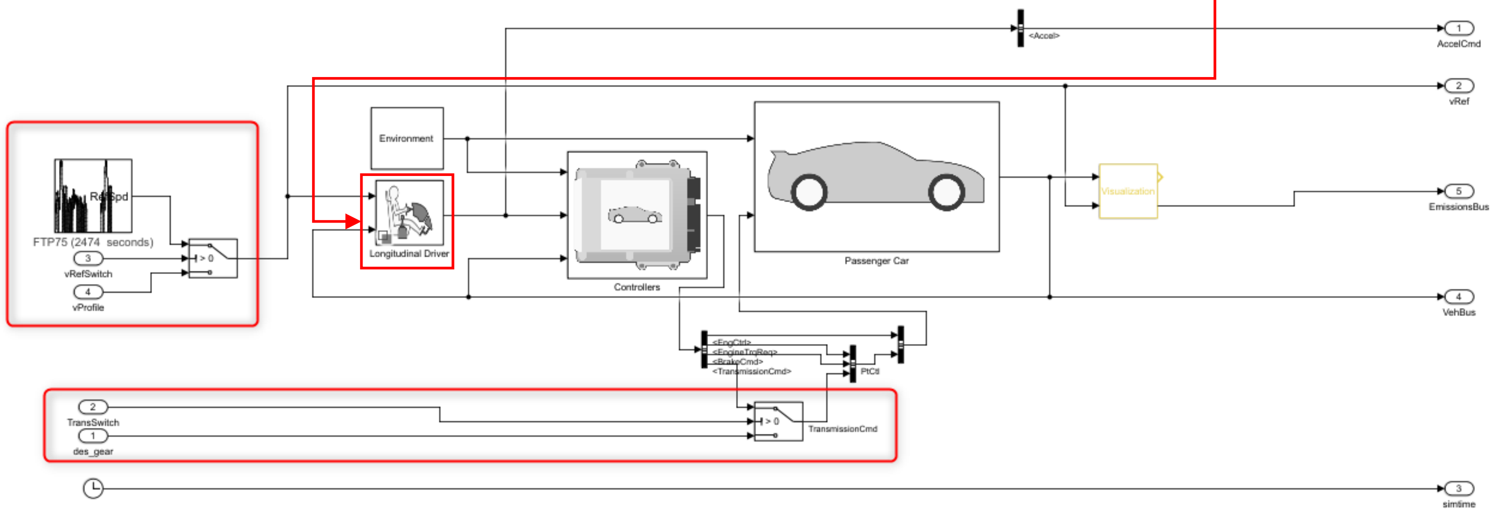
Driving Cycle:
Reference Velocity to be followed



Explainability in Reinforcement Learning

Matlab: Conventional Vehicle Reference Application

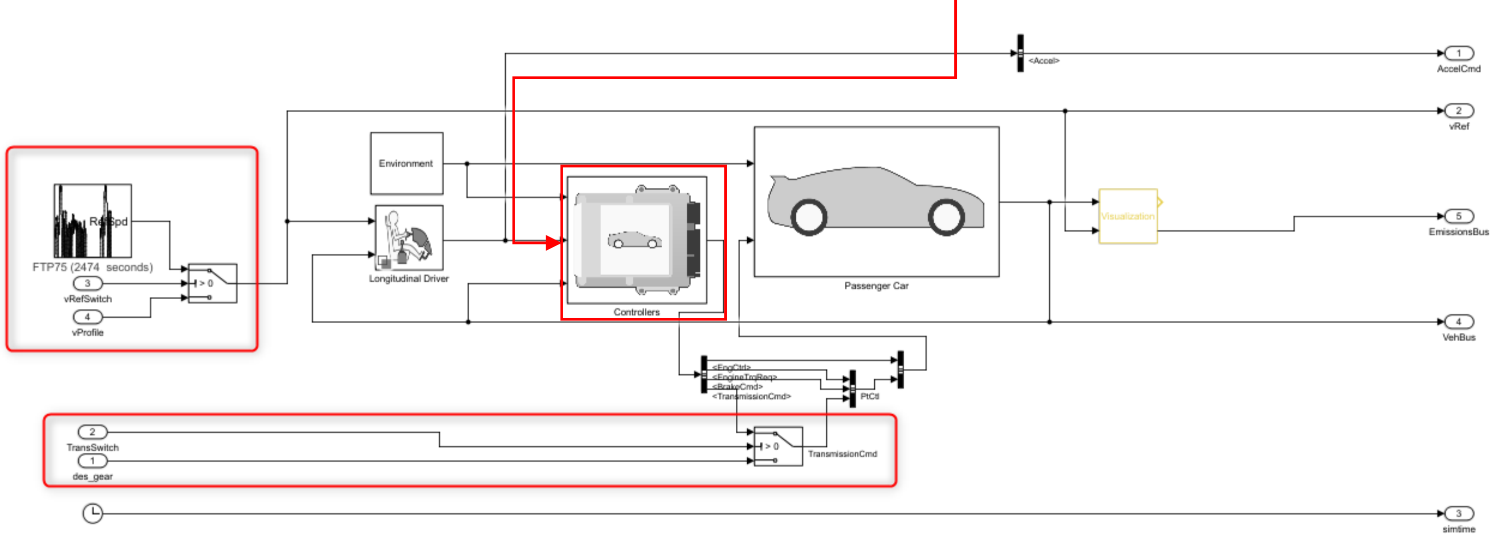
Driver:
Driver generates normalized acceleration and braking commands.



Explainability in Reinforcement Learning

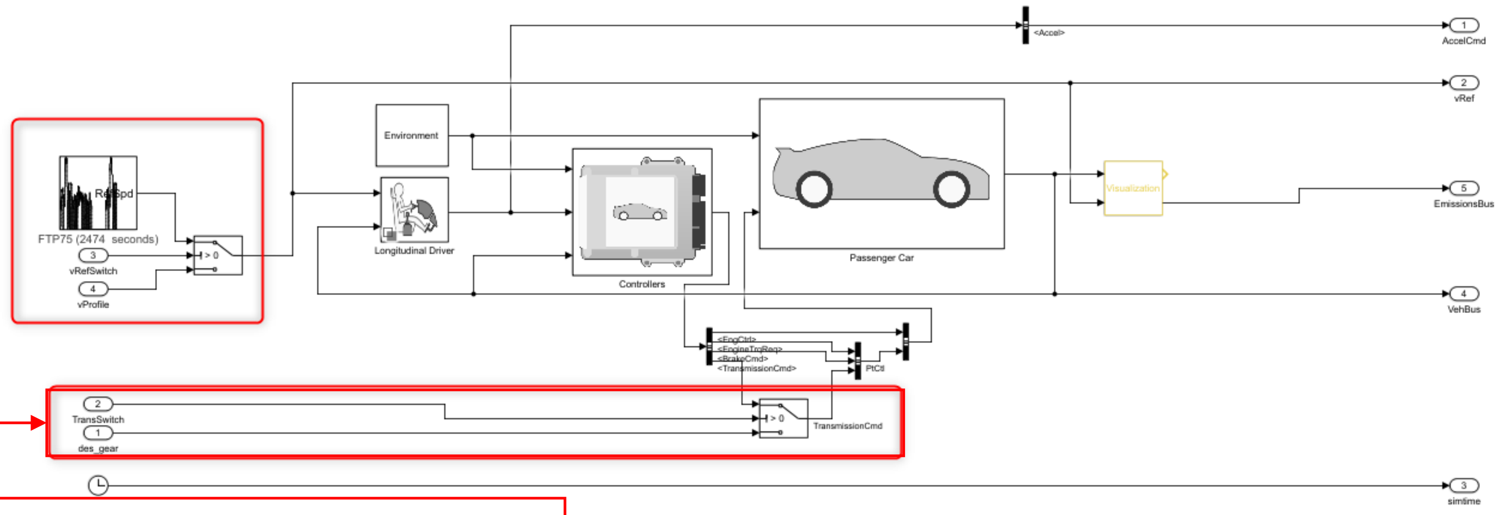
Matlab: Conventional Vehicle Reference Application

Control Unit: Implements a powertrain control module (PCM) containing a transmission control module (TCM) and engine control module (ECM)



