Autonomous driving via reinforcement learning

Roy Amante Salvador and Maria Isabel Saludares

CS 295 Reinforcement Learning | Project Presentation

SAE (J3016) automation levels

<table>
<tr><th>SAE Level</th><th>Name</th><th colspan="2">Narrative definition</th><th>Execution of steering and acceleration/ deceleration</th><th>Monitoring of driving environment</th><th>Fallback performance of dynamic driving task</th><th>System capability (driving modes)</th></tr>
<tr><td colspan="8">Human driver monitors the driving environment</td></tr>
<tr><td>0</td><td>No Automation</td><td colspan="2">The full-time performance by the human driver of all aspects of the dynamic driving task, even when "enhanced by warning or intervention systems"</td><td>Human driver</td><td rowspan="3">Human driver</td><td rowspan="3">Human driver</td><td>n/a</td></tr>
<tr><td>1</td><td>Driver Assistance</td><td>The driving mode-specific execution by a driver assistance system of "either steering or acceleration/deceleration"</td><td rowspan="2">using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task</td><td>Human driver and system</td><td rowspan="2">Some driving modes</td></tr>
<tr><td>2</td><td>Partial Automation</td><td>The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration</td><td>System</td></tr>
<tr><td colspan="8">Automated driving system monitors the driving environment</td></tr>
<tr><td>3</td><td>Conditional Automation</td><td rowspan="3">The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task</td><td>with the expectation that the human driver will respond appropriately to a request to intervene</td><td rowspan="3">System</td><td rowspan="3">System</td><td>Human driver</td><td>Some driving modes</td></tr>
<tr><td>4</td><td>High Automation</td><td>even if a human driver does not respond appropriately to a request to intervene</td><td rowspan="2">System</td><td>Many driving modes</td></tr>
<tr><td>5</td><td>Full Automation</td><td>under all roadway and environmental conditions that can be managed by a human driver</td><td>All driving modes</td></tr>
</table>

Autonomous Driving

Google's AVs (Waymo)

- ➢ *5M+ km driven*
- ➢ *american avenues, streets, and roads*

Uber's AVs

- ➢ 1.5M+ km in testing
- ➢ Pittsburgh, Phoenix, San Francisco, Toronto
- ➢ crash: killed a pedestrian in Arizona

autonomous levels (SAE J3016)

1 **Driver Assistance: *hands on***

2 **Partial Automation: *hands off***

3 **Conditional Automation: *eyes off***

4 **High Automation: *minds off***

5 **Full Automations: *steering wheel optional***

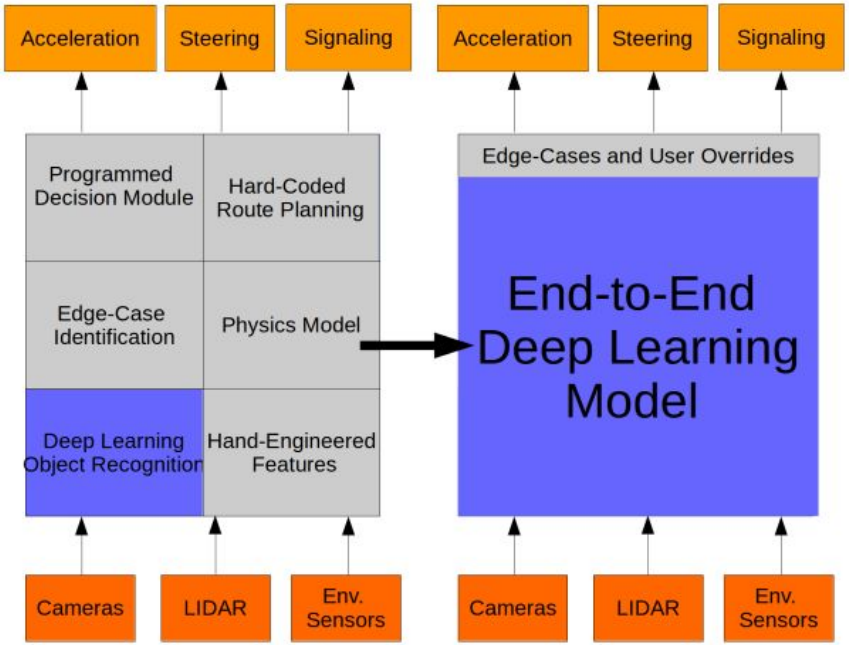
Autonomous Driving

Five increasingly sophisticated and autonomous levels (SAE J3016):

1	**Driver Assistance: *hands on***	vehicle performs minor steering or acceleration tasks; all other operations are under full human control
2	**Partial Automation: *hands off***	vehicle automatically responds to safety situations, but the driver must remain alert and responsive.
3	**Conditional Automation: *eyes off***	vehicle performs certain “safety-critical functions” under various traffic or environmental conditions.
4	**High Automation: *minds off***	vehicle can operate without requiring human input.
5	**Full Automations: *steering wheel optional***	vehicle operates with full automation in any environment (weather or traffic).

