Park It Right!

Keerthana Nandanavanam
nandanav@usc.edu
Krishna Manoj Maddipatla
km69564@usc.edu
Nidhi Chaudhary
nidhicha@usc.edu
Sumanth Mothkuri
mothkuri@usc.edu

Agenda

01 OBJECTIVE

02 Simulation Environment

03 Research

04 Reward System

05 Hyperparameter Tuning

06 Results

07 Trained Model

08 Appendix

Objective

- Create an autonomous car parking agent to park in a designated spot in a simulated environment with obstacles

- Real life applications
  - Automated driving cars
  - Parking in crowded areas
  - Vacuum cleaners

Agenda

01 Objective

02 SIMULATION ENVIRONMENT

03 Research

04 Reward System

05 Hyperparameter Tuning

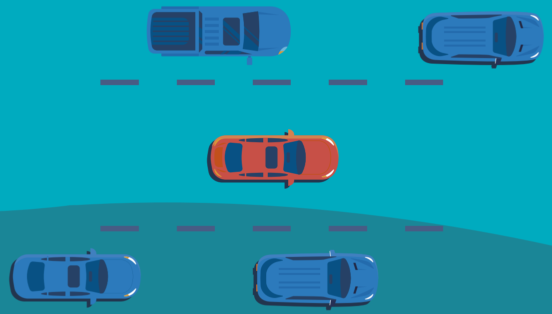
06 Results

07 Trained Model

08 Appendix

Game Environment

- Open source 3D game in Unity
- **Level 1:**
    - A bounded arena consisting of multiple parking spots
    - Randomly highlighted spots
    - Fixed obstacles
- **Level 2:**
    - Two storey, bounded arena
    - Moving obstacles



Game Modifications

SCOREBOARD

Parking Score,
Obstacle Hit Score,
Wall Hit Score,
Cumulative Reward

NAVIGATION

Keyboard navigation instead of Touchscreen

OBSTACLES

Converted boundaries and walls to collision objects

PARKING SPOTS

Randomly assign a parking spot from the set of available spots

Agenda

01 Objective

02 Simulation Environment

03 RESEARCH

04 Reward System

05 Hyperparameter Tuning

06 Results

07 Trained Model

08 Appendix

Research

- Reinforcement Learning
  - Difficult to get training data in supervised learning
  - End goal for the Agent is to discover a behavior (a Policy) that maximizes a reward
  - Good Support provided by Unity ML-agents package

Proximal Policy Optimization

- Training data is generated based on the current policy rather than relying on static data
- Involves collecting a small batch of experiences interacting with the environment and using that batch to update its decision-making policy
- More stable than Deep Q Learning
- Easy to implement and tune

Imitation Learning

- Provide the agent with a set of demonstrations.

- The agent then tries to learn the optimal policy by imitating the expert's decisions.

- Generative Adversarial Imitation Learning(GAIL) - directly extracting a policy from data, as if it were obtained by reinforcement learning following inverse reinforcement learning

Agent design

INPUT OBSERVATION SPACE

- Size = 27
- Relative and normalized distance

EPISODE BEGIN

- Random target location is generated
- Car Agent will start at random location

HEURISTIC

- Take left
- Take right
- No action

EPISODE END

- Parked correctly
- Hit Obstacle/Wall

Agenda

01 Objective

02 Simulation Environment

03 Research

04 **REWARD SYSTEM**

05 Hyperparameter Tuning

06 Results

07 Trained Model

08 Appendix

Reward System

S.No	Condition	Reward [PPO]	Reward [GAIL]
1.	**Hit the wall [Episode Ends]**	**-0.5**	**-0.5**
2.	**Hit an obstacle [Episode Ends]**	**-0.5**	**-0.5**
3.	**Car Parked [Episode Ends]**	**+5**	**+5**
4.	Within 2.5 units of distance to the goal location	+0.00008	+0.00003
5.	Best current distance to the goal location	+0.00002	+0.00002
6.	Moving towards the goal but not the best distance to the goal in the current episode	-0.00004	+0.00001
7.	Moving away from the goal	-0.00008	-0.00002
8.	Within 2 units of distance to the wall	-0.005	-0.005
9.	Within 2 units of distance to the obstacle	-0.005	-0.005

Agenda

01 Objective

02 Simulation Environment

03 Research

04 Reward System

05 HYPERPARAMETER TUNING

06 Results

07 Trained Model

08 Appendix

Hyperparameters

- Performed on 9 different hyperparameters *
- Low learning rate of 1e-05, high batch and buffer size for stability

