DeepRacer: The Blind Leads the Blind

David Darling and Ivy Truong

Project Statement:

- Expand knowledge with Machine Learning
- Optimize reward function to provide consistency and versatility

Methods

- Keep DeepRacer driving on track
- Reward DeepRacer for making progress to complete a lap on the track
- Maintain a minimum speed for DeepRacer
- Continue testing model if reward function continues to improve over time
- Testing DeepRacer on a different track

Experiment, Results, and Demo

Reward Function

```python
# V8-3

def reward_function(params):

    # Read input parameters
    track_width = params['track_width']
    distance_from_center = params['distance_from_center']
    progress = params['progress']
    speed = params['speed']
    all_wheels_on_track = params['all_wheels_on_track']

    # Calculate 3 markers that are at varying distances away from the center line
    marker_1 = 0.1 * track_width
    marker_2 = 0.25 * track_width
    marker_3 = 0.5 * track_width

    # Give higher reward if the car is closer to center line and vice versa
    if distance_from_center <= marker_1:
        reward = 10.0
    elif distance_from_center <= marker_2:
        reward = 5.0
    elif distance_from_center <= marker_3:
        reward = 1.0
    else:
        reward = 1e-3  # likely crashed/ close to off track

    # Reward the deepracer based on progress made after completing 25% of the track.
    if progress >= 25:
        reward = reward + (progress / 100) # 25% -> reward + 0.25 // 60% -> reward + 0.60

    # Reward the deepracer if it's speed is higher than 1.0 m/s, otherwise penalize it
    if speed > 0.65:
        reward += 0.5
    else:
        reward -= 0.5

    # Check if all wheels are on track, if they are reward deepracer, if not give penalty
    if all_wheels_on_track:
        reward += 1.0
    else:
        reward -= 3.0

    # Check if reward is 0 or below and if so make reward a small decimal value
    if reward <= 0.0:
        reward = 1e-6

    return float(reward)
```

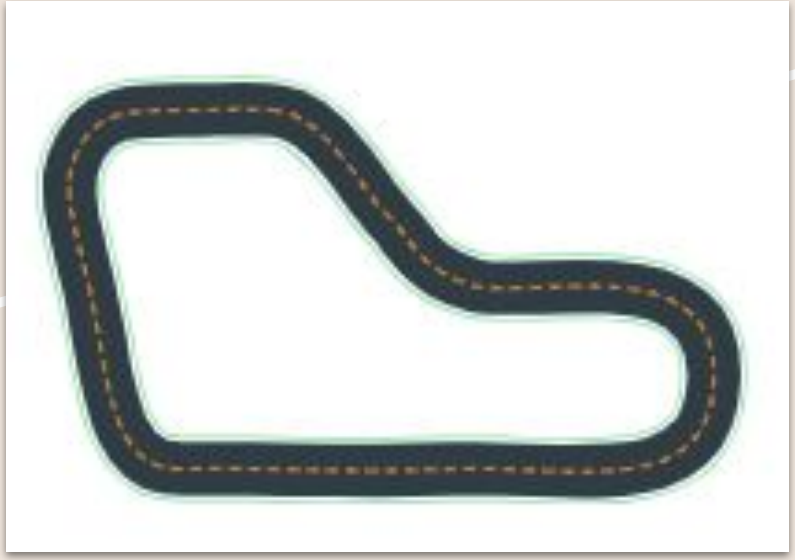
Hyperparameters

Hyperparameter	Value
Gradient descent batch size	64
Entropy	0.01
Discount factor	0.999
Loss type	Mean squared error
Learning rate	0.0003
Number of experience episodes between each policy-updating iteration	20
Number of epochs	10