Explainability in Reinforcement Learning

Matlab: Conventional Vehicle Reference Application

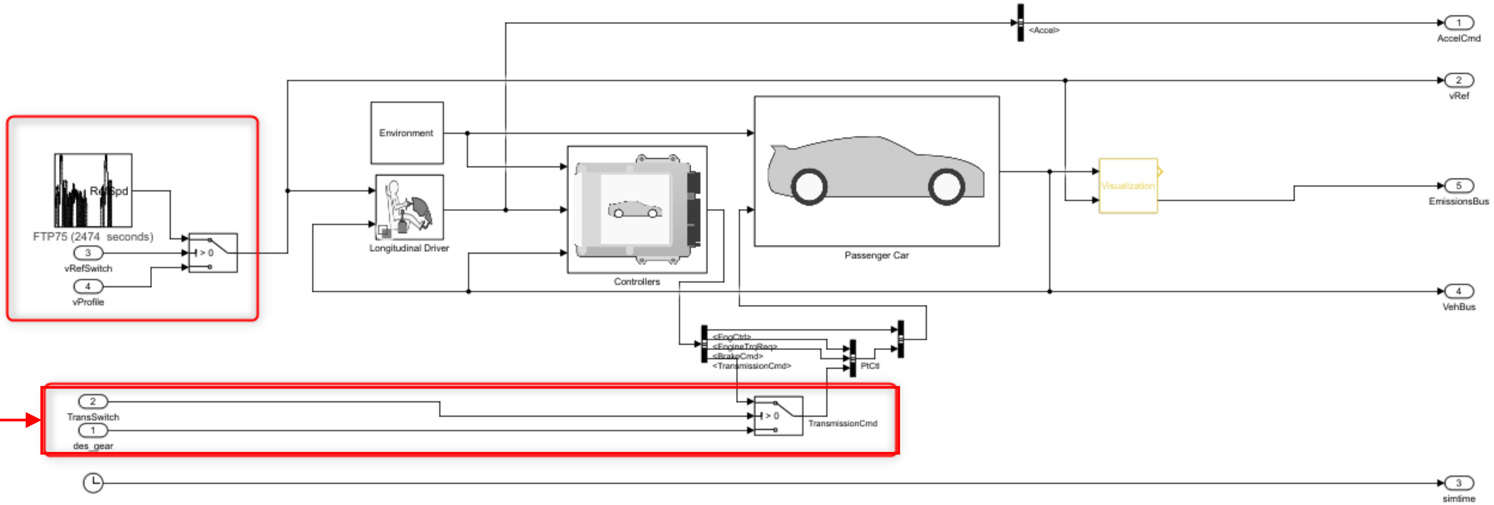


TransSwitch: External input for transmission control model. Allows for external input of gear.

Explainability in Reinforcement Learning

Goal:

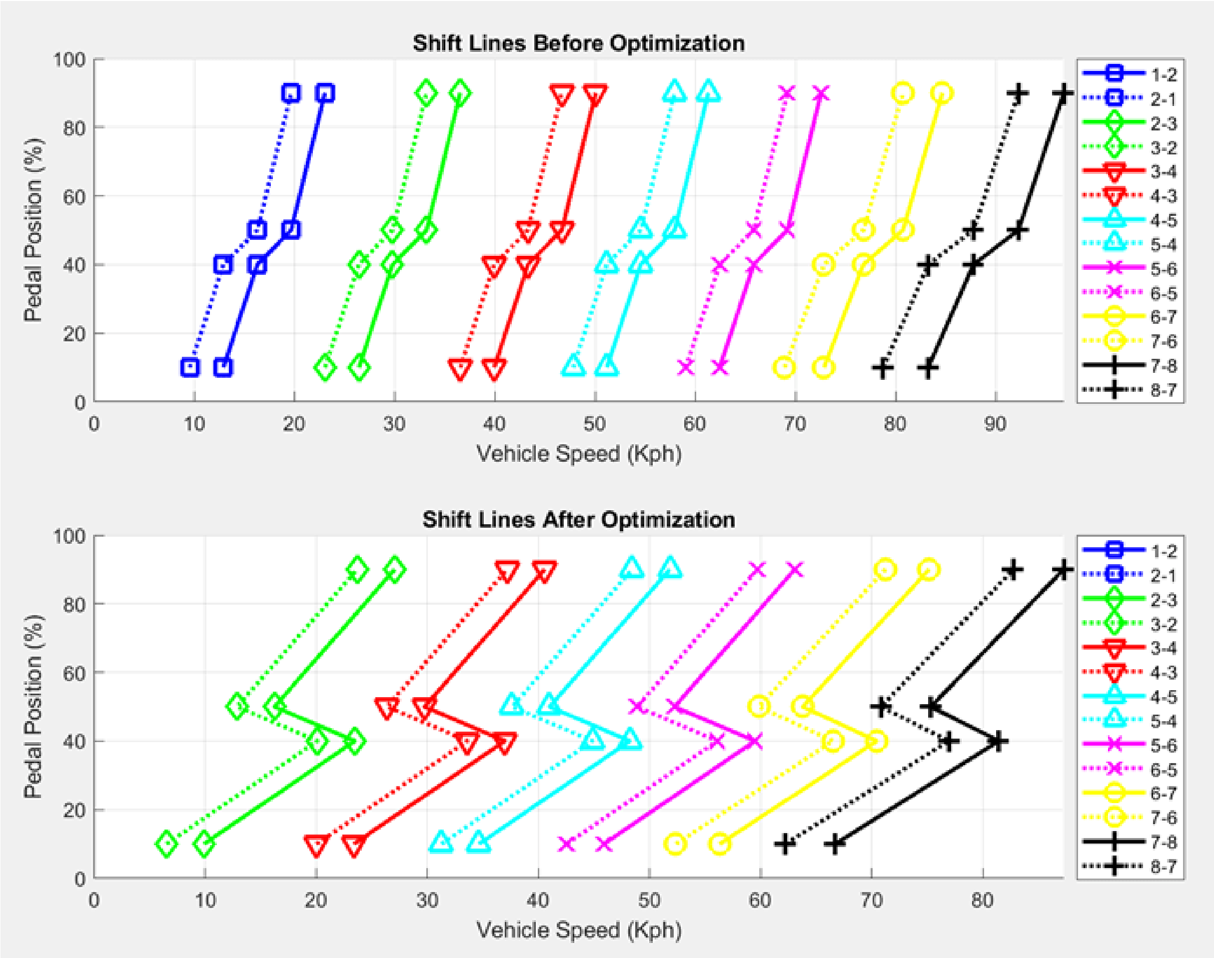
Can we replace this gear shifting logic with an explainable RL-based policy?



Explainability in Reinforcement Learning

Benchmark

- Matlab optimised controller
- Inputs:
  - Vehicle Speed
  - Pedal Position
  - Current Gear

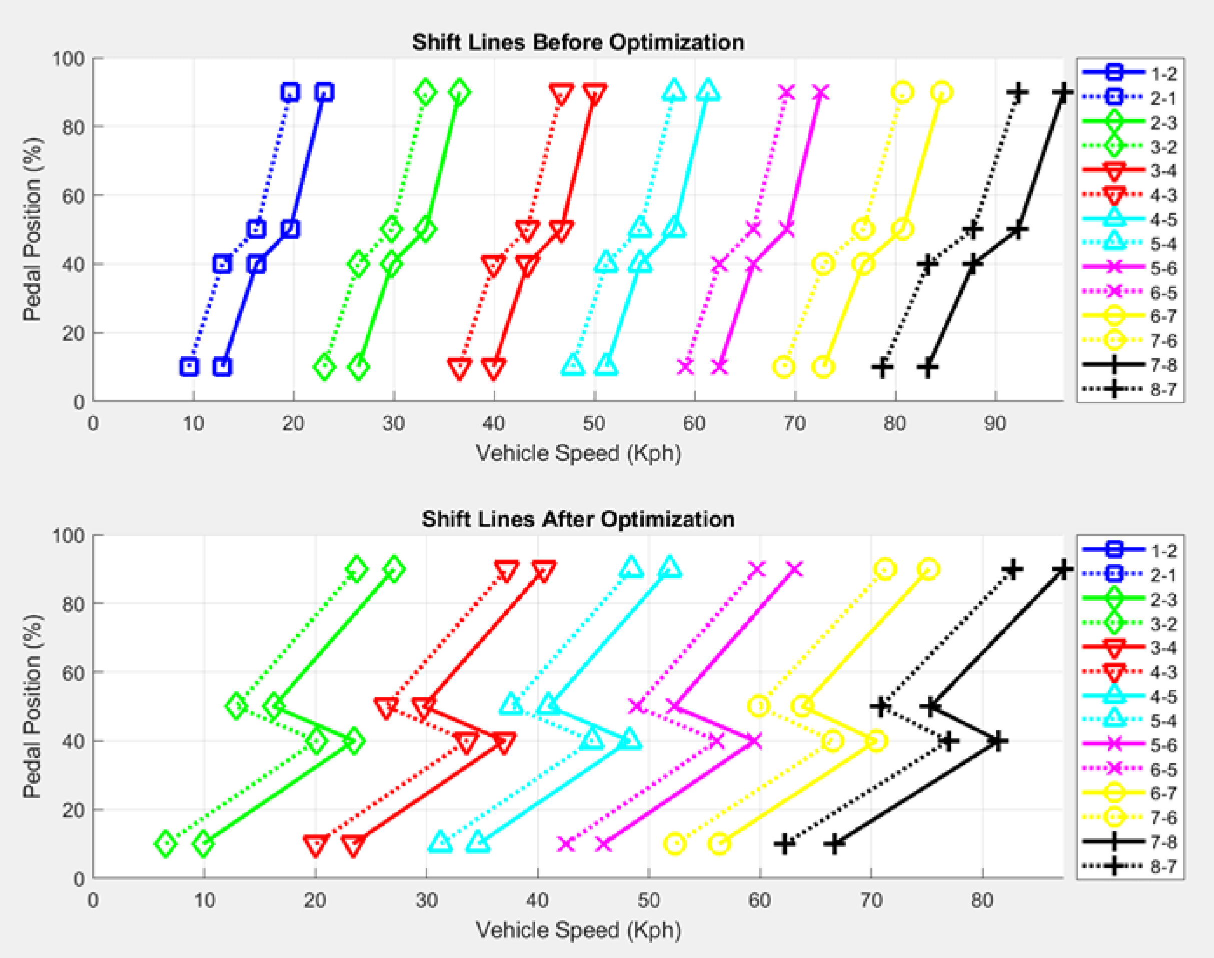


Explainability in Reinforcement Learning

Benchmark

- Matlab optimised controller
- Inputs:
  - Vehicle Speed
  - Pedal Position
  - Current Gear

