Failure-Scenario Maker for Rule-based Agent using Multi-agent Adversarial Reinforcement Learning and its Application to Autonomous Driving

Akifumi Wachi

(IBM Research AI)

IJCAI 2019 MACAO

Autonomous driving era will arrive!

Safety-critical Systems

- Traffic accidents in real environments may lead to catastrophic and tragic results.

- To guarantee reliability of autonomous driving algorithms, we should test them in simulators before deployment.

⇩

How should we test autonomous driving algorithms before deployment?

Definition of Failure

There are several types of *failures*.

Perception [1]

Stop Sign → Speed Limit Sign

Mechanical Failure

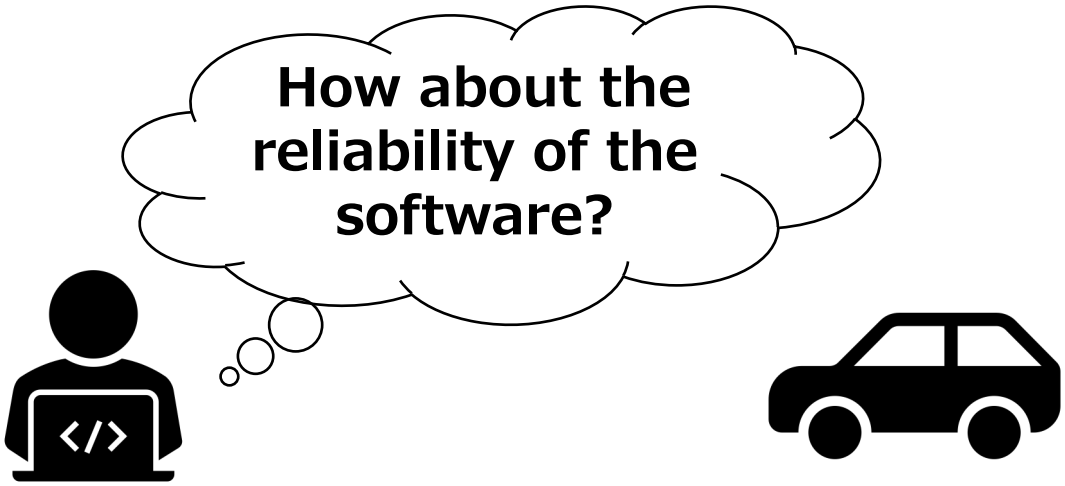
Blowout

Engine Trouble

In this work, failure means collisions with other cars or objects.

[1] Eykholt, Kevin et al. "Robust physical-world attacks on deep learning models." *arXiv preprint arXiv:1707.08945* (2017).

How Should We Test?



- Simplest way is to test (almost) all possible cases.
  → Computational cost is enormous.

⇩

- Alternatively, finding failure-scenarios is an effective and efficient approach.

Training of Astronauts

Green card

Astronaut behaves in adversarial way such that another astronaut fails.