PPO + LSTM

- batch size = 512
- buffer size = 10240
- beta = 0.001
- epsilon = 0.3
- hidden units= 64
- Number of layers = 2
- Normalize = True
- lambd=0.92

PPO + LSTM + GAIL

- batch size = 256
- buffer size = 20480
- beta = 0.03
- epsilon = 0.1
- hidden units= 64
- Number of layers = 2
- Normalize = False
- lambd=0.92
- **Gail strength = 0.7**

* Refer appendix

Agenda

01 Objective

02 Simulation Environment

03 Research

04 Reward System

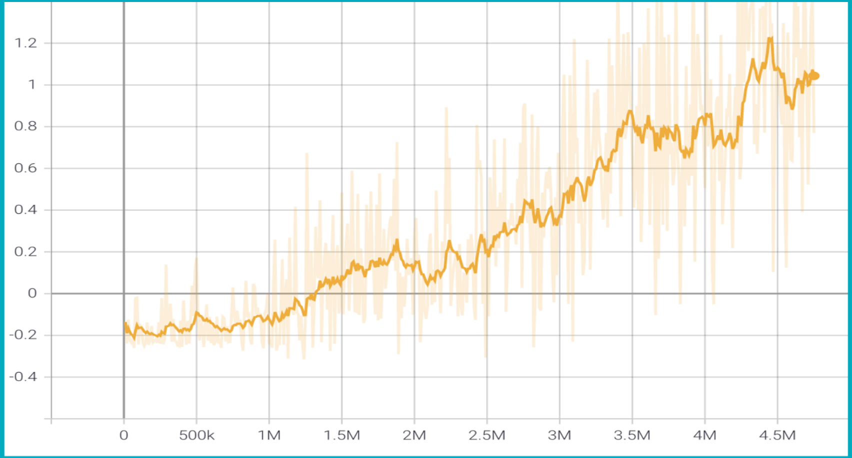
05 Hyperparameter Tuning

06 RESULTS

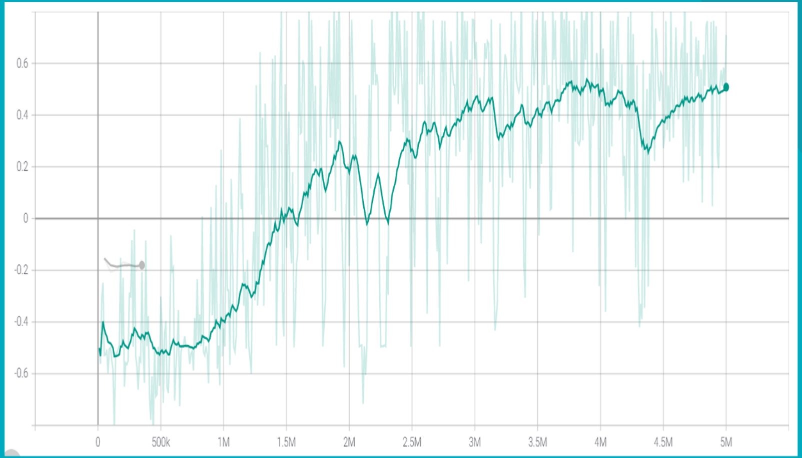
07 Trained Model

08 Appendix

Results



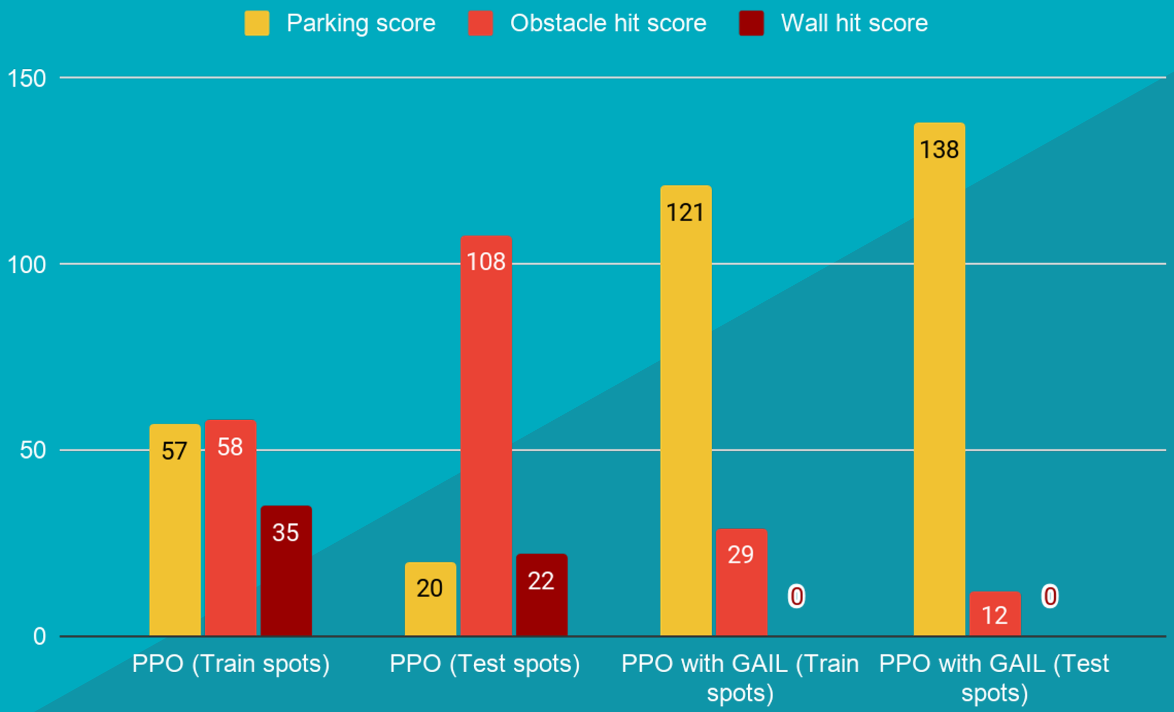
PPO



PPO with GAIL

- Cumulative rewards keep on increasing with the number of steps for both PPO and PPO with GAIL.
- Entropy decreases for both as well!

Inference Statistics



138/150

Times parked with PPO + GAIL for new locations

121/150

Times parked with PPO + GAIL for same locations

GAIL with PPO performs much better!

Agenda

01 Objective

02 Simulation Environment

03 Research

04 Reward System

05 Hyperparameter Tuning

06 Results

07 TRAINED MODEL

07 Appendix

Demo

Future Work

1. LEVEL 2 RESEARCH
2. GAME MODIFICATION
3. LEVEL 2 TRAINING

Work Division

KEERTHANA

- Algorithm Research
- GAIL Implementation
- Hyperparameter Tuning
- Positive reward system design

KRISHNA

- Game modifications
- PPO, GAIL
- Hyperparameter Research & Design
- Positive reward system design

NIDHI

- Algorithm Research
- PPO, RDN
- Hyperparameter Research & Design
- Negative reward system design

SUMANTH

- Curiosity Learning
- Hyperparameter Tuning
- Object detection
- Architecture Design

THANK YOU