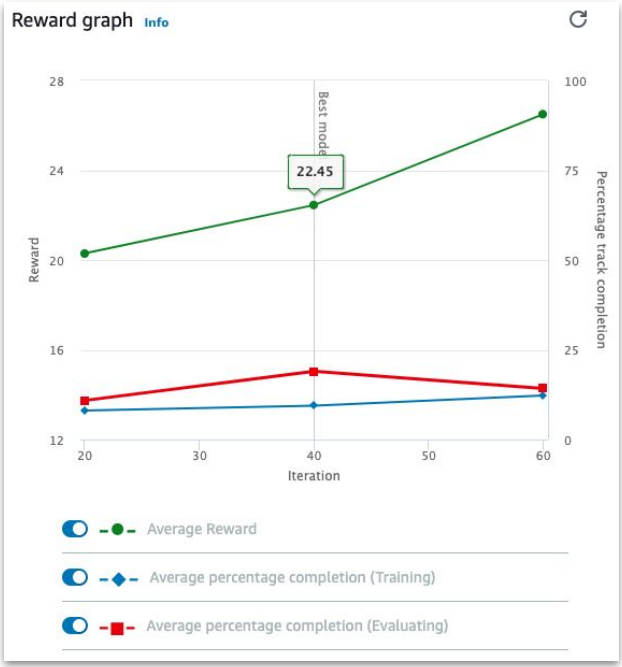
reInvent: 2018

Track Info

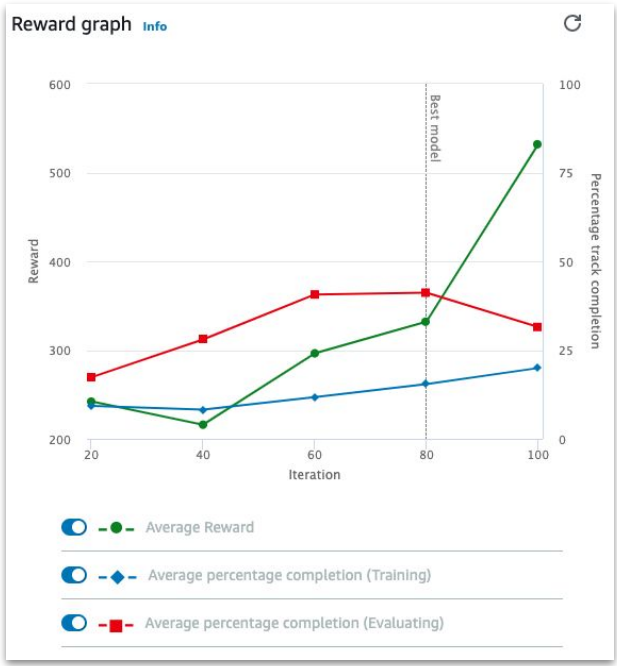
Length: 17.6 m
Width: 76 cm



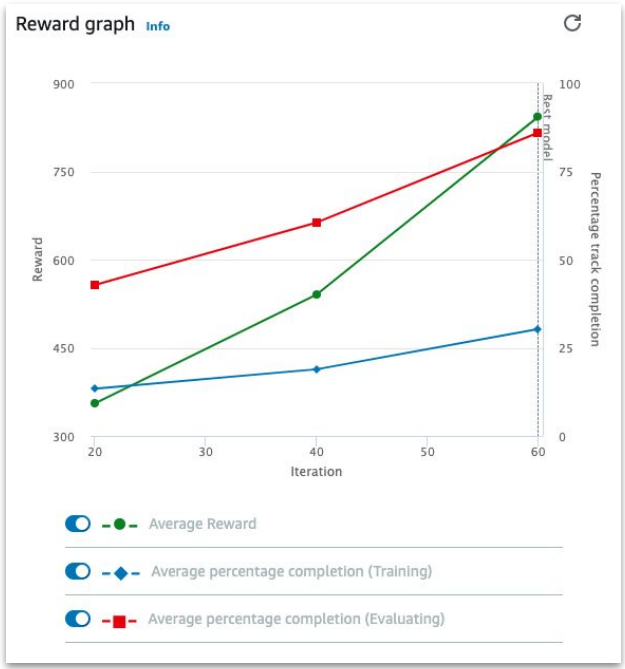