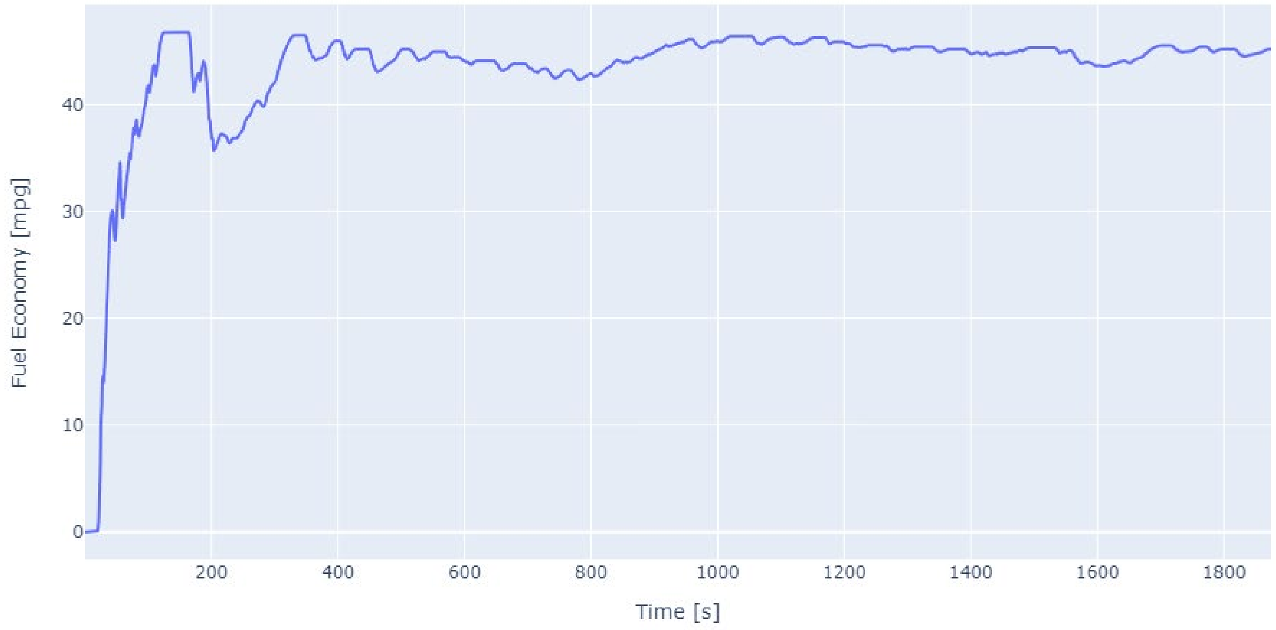
State Space



Explainability in Reinforcement Learning

Evaluation

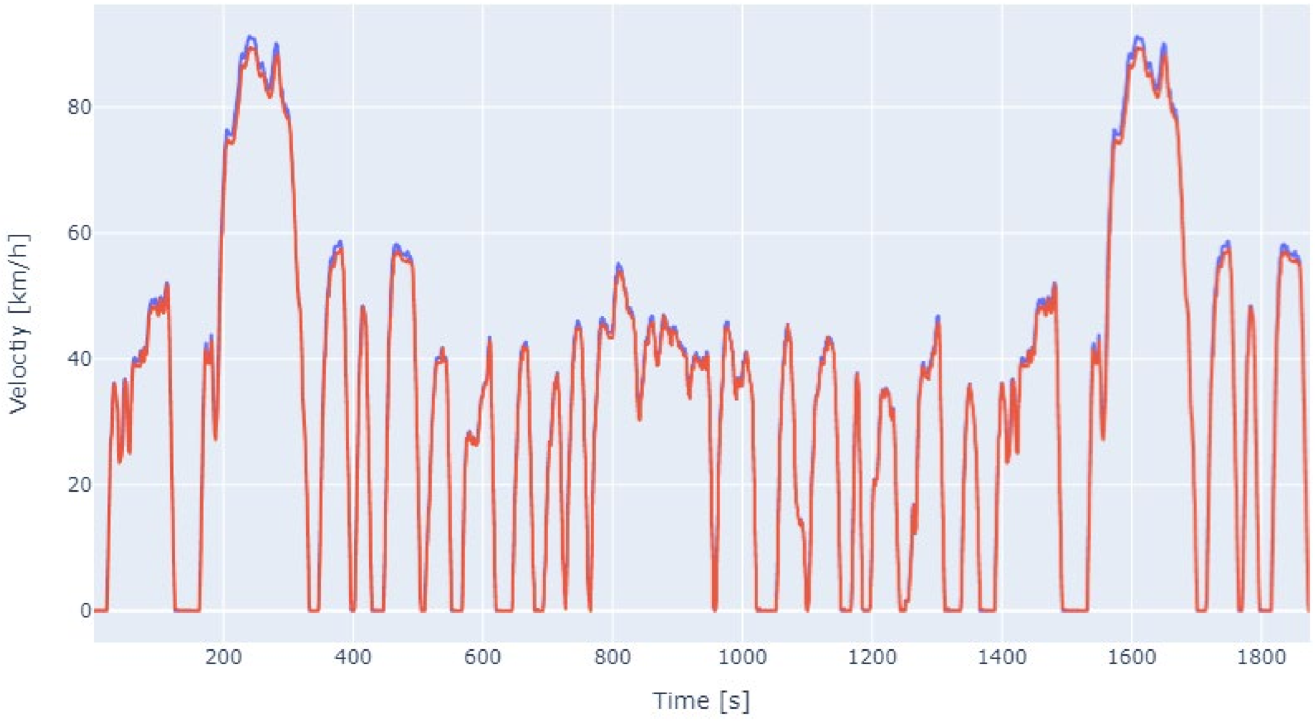
- Fuel Economy
  - Total distance / total fuel for the complete driving cycle.



Explainability in Reinforcement Learning

Evaluation

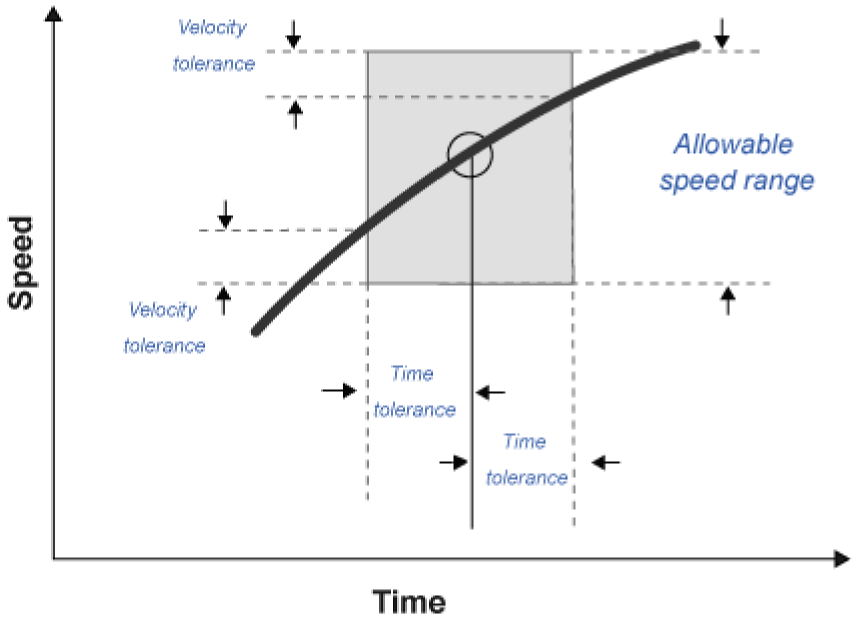
- Velocity Following



Explainability in Reinforcement Learning

Speed Faults

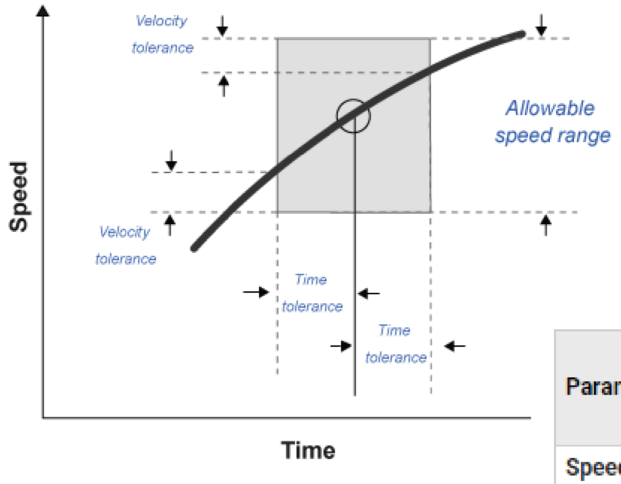
- European Union Commission. "Speed trace tolerances". *European Union Commission Regulation*. 32017R1151, Sec 1.2.6.6, June 1, 2017.