Adoption

legal liability
policy makers
customer acceptance
(cost, infrastructure, technology)



Rosenzweig, J., & Bartl, M. (2015). A review and analysis of literature on autonomous driving. *E-Journal Making-of Innovation*.
https://towardsdatascience.com/reinforcement-learning-from-grid-world-to-self-driving-cars-52bd3e647bc4

Autonomous Driving

Maker	2016	
	Distance between disengagements	Distance
Waymo	5,127.9 miles (8,252.6 km)	635,868 miles (1,023,330 km)
BMW	638 miles (1,027 km)	638 miles (1,027 km)
Nissan	263.3 miles (423.7 km)	6,056 miles (9,746 km)
Ford	196.6 miles (316.4 km)	590 miles (950 km)
General Motors	54.7 miles (88.0 km)	8,156 miles (13,126 km)
Delphi Automotive Systems	14.9 miles (24.0 km)	2,658 miles (4,278 km)
Tesla	2.9 miles (4.7 km)	550 miles (890 km)
Mercedes Benz	2 miles (3.2 km)	673 miles (1,083 km)
Bosch	0.68 miles (1.09 km)	983 miles (1,582 km)
Volkswagen	5.56 miles (8.95 km)	9 miles (14 km)

Wang, Brian (25 March 2018). "Uber' self-driving system was still 400 times worse [than] Waymo in 2018 on key distance intervention metric". NextBigFuture.com. Retrieved 25 March2018.

<table>
<tr><th>SAE Level</th><th>Name</th><th colspan="2">Narrative definition</th><th>Execution of steering and acceleration/ deceleration</th><th>Monitoring of driving environment</th><th>Fallback performance of dynamic driving task</th><th>System capability (driving modes)</th></tr>
<tr><td colspan="8">Human driver monitors the driving environment</td></tr>
<tr><td>0</td><td>No Automation</td><td colspan="2">The full-time performance by the human driver of all aspects of the dynamic driving task, even when "enhanced by warning or intervention systems"</td><td>Human driver</td><td rowspan="3">Human driver</td><td rowspan="3">Human driver</td><td>n/a</td></tr>
<tr><td>1</td><td>Driver Assistance</td><td>The driving mode-specific execution by a driver assistance system of "either steering or acceleration/deceleration"</td><td rowspan="2">using information about the driving environment and with the expectation that the human driver performs all remaining aspects of the dynamic driving task</td><td>Human driver and system</td><td rowspan="2">Some driving modes</td></tr>
<tr><td>2</td><td>Partial Automation</td><td>The driving mode-specific execution by one or more driver assistance systems of both steering and acceleration/deceleration</td><td>System</td></tr>
<tr><td colspan="8">Automated driving system monitors the driving environment</td></tr>
<tr><td>3</td><td>Conditional Automation</td><td rowspan="3">The driving mode-specific performance by an automated driving system of all aspects of the dynamic driving task</td><td>with the expectation that the human driver will respond appropriately to a request to intervene</td><td rowspan="3">System</td><td rowspan="3">System</td><td>Human driver</td><td>Some driving modes</td></tr>
<tr><td>4</td><td>High Automation</td><td>even if a human driver does not respond appropriately to a request to intervene</td><td rowspan="2">System</td><td>Many driving modes</td></tr>
<tr><td>5</td><td>Full Automation</td><td>under all roadway and environmental conditions that can be managed by a human driver</td><td>All driving modes</td></tr>
</table>

Simulation to Real Environment

generating simulation data for training

visoumotor representation from simulated to real environment [1]

robotic control transfer from simulation to real [2, 3]

reinforcement learning with imagined goals [4]