AND STAY TUNED FOR LEVEL 2!

ANY QUESTIONS?

APPENDIX

Project Timeline

Hyperparameters

learning rate = 1e-05
Lambd = 0.92
No normalization

SL.No	Parameters	Steps	Result
1	PPO, batch size = 256 , buffer size = 10240, beta = 0.01, epsilon = 0.3, layers = 2, hidden units = 128, time horizon = 256	5M	✖
2	PPO, batch size = 32, buffer size = 2048, beta = 0.01, epsilon = 0.3, layers = 2, hidden units = 64, time horizon = 128	1M	✖
3	PPO, batch size = 32, buffer size = 3028, beta = 0.03, epsilon = 0.1, layers = 2, hidden units = 64, time horizon = 256	1M	✖
4	PPO, batch size = 256, buffer size = 20480, beta = 0.03, epsilon = 0.1, layers = 2, hidden units = 64, time horizon = 256	1M	✖
5	PPO, batch size = 256, buffer size = 20480, beta = 0.03, epsilon = 0.1, layers = 3, hidden units = 128, time horizon = 256	5M	✖

Hyperparameters

learning rate = 1e-05
Lambd = 0.92

SL.No	Parameters	Steps	Result
6	PPO with RND, gamma: 0.99, strength: 0.01, encoding_size: 64, learning_rate: 0.0001, batch size = 512, buffer size = 10240, beta = 0.001, epsilon = 0.3, normalize = True, layers = 2, hidden units = 64, time horizon = 128	4M	✗
7	PPO with Curiosity, gamma: 0.99, strength: 0.2, encoding_size: 128, learning_rate: 0.0001, batch size = 512, buffer size = 10240, beta = 0.001, epsilon = 0.3, normalize = True, layers = 2, hidden units = 64, time horizon = 128	1M	✗
8	**PPO, LSTM, batch size = 512, buffer size = 10240, beta = 0.001, epsilon = 0.3, hidden units= 64 number of layers = 2, normalize = True**	5M	✓
9	**PPO with gail, LSTM, batch size = 256, buffer size = 20480, beta = 0.03, epsilon = 0.1, hidden units = 64, number of layers = 2, gail strength = 0.7, normalize = False**	5M	✓