Explainability in Reinforcement Learning

Velocity Following

- European Union Commission. "Speed trace tolerances". *European Union Commission Regulation*. 32017R1151, Sec 1.2.6.6, June 1, 2017.

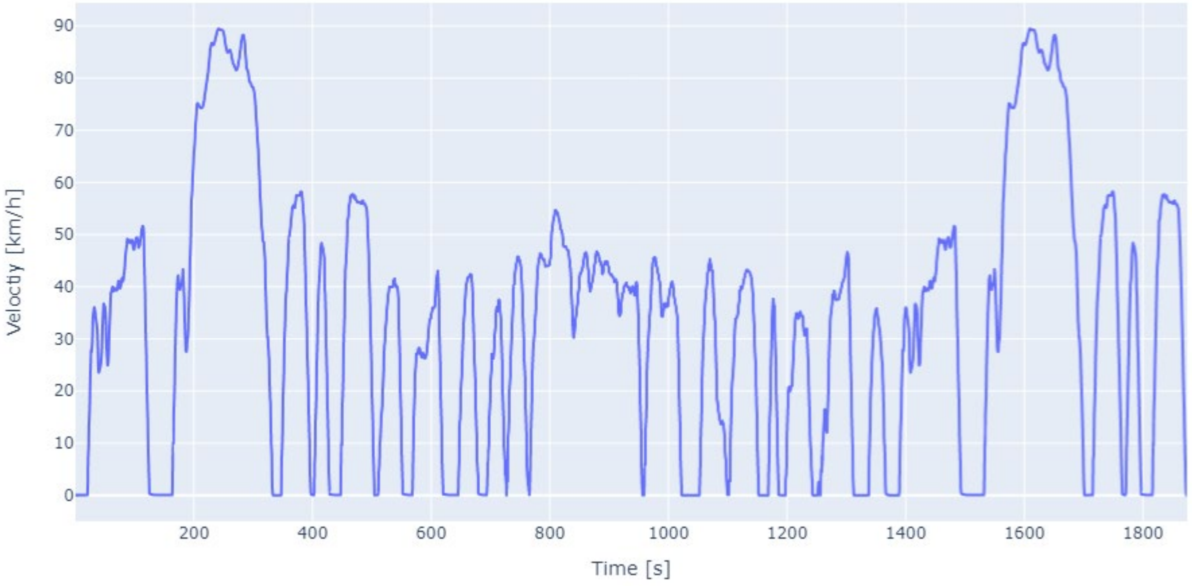


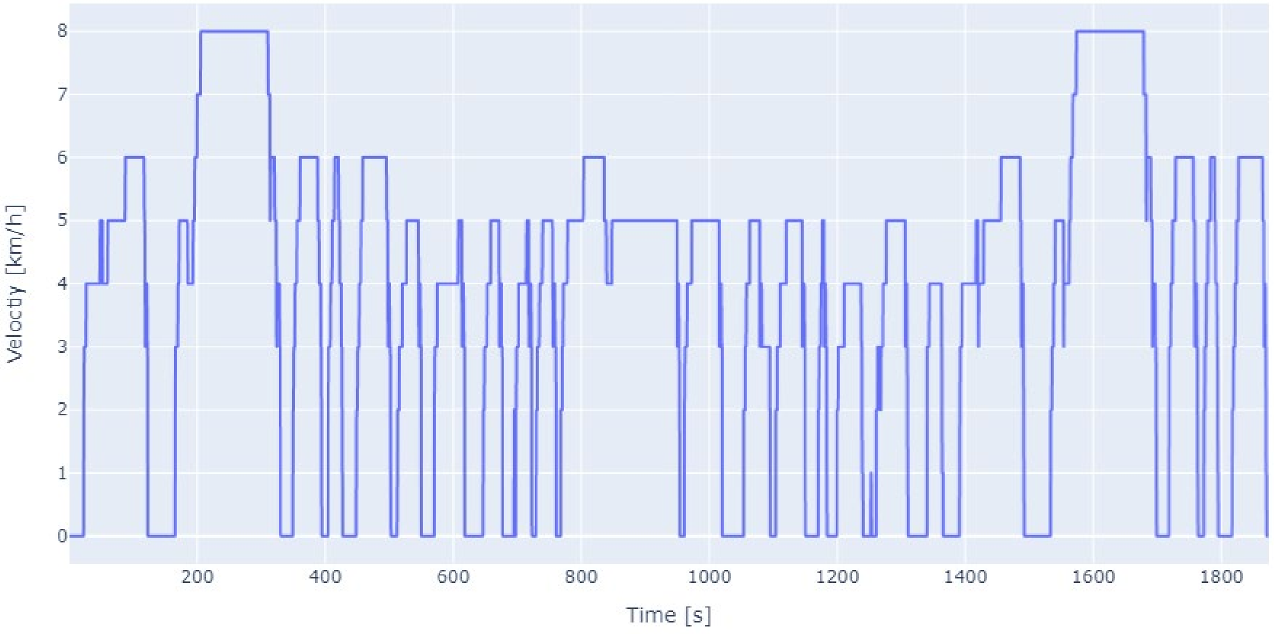
Parameter	Description	WLTP Tests[2]
Speed tolerance	Speed tolerance above the highest point and below the lowest point of the drive cycle speed trace within the time tolerance.	2.0 km/h
Time tolerance	Time that the block uses to determine the allowable speed range.	1.0 s
Maximum number of faults	Maximum number of faults allowed during the drive cycle without causing fault failure.	10
Maximum single fault time	Maximum fault duration allowed without causing fault failure.	1.0 s
Maximum total fault time	Maximum allowed accumulated time under fault condition without causing fault failure.	*Not specified*

Explainability in Reinforcement Learning

Benchmark

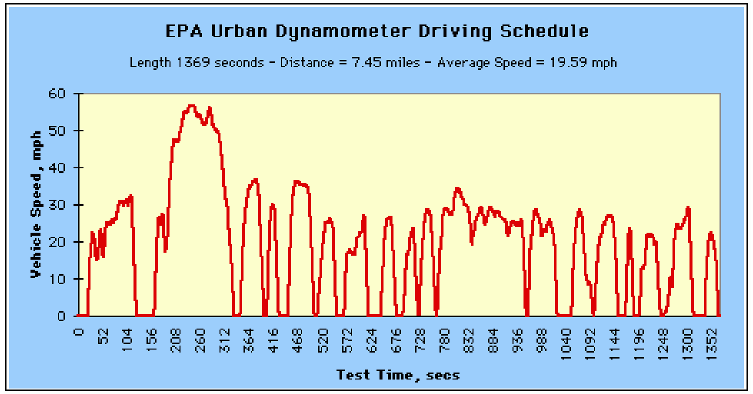
	Fuel Economy	# Faults
Matlab	38.76	8

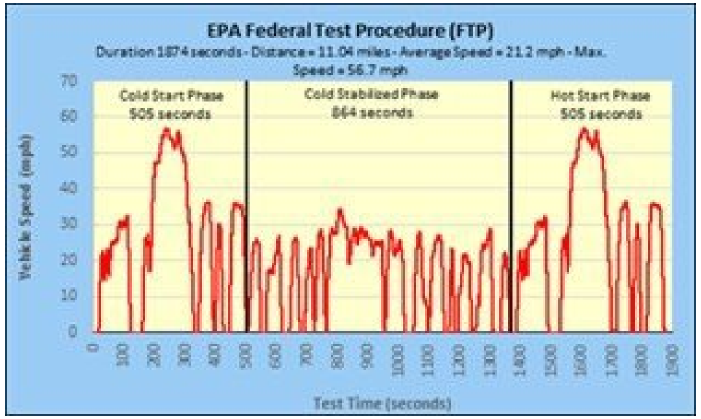




Explainability in Reinforcement Learning

Driving Cycles





Explainability in Reinforcement Learning

Gymnasium Environment

- Compile Matlab simulation into a *.dll file.
  - Can be called in Python via a step() function.
  - Includes our reward function

Explainability in Reinforcement Learning

Gymnasium Environment

- Compile Matlab simulation into a *.dll file.
  - Can be called in Python via a step() function.
  - Includes our reward function

$$R(t) = -\boxed{\alpha}|v_{\text{ref}}(t) - v_{\text{act}}(t)| - \sum_{t-\Delta t}^{t} f(\tilde{t})$$