1. Towards Adapting Deep Visuomotor Representations from Simulated to Real Environments, Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, Trevor Darrell. In the proceedings of the Workshop on Algorithmic Foundations of Robotics (WAFR), San Francisco, CA, USA, December 2016. (arXiv 1511.07111)
2. Sim-to-Real Transfer of Robotic Control with Dynamics Randomization, Xue Bin (Jason) Peng, Marcin Andrychowicz, Wojciech Zaremba, Pieter Abbeel. In the proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, May 2018. (arXiv 1710.064537, video)
3. Transfer from Simulation to Real World through Learning Deep Inverse Dynamics Model, Paul Christiano, Zain Shah, Igor Mordatch, Jonas Schneider, Trevor Blackwell, Joshua Tobin, Pieter Abbeel, Wojciech Zaremba. arXiv 1610.03518
4. Nair, A. V., Pong, V., Dalal, M., Bahl, S., Lin, S., & Levine, S. (2018). Visual reinforcement learning with imagined goals. In Advances in Neural Information Processing Systems (pp. 9191-9200).

Environment Simulators

Microsoft AirSim (Unreal, Unity)

Unity ML (Unity)

Udacity

physics engine
can simulate any scenario, run tests before deploying to autonomous vehicle (AV)
flexibility in setting environment conditions: weather, time of day, etc
flexibility in building environments: build city scapes, rough terrains, etc.

Simulator

settings:

- collisions
- time of day
- weather (bugs)

Microsoft AirSim Neighborhood v 1.2.1 *simulator*

- access to car controls, like steering, velocity, acceleration, brakes
- has information on car states like speed, quaternion (position, velocity, acceleration, orientation)

End-to-end framework for autonomous driving

- raw sensor inputs -> driving actions (continuous)
- Handles partially observable scenarios
- Integration of attention models to extract relevant information

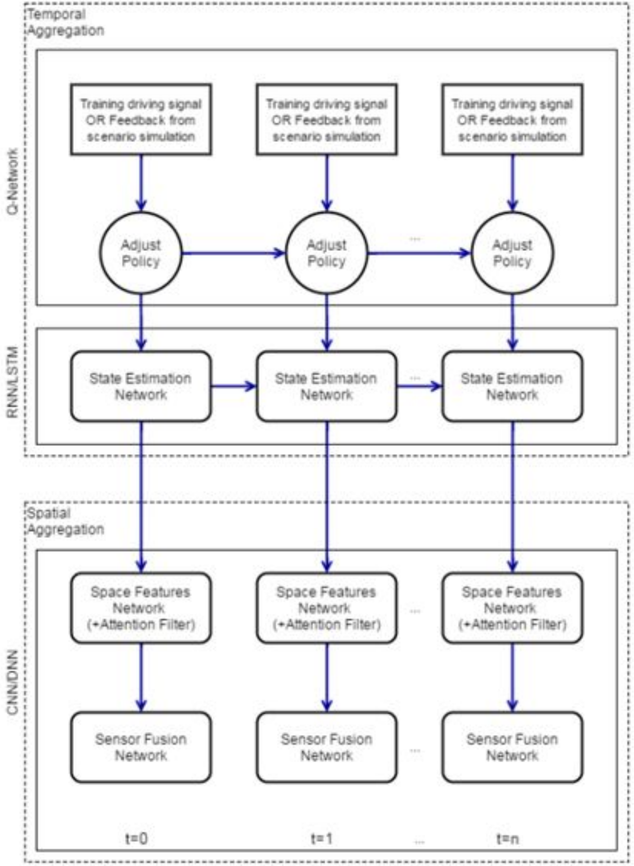
End-to-end framework for autonomous driving [1]:

spatial aggregation: sensor fusion, spatial features

action and glimpse network[2]

recurrent temporal aggregation

deep Recurrent Q-Learning: LSTM + DQNs = DQRN [3]



1. *Sallab, A. E., Abdou, M., Perot, E., & Yogamani, S. (2017). Deep reinforcement learning framework for autonomous driving. Electronic Imaging, 2017(19), 70-76.*
2. *"End-to-end Learning of Action Detection from Frame Glimpses in Videos" at https://arxiv.org/pdf/1511.06984.pdf*
3. *Hausknecht, M., & Stone, P. (2015, September). Deep recurrent q-learning for partially observable mdps. In 2015 AAAI Fall Symposium Series.*

Objectives

create a wrapper class for OpenAI Gym with AirSim

train a model for autonomous car driving via reinforcement learning

investigate the effects during the learning and its effect in training multiple policy networks:

