

# AWS DeepRacer-Advanced

陳昀(Ingrid)



# Advanced Example- Waypoints Template



# Lab

- Find the lab content here: [Link](#)
- Create a car
- Start training your time-trial model
- Submit your model to the Virtual Race

## Training configuration

Environment simulation  
re:Invent 2018

Reward function

Show

Sensor(s)

Camera

Action space type

Discrete

Action space

No.	Steering angle (°)	Speed (m/s)
0	-20.0	1.00
1	-20.0	2.00
2	0.0	1.00
3	0.0	2.00
4	20.0	1.00
5	20.0	2.00

Framework

Tensorflow

Reinforcement learning algorithm

PPO

Hyperparameter

Value

Gradient descent batch size

64

Entropy

0.01

Discount factor

0.999

Loss type

Huber

Learning rate

0.00003

Number of experience episodes  
between each policy-updating  
iteration

20

Number of epochs

10

# Advanced Example- Waypoints Template

## Select action space [Info](#)

### Action spaces

☐ Continuous action space  
A continuous action space allows the agent to select an action from a range of values for each state.

☒ Discrete action space  
A discrete action space represents all of the agent's possible actions for each state in a finite set.

## Define discrete action space [Info](#)

### Steering angle

The steering angle determines to what degree the front wheels of your agent can turn. Steering angle granularity describes the increments between angles. Choose higher numbers for smoother actions. Higher numbers also expand the action space and thus increase training time.

Pro tip: Too high a steering angle can empower the agent to make unnecessarily excessive turns and can cause zig-zagging.

Steering angle granularity

Maximum steering angle

 degrees

Max values are between 1 and 30.

### Speed

The speed determines how fast your agent can drive. For the agent to be able to drive faster, set a higher speed. On a given track, you must balance the desire for speed against the concern for keeping the agent on the track while it maneuvers curves at a high speed.

Pro tip: The higher the speed limit and more actions, the vehicle has a better chance of driving faster, but the model may take longer to converge.

Speed granularity

Maximum speed

 m/s

Select values between 0.1 and 4.

# Advanced Example- Waypoints Template

```
1  def reward_function(params):
2
3      center_variance = params["distance_from_center"] / params["track_width"]
4      #racing line
5      left_lane = []#Fill in the waypoints
6
7      center_lane = []#Fill in the waypoints
8
9      right_lane = []#Fill in the waypoints
10
11     #Speed
12     fast = []#Fill in the waypoints, 2m/s
13     slow = []#Fill in the waypoints, 1m/s
14
15     reward = 21
16
```

# Advanced Example- Waypoints Template

```
if params["all_wheels_on_track"]:
    reward += 10
else:
    reward -= 10

if params["closest_waypoints"][1] in left_lane and params["is_left_of_center"]:
    reward += 10
elif params["closest_waypoints"][1] in right_lane and not params["is_left_of_center"]:
    reward += 10
elif params["closest_waypoints"][1] in center_lane and center_variance < 0.4:
    reward += 10
else:
    reward -= 10

if params["closest_waypoints"][1] in fast:
    if params["speed"] == 2 :
        reward += 10
    else:
        reward -= 10
elif params["closest_waypoints"][1] in slow:
    if params["speed"] == 1 :
        reward += 10
    else:
        reward -= 10

return float(reward)
```

# Advanced Example- Waypoints Template

## waypoints

**Type:** list of [float, float]

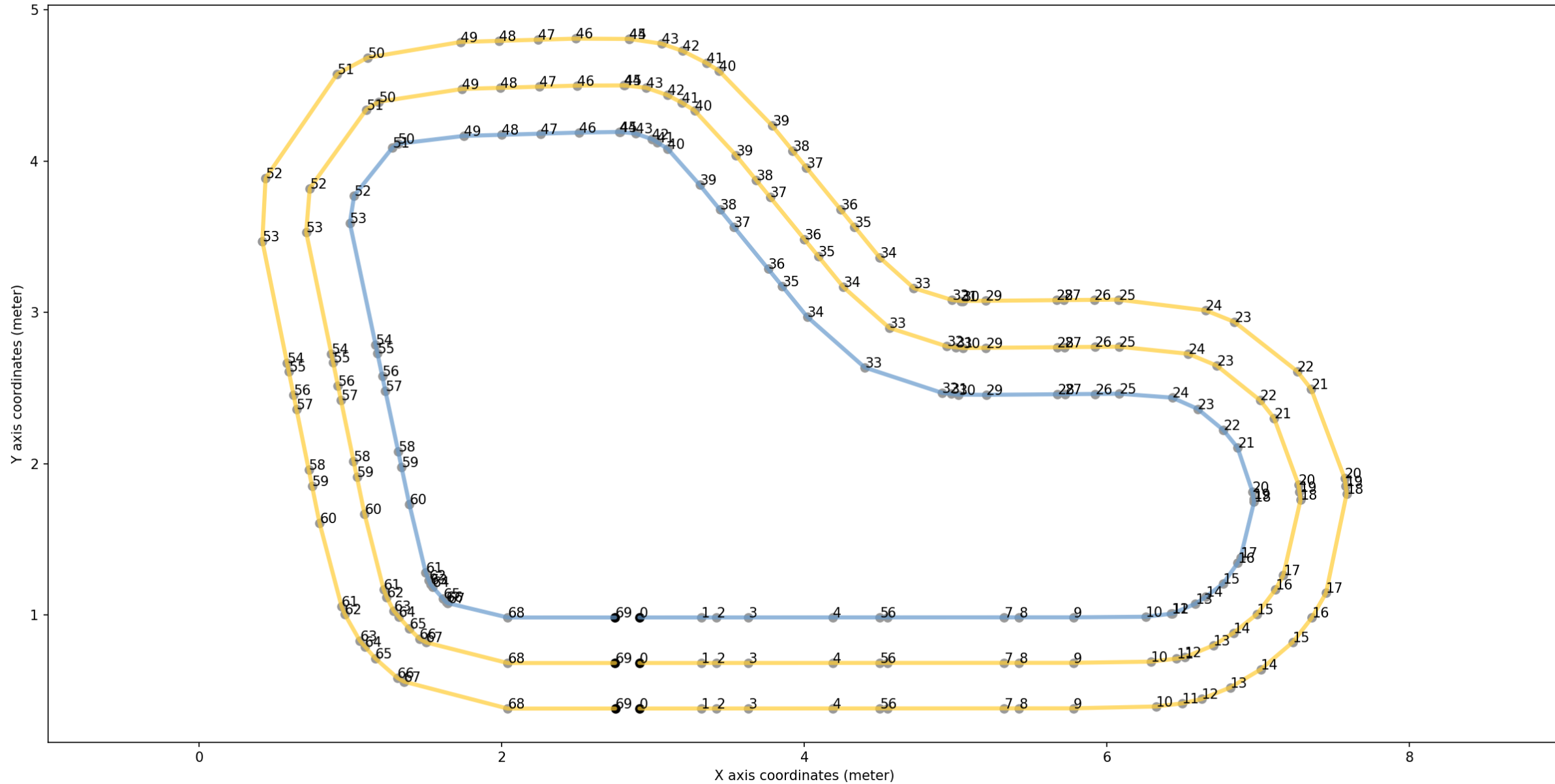
**Range:**  $[[x_{w,0}, y_{w,0}] \dots [x_{w,Max-1}, y_{w,Max-1}]]$

An ordered list of track-dependent  $Max$  milestones along the track center. Each milestone is described by a coordinate of  $(x_{w,i}, y_{w,i})$ . For a looped track, the first and last waypoints are the same. For a straight or other non-looped track, the first and last waypoints are different.