Free parameter

Explainability in Reinforcement Learning

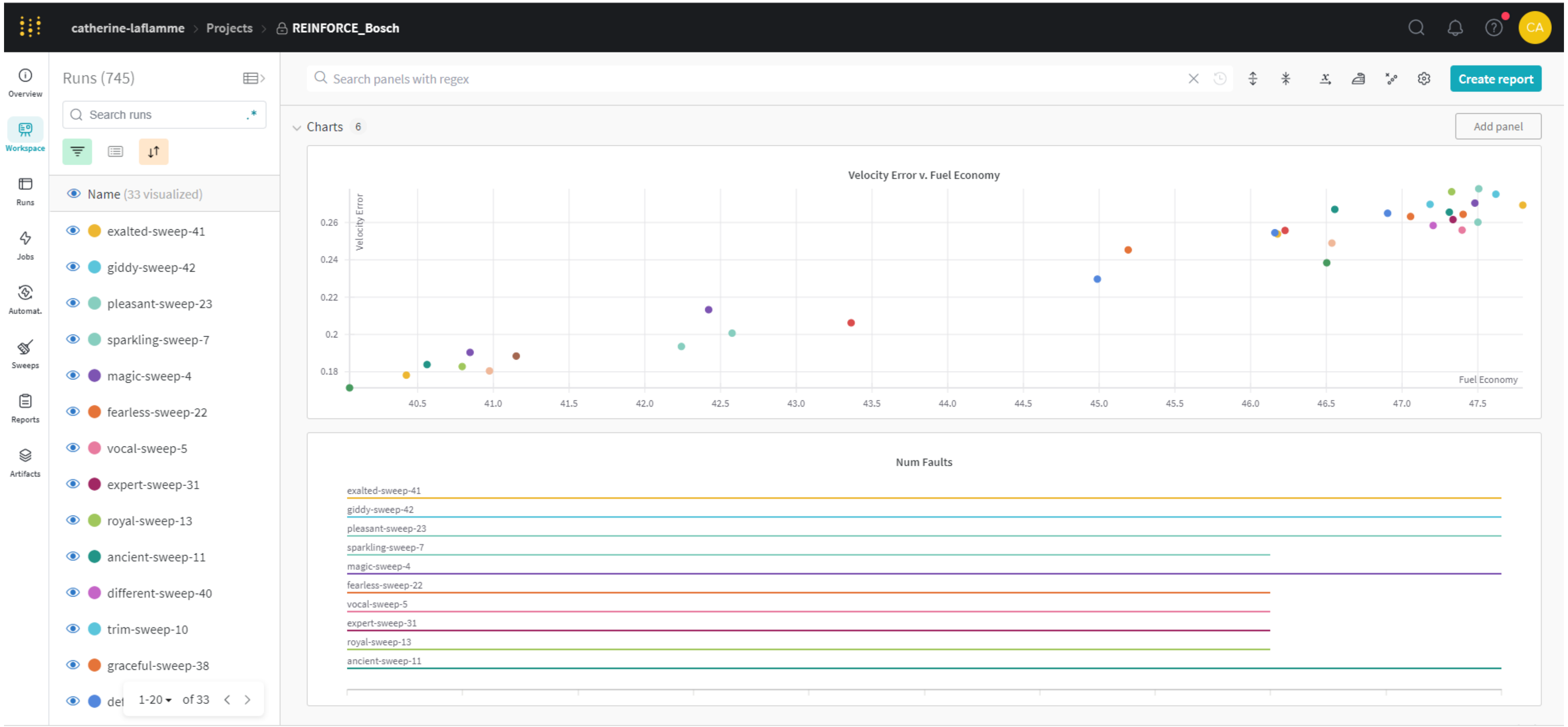
Stable Baslines implementation

- Gymnasium environments are compatible with stable baselines.
- Stable Baselines3 is "a set of reliable implementations of reinforcement learning algorithms in PyTorch."
- PPO:

```
model = PPO("MultiInputPolicy", env, verbose=1, gamma=gamma, learning_rate=learning_rate,
            n_steps=n_steps, batch_size=batch_size, stats_window_size=1, seed=seed)
```

Explainability in Reinforcement Learning

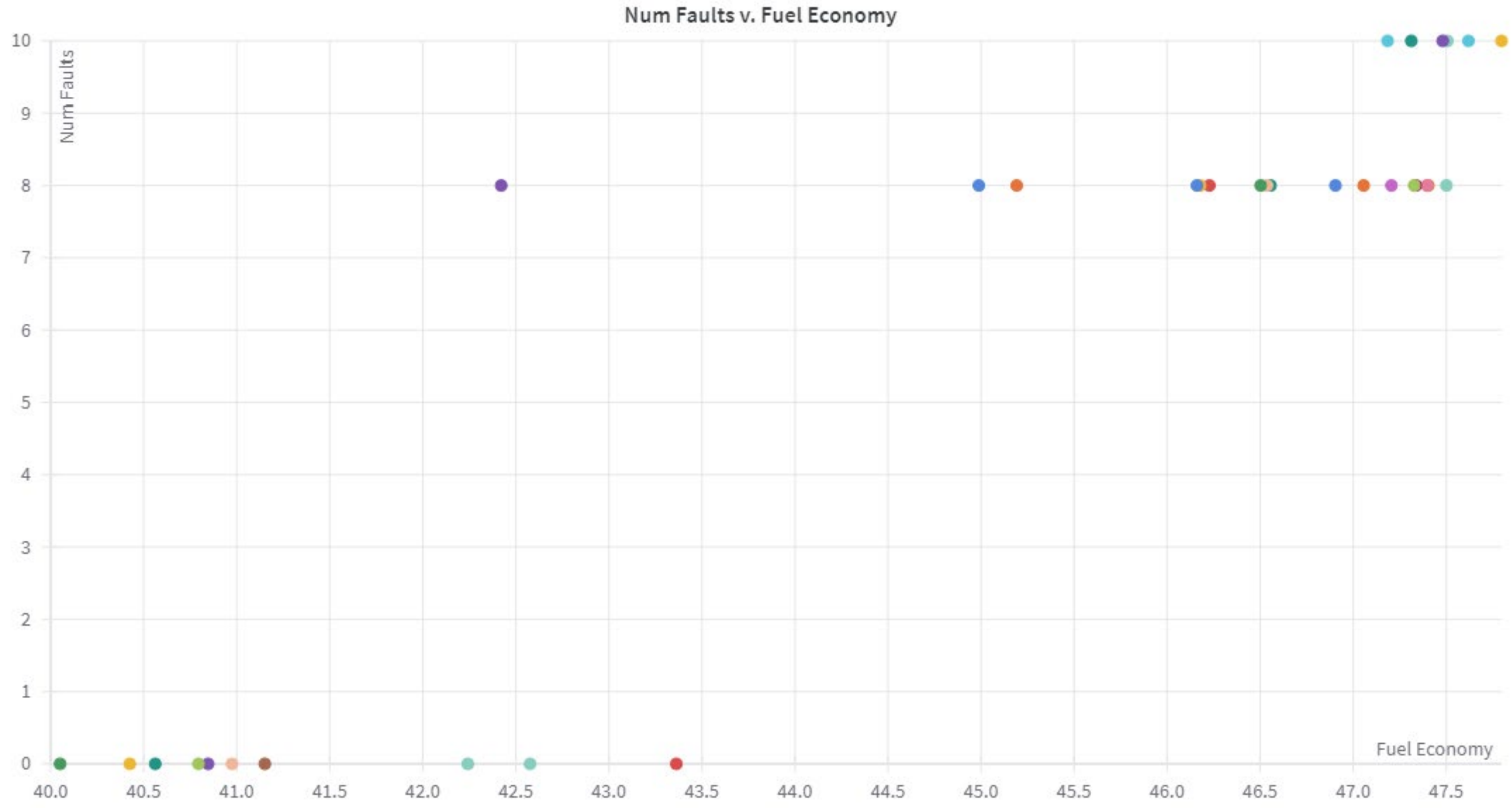
Hyperparameter Tuning with Weights and Biases