Training Reward Graphs



V2

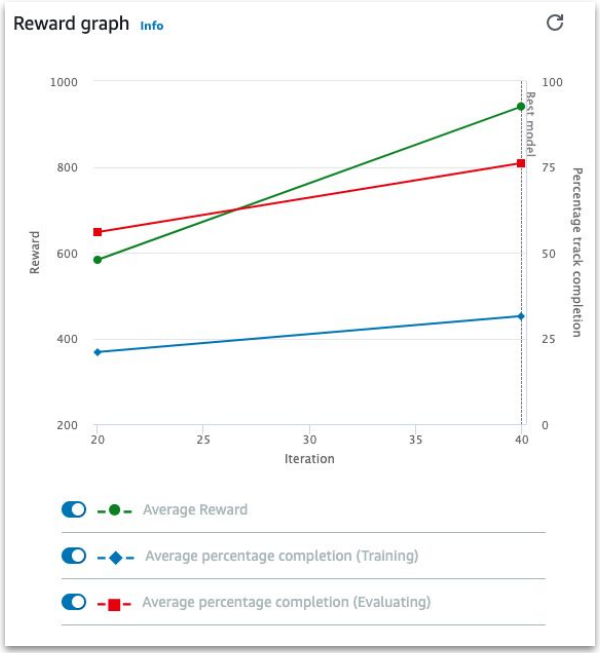


V8

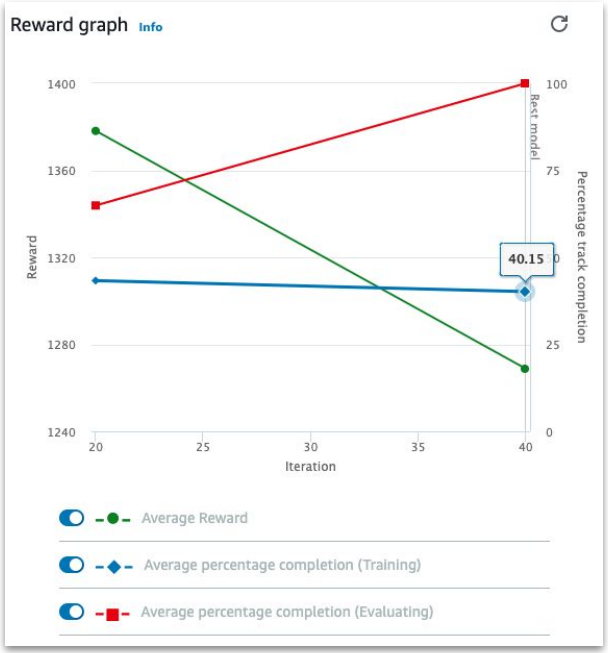


V8-1

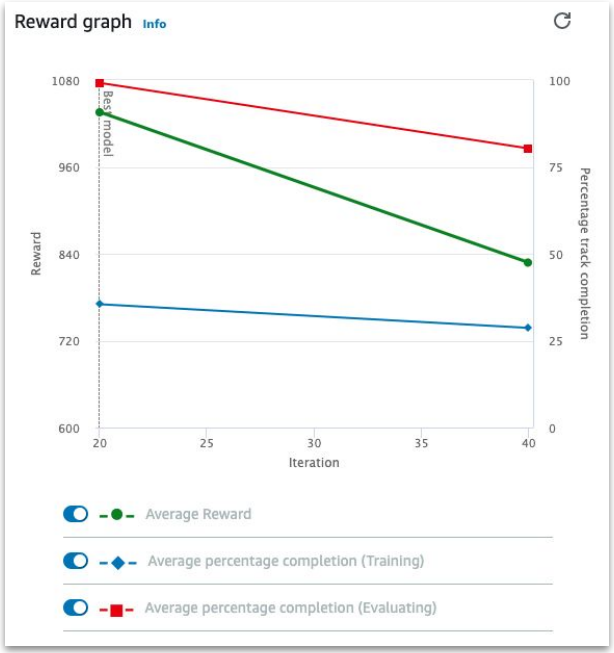
Training Reward Graphs Cont.