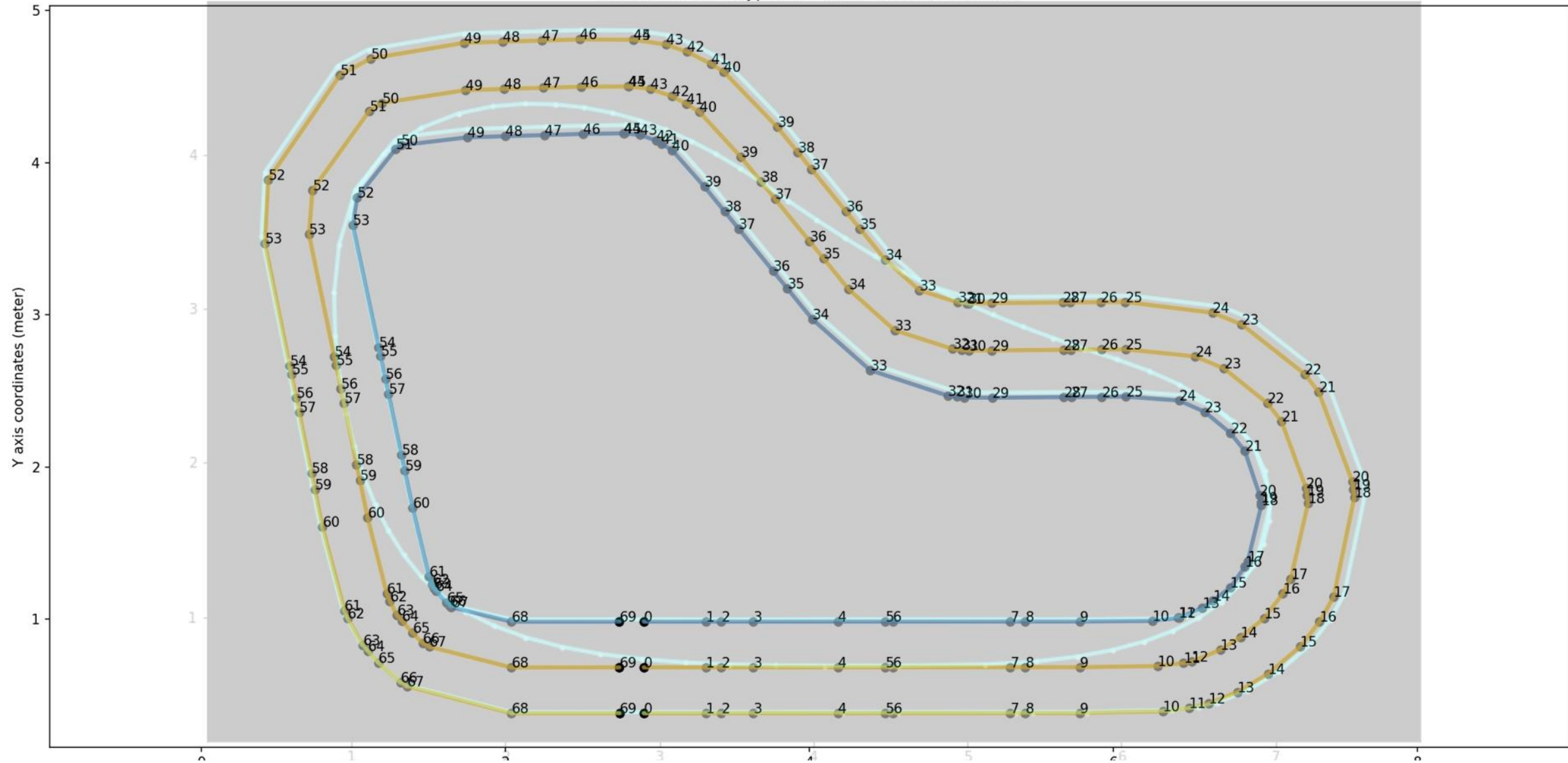




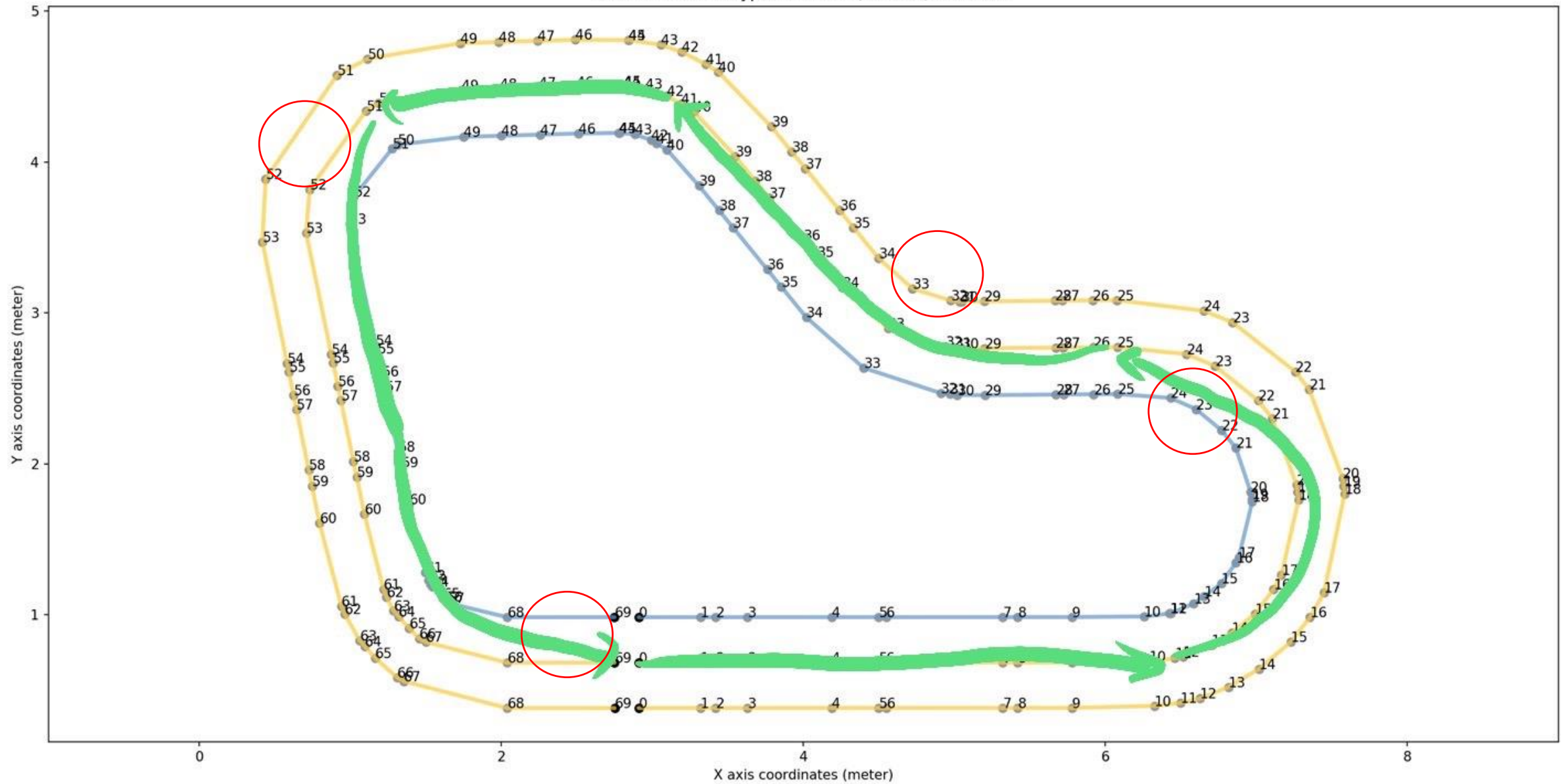
reInvent Track Waypoints - Inner, Center, and Outer