Explainability in Reinforcement Learning

Results RL Agent

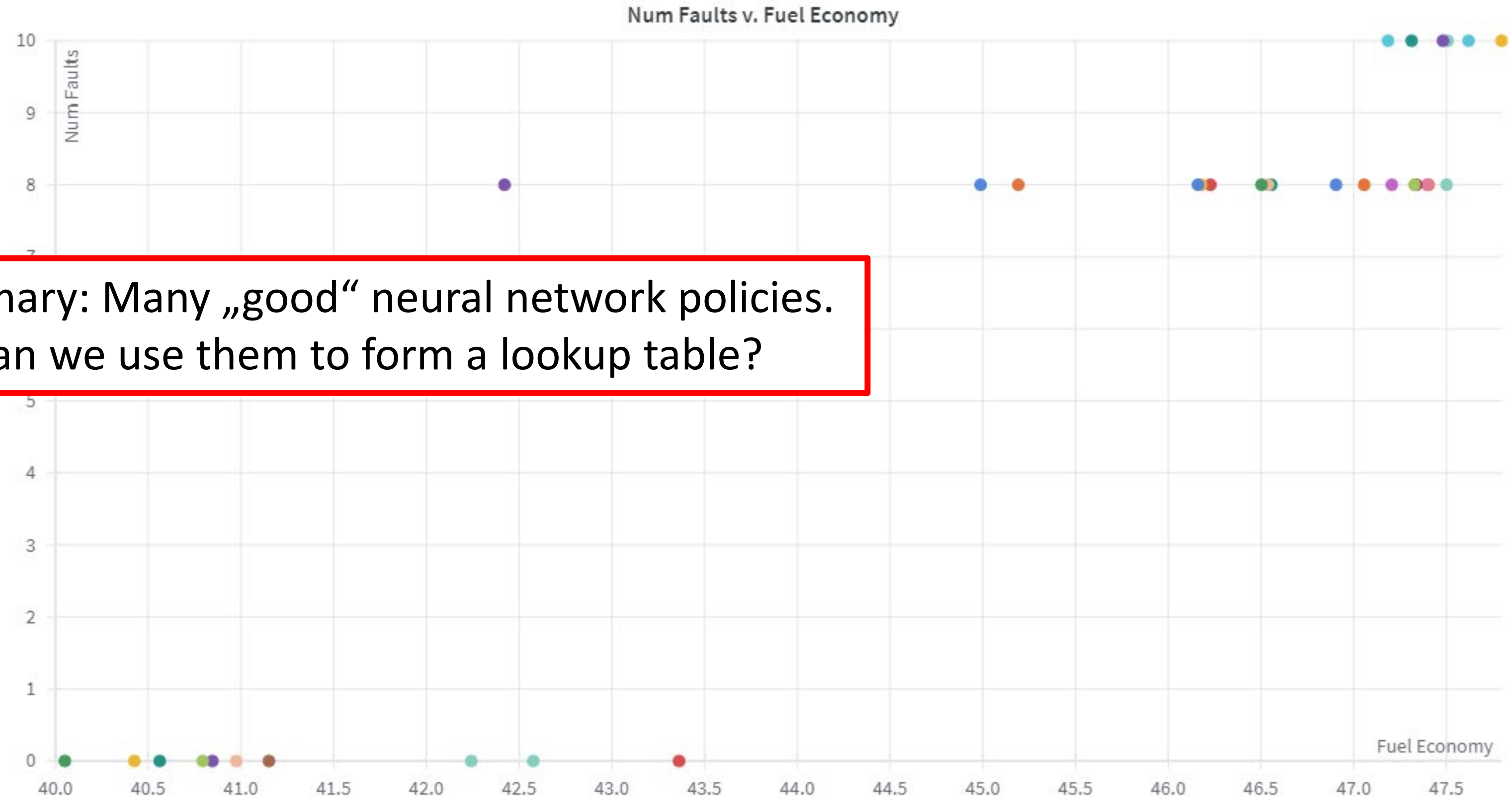
	Fuel Economy	# Faults
Matlab	38.76	8



Explainability in Reinforcement Learning

Results RL Agent

	Fuel Economy	# Faults
Matlab	38.76	8



Summary: Many „good“ neural network policies.
Can we use them to form a lookup table?

Explainability in Reinforcement Learning

Q-Table Attempt

- Note: We did attempt to train a Q-table from scratch.
  - How to discretize continuous variables?
  - Very inefficient training

- Never obtained a policy better than the benchmark.

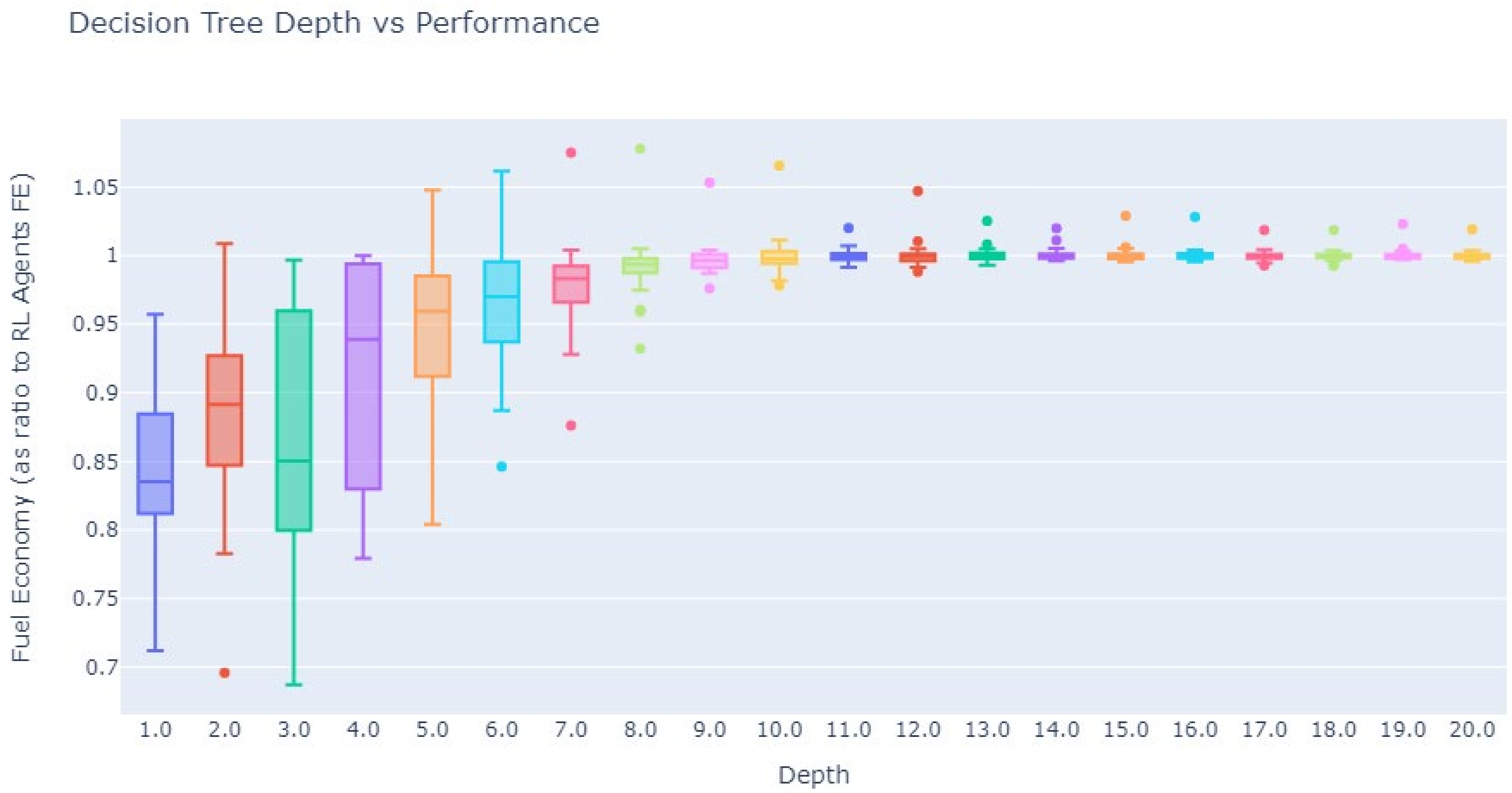
Explainability in Reinforcement Learning

Method

1. Train RL Agent
2. Sample the policy network (small state space -> can sample uniformly.)
3. Train a decision tree with standard supervised learning, fixing the depth/number of nodes of the decision tree.
4. Implement the decision tree as a lookup-table policy and test on the different driving cycles.

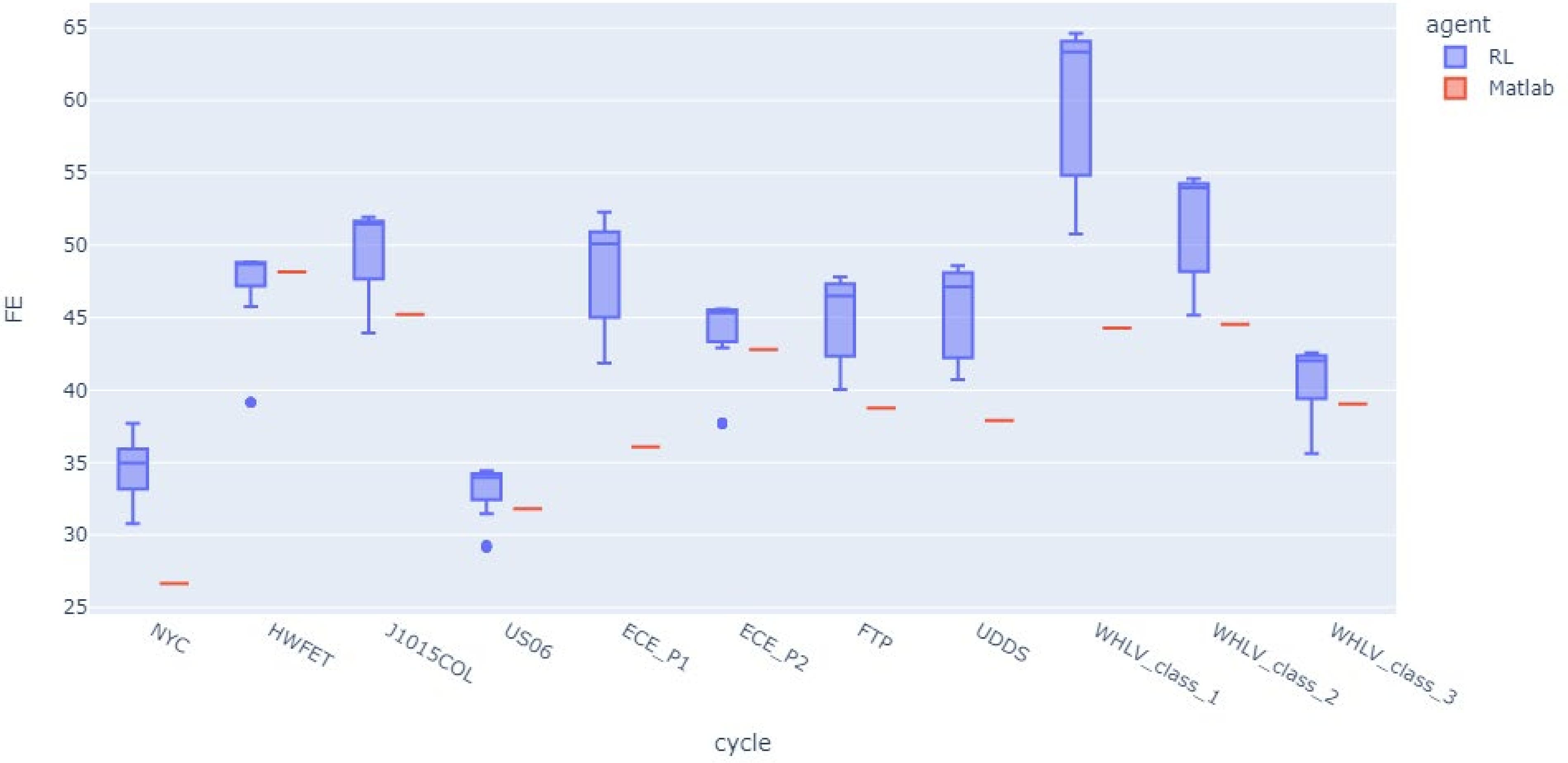
Explainability in Reinforcement Learning