Green Card in Astronaut Training

Trainee

Green Card

Pretend to be sick

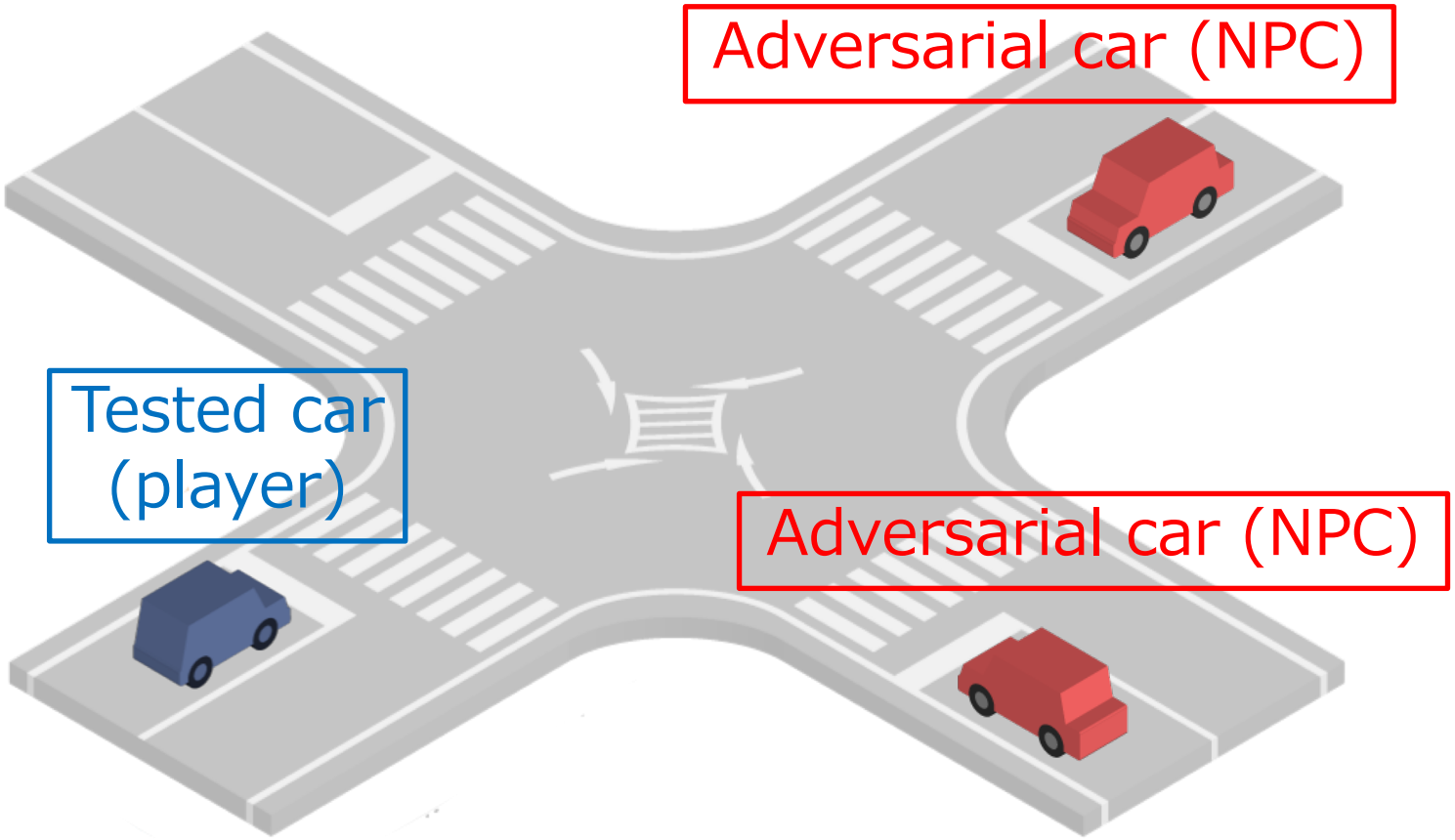
Astronaut 2 (Buzz Aldrin) is now tested.

Astronaut 1

Adversarial action

Astronaut 2

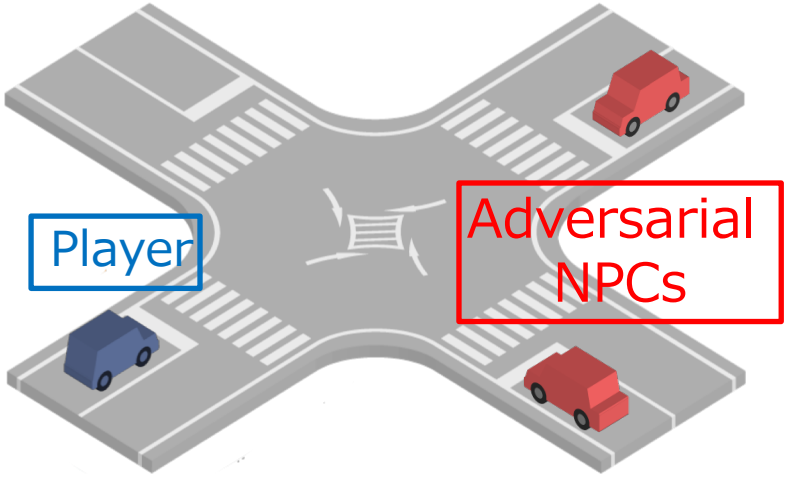
Key Idea: Adversarial Testing



Adversarial cars (non-player characters, NPCs) try to make tested car (player) fail.

Key Ideas: Adversarial RL

Adversarial cars are trained to make tested car fail using **multi-agent reinforcement learning (MARL).**

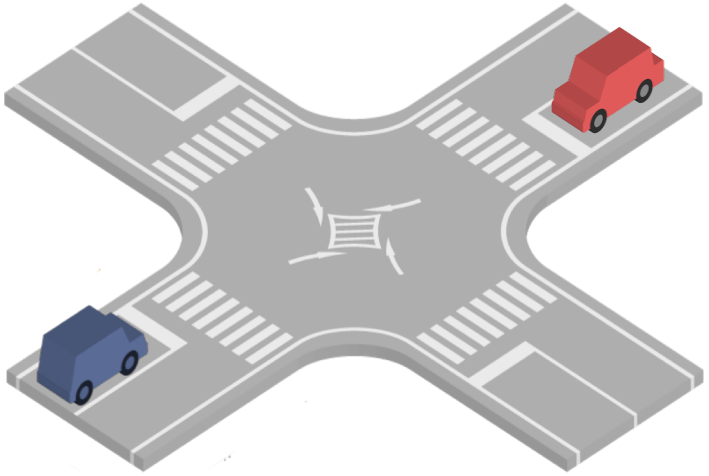


- Why RL?
  1. Humans don't have to specify details of NPCs' behaviors. ⇨ Reduction of human cost
  2. NPCs make player fail in different way from humans. ⇨ High coverage when combined with human-dependent approaches

Difficulties of Adversarial Testing

What happens if we simply train NPCs to make player fail?