V8-2

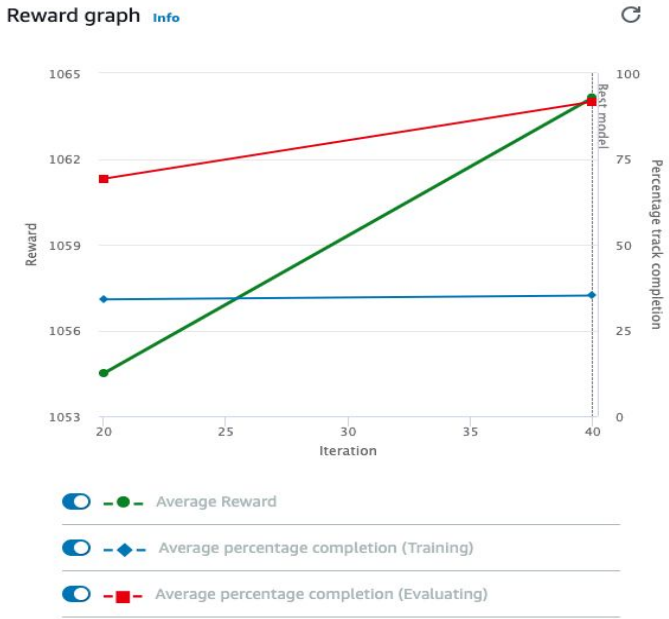


V8-3



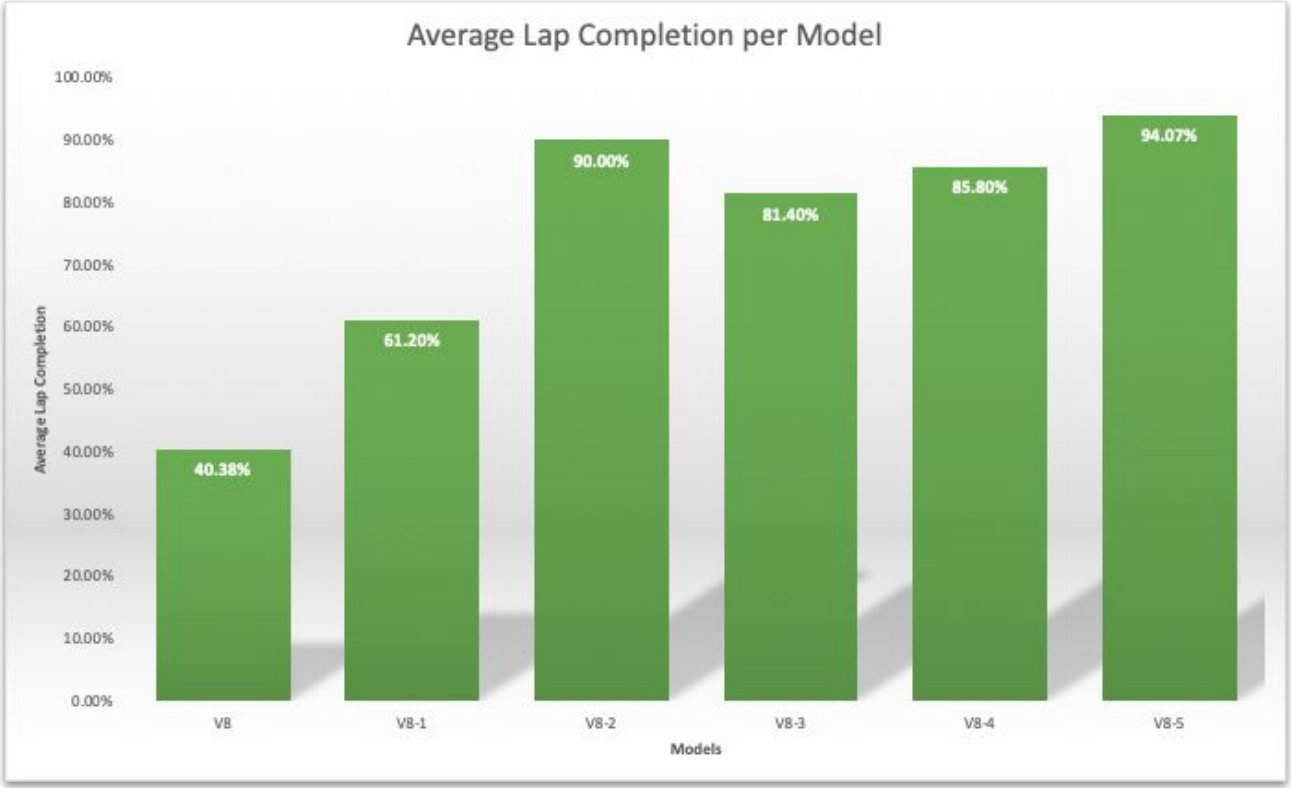
V8-4

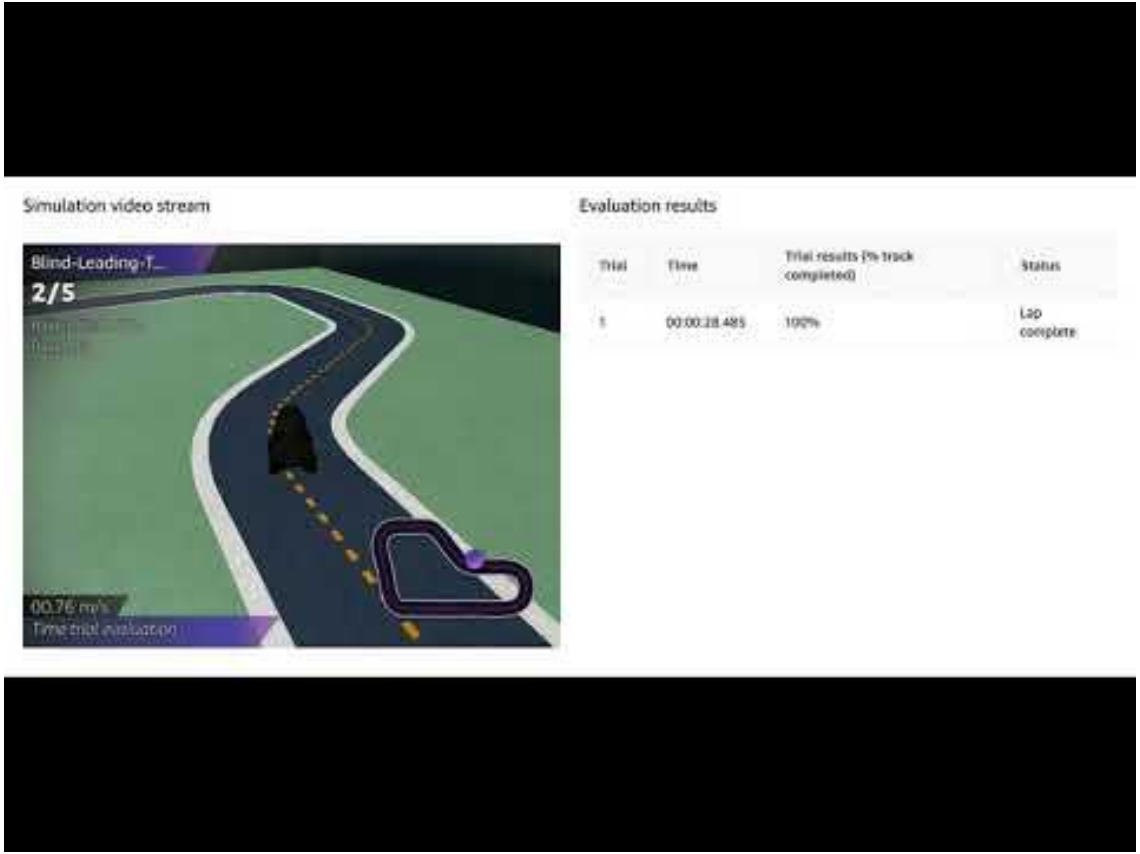
Training Reward Graphs Cont.



V8-5

Evaluations

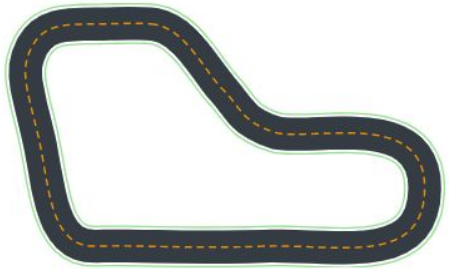




re:Invent 2018