Performance on Driving Cycle „FTP“



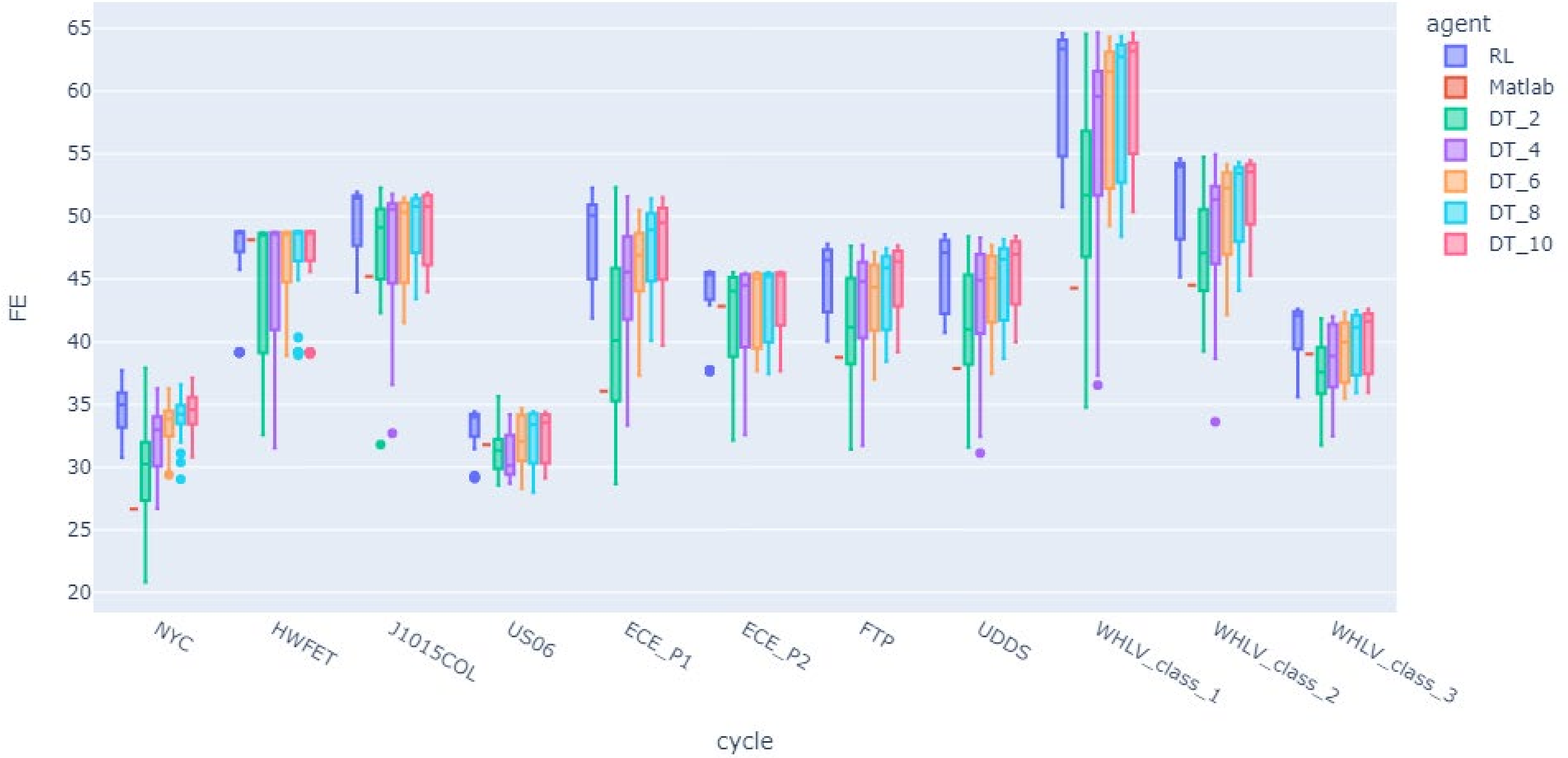
Explainability in Reinforcement Learning

Testing on Different Driving Cycles



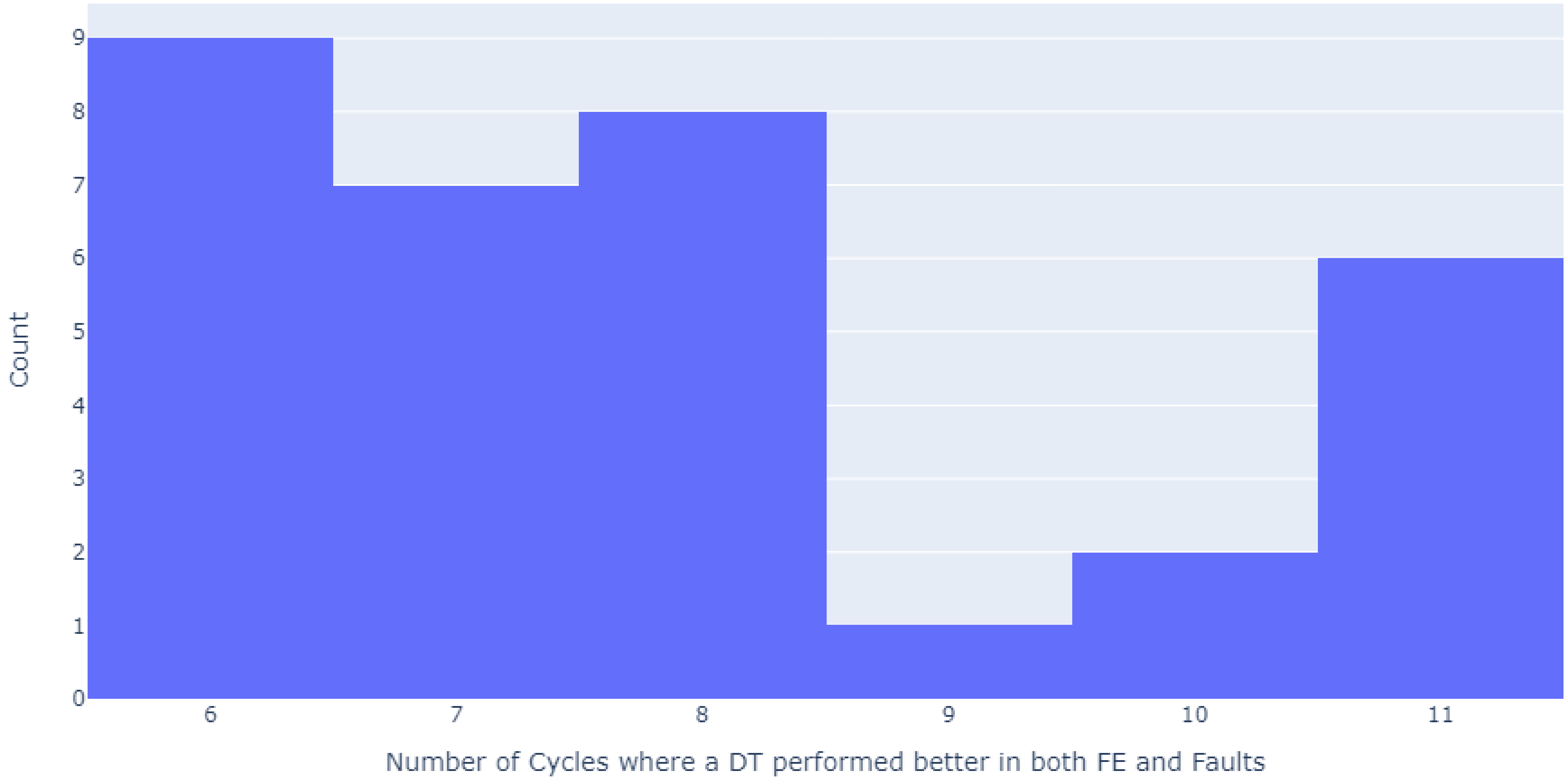
Explainability in Reinforcement Learning

Testing on Different Driving Cycles



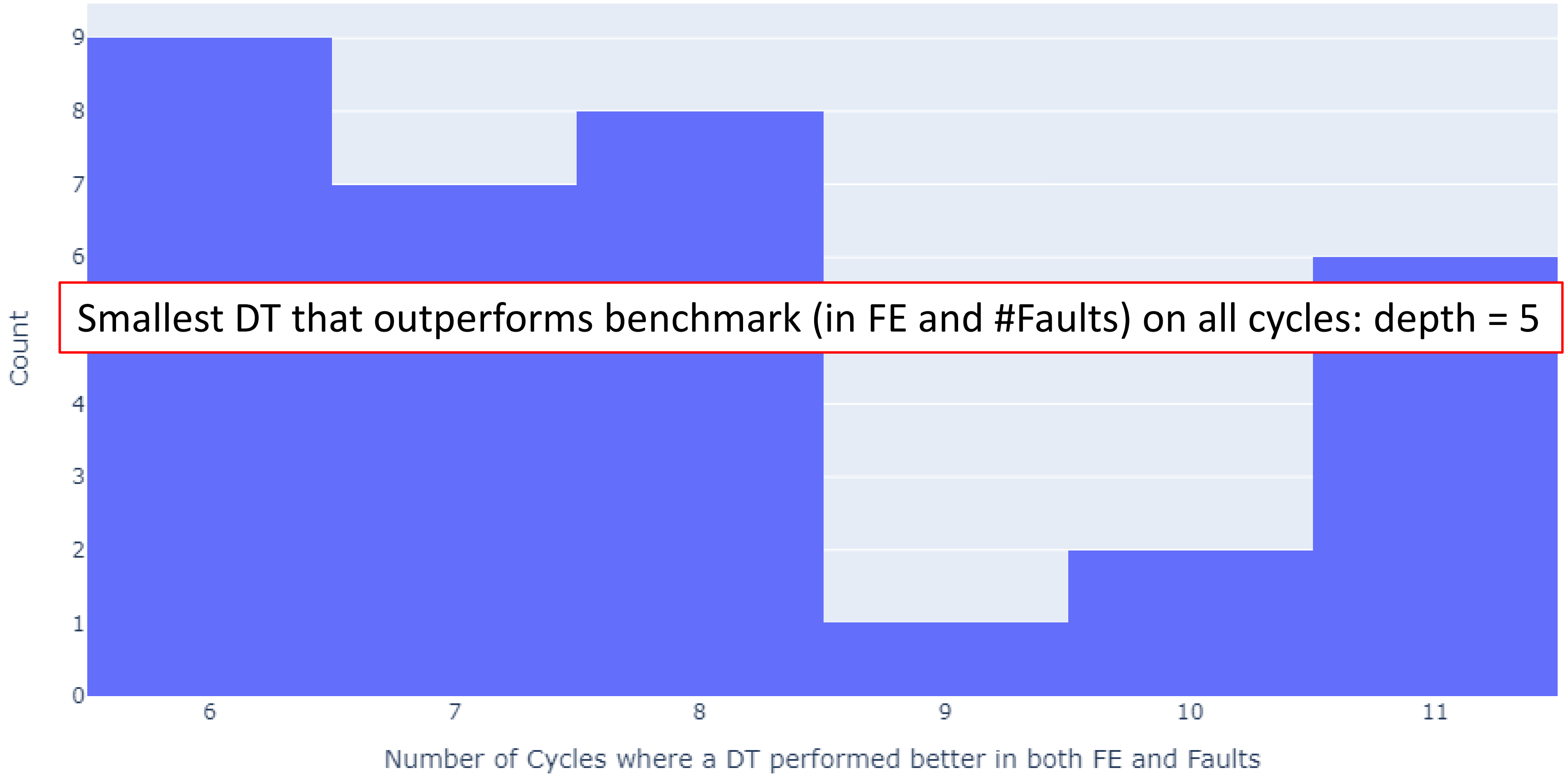
Explainability in Reinforcement Learning

Testing on Different Driving Cycles



Explainability in Reinforcement Learning

Testing on Different Driving Cycles



Explainability in Reinforcement Learning

Example Result

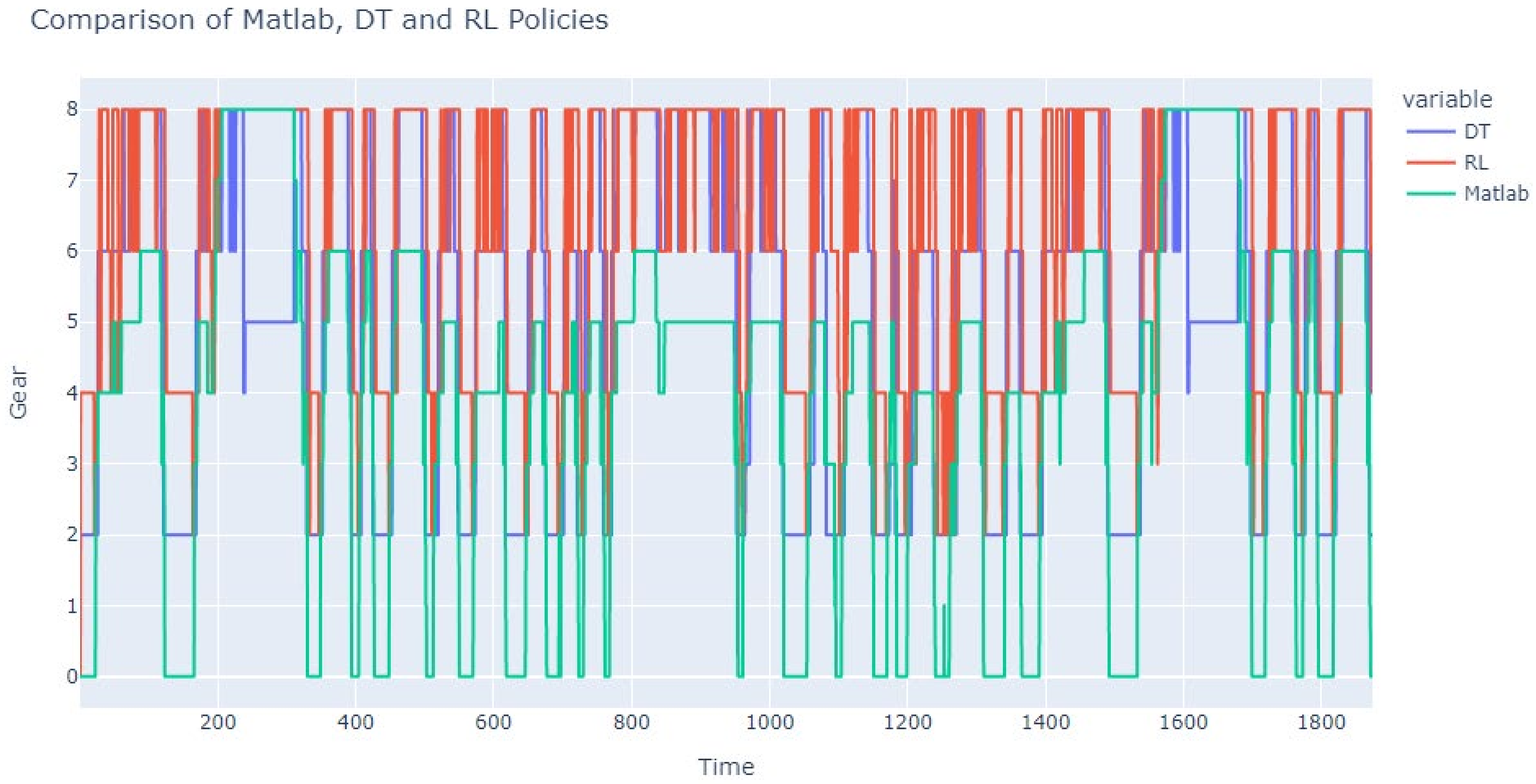
Agent	Fuel Economy	# Faults
RL	46.502527	8
DT (Depth 3)	40.878837	0
Matlab Benchmark	38.762721	8

DT (Depth 3, Nodes 15)

Vehicle Speed [km/h]	20.45	39.55		
Engine Speed [rpm]	**2090.91**	**3484.85**	**3545.45**	**3666.67**
Pedal Position [0-1]	**0.50**			

Explainability in Reinforcement Learning

Results



Explainability in Reinforcement Learning

Conclusions

- Sampling a neural network policy could be a method to create a user-friendly and explainable policy for small state spaces.
- Larger state spaces (where uniform sampling is not possible) could be possible but not studied here.
- Despite small tables, still improved performance compared to benchmarks.

Thank you!

Federal Ministry
Republic of Austria
Climate Action, Environment,
Energy, Mobility,
Innovation and Technology

REINFORCE is funded under project number 887500 by the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation and Technology (BMK) as part of the 9th call for proposals of the ICT of the Future programme.