⇨ NPCs try everything to attack (hit) the player.



Our ultimate goal is to improve tested algorithms.

- We need natural failure-scenarios.
- Unnatural failure scenarios are useless for improving the algorithm of the player.

Natural Failure Scenarios

To obtain natural failure-scenarios, we consider two types of reward function.

Personal reward

Reward that characterizes NPCs' own objectives

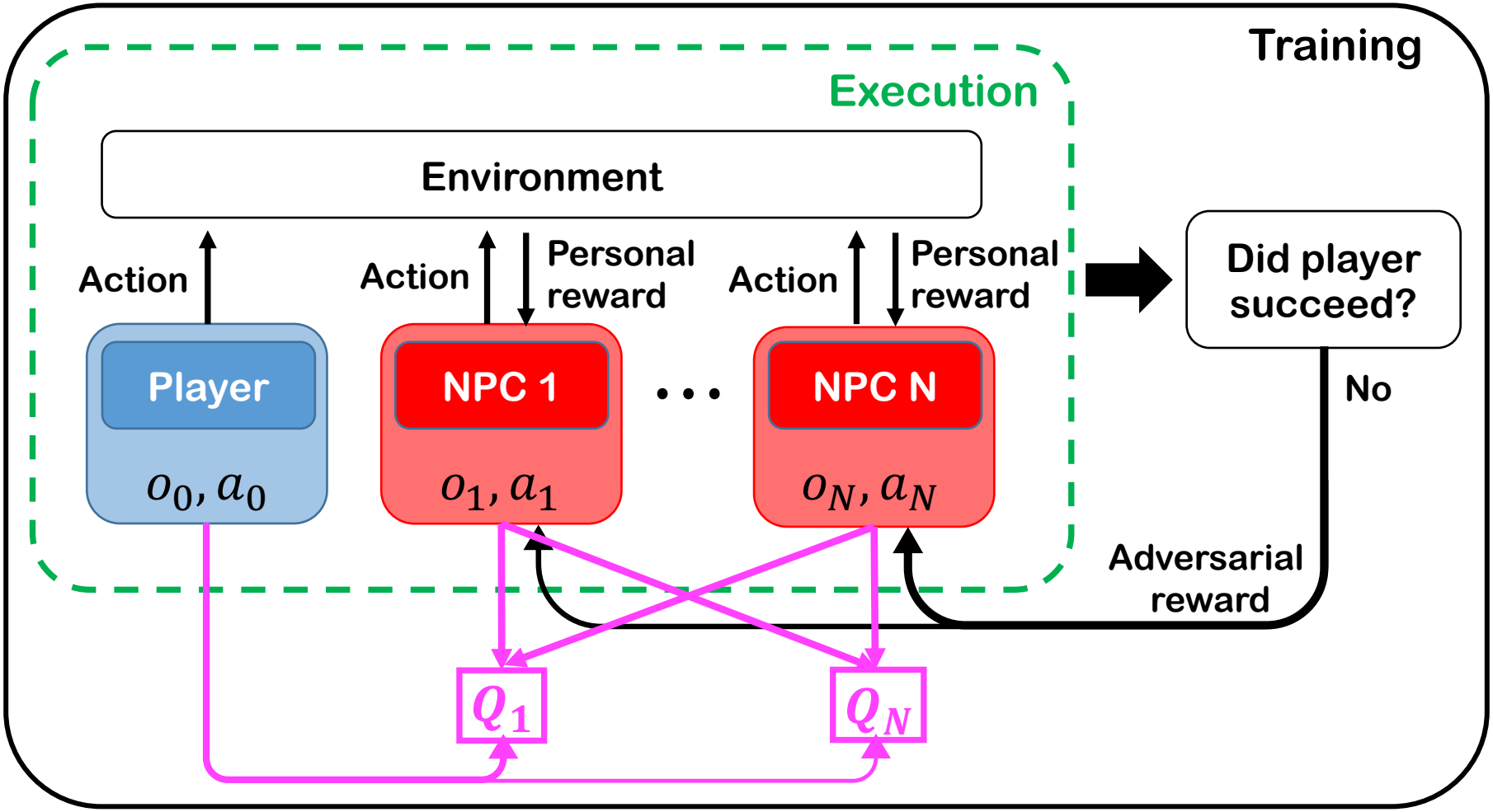
Adversarial reward

Reward that is given to NPCs when player fails

Personal reward is defined to discourage NPCs from:

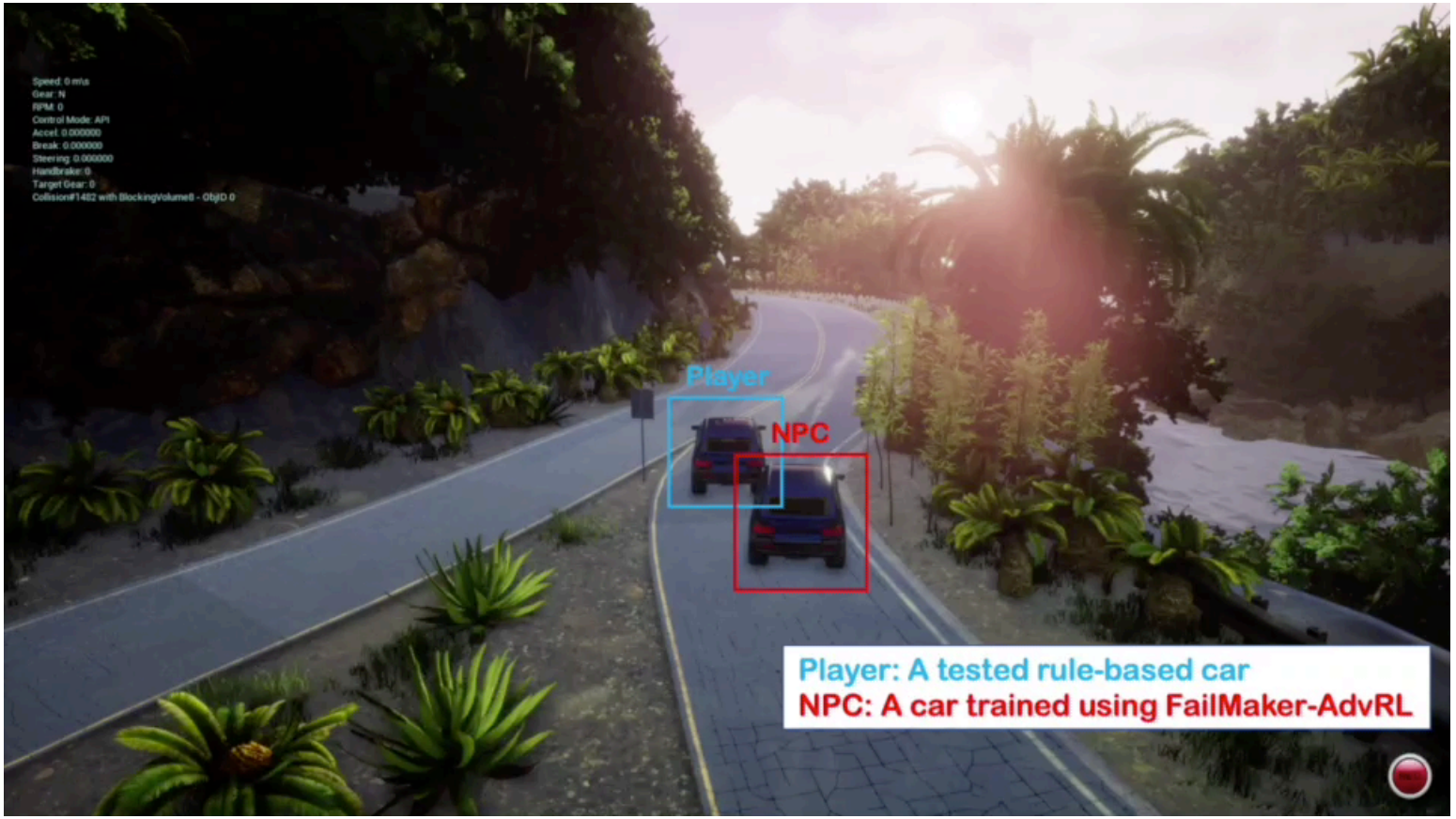
- Violating traffic rules unrealistically
- Getting damaged by hitting other cars or objects
- etc.

Overall Structure



Simulation Result (AirSim)

NPC is passing player in left lane, causing player to collide with rock.



Conclusion

1. Proposed framework for testing autonomous driving algorithms using multi-agent adversarial reinforcement learning.

2. Proposed mechanism for obtaining natural failure-scenario that is useful for improving tested algorithms.

3. Demonstrated effectiveness of our proposed method in numerical simulations.

Future Work

1. Apply our method to more sophisticated tested algorithms and more realistic environments that include pedestrians or traffic signs.

2. Create integrated adversarial situations while incorporating perception capabilities.

Thank you!