reInvent Track Waypoints - Inner, Center, and Outer



reInvent Track Waypoints - Inner, Center, and Outer



# Advanced Example- Waypoints Template

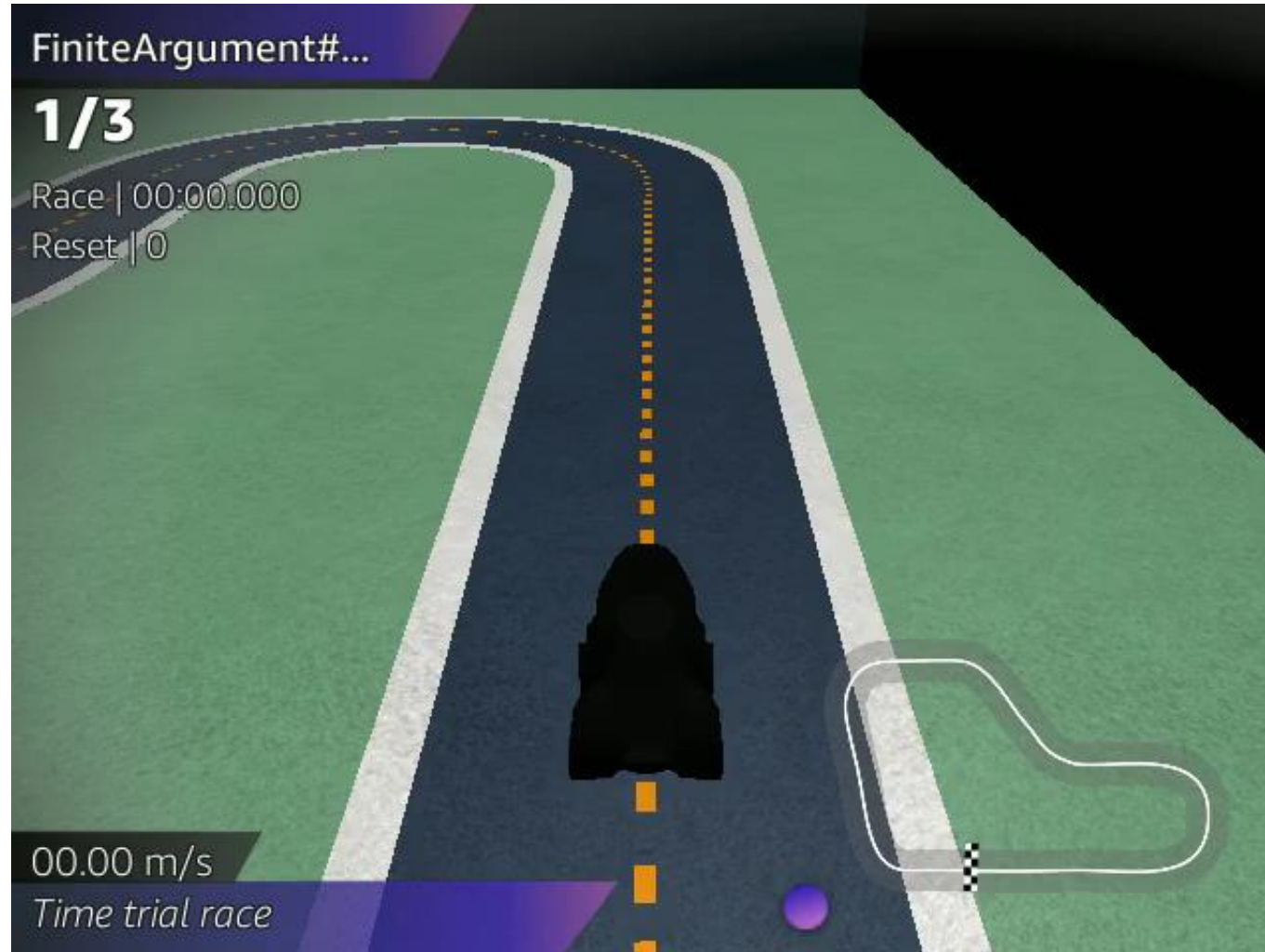
```
left_lane = [23, 24, 50, 51, 52, 53, 61, 62, 63, 64, 65, 66, 67, 68] # Fill in the waypoints

center_lane = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 27, 28, 35, 36,
               37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 54, 55, 56, 57, 58, 59, 60, 69,
               70] # Fill in the waypoints

right_lane = [29, 30, 31, 32, 33, 34] # Fill in the waypoints

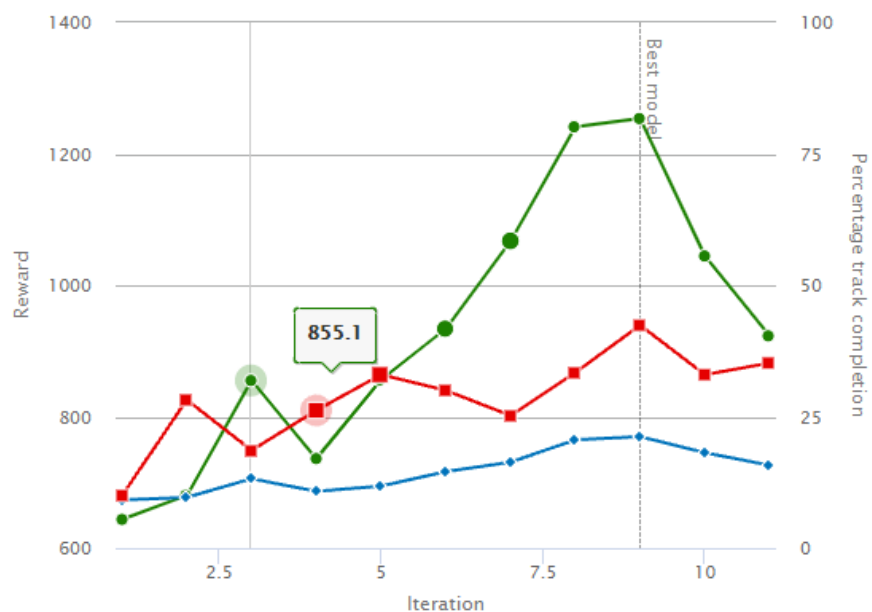
# Speed
fast = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44,
        45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69,
        70] # Fill in the waypoints, 2m/s
slow = [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24] # Fill in the waypoints, 1m/s
```