Inspired by Monza, re:Invent 2018 was the first Championship Cup track. This short, classic speedway remains a perennial rookie favorite.

Length: 17.6 m (57.97')
Width: 76 cm (30")

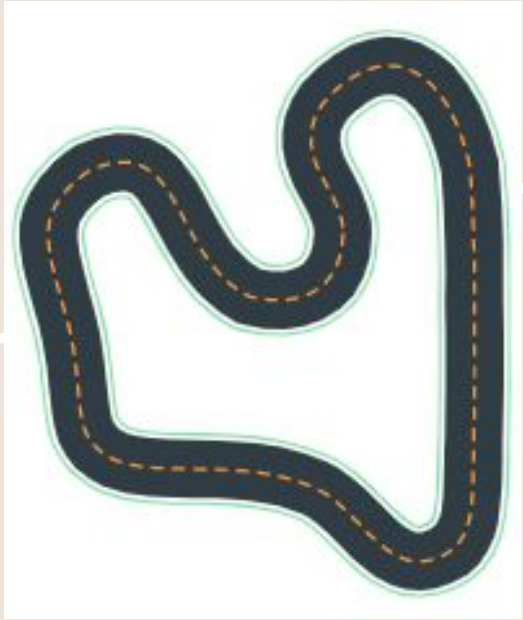


Video of the DeepRacer driving on the re:Invent 2018 track with the most updated reward function.

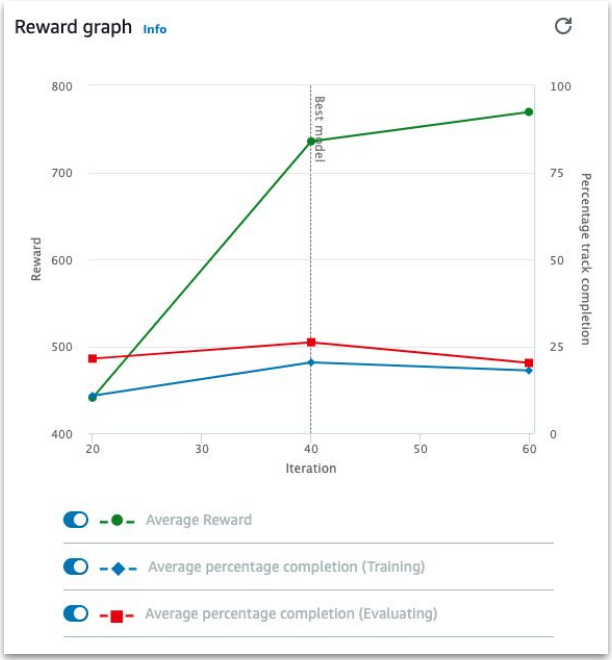
Shanghai Sudu

Track Info

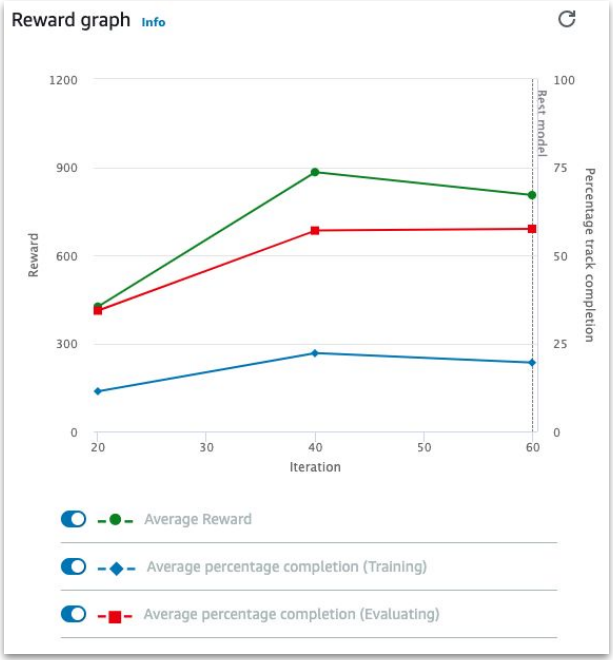
Length: 22.92 m
Width: 76 cm



Training Reward Graphs



V8-6



V8-7

[redacted]

Simulation video stream

Evaluation results

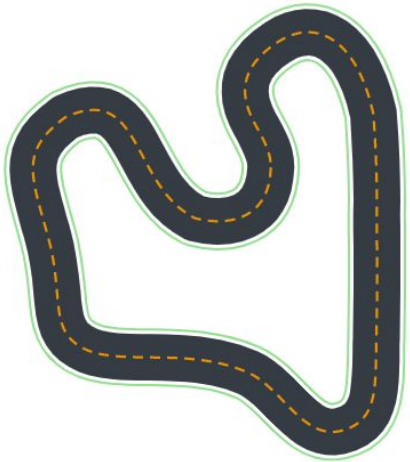
Trial	Time	Trial results (% track completed)	Status
1	00:00:17.908	46%	Off track
2	00:00:58.485	100%	Lap complete

[redacted]

Shanghai Sudu Training

August's training track for the 2019 Virtual Circuit World Tour, the Shanghai Sudu is an ideal speedway for experimenting with bold corners and basic straightaways.

Length: 22.92 m (75.2')
Width: 76 cm (30")



Video of DeepRacer driving on Shanghai Sudu Training track with most updated reward function.

Limitations and Future Works

Limitations

Local Environment

- Difficult to set up
  - Many libraries to set up
  - Deprecated githubs
- Space Requirement
  - Running another OS
    - Linux
  - Storage for data
- Favors certain GFX cards
- Virtualbox

AWS Console

- Flexibility in environment design
  - RL Algorithm
- Cost
- Time

Future Work

- More training time for our model
- Improve speed and lap times
- Try out more parameters
  - Waypoints
  - Steering angle
- Optimizing hyperparameters
- Watch a physical DeepRacer run using our reward function

Questions?