# Advanced Example- Waypoints Template



# Advanced Example- Result & Clone model

Reward graph [Info](#)



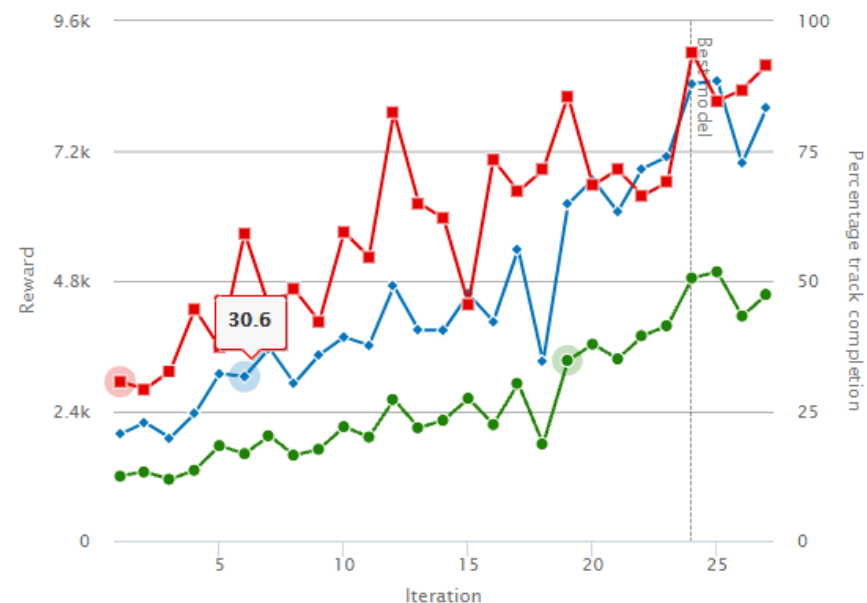
- Average reward (Training)
- Average percentage completion (Training)
- Average percentage completion (Evaluating)

## Stop condition

Maximum time

00:30:00 / 00:30:00

Reward graph [Info](#)



- Average reward (Training)
- Average percentage completion (Training)
- Average percentage completion (Evaluating)

## Stop condition

Maximum time

03:30:00 / 03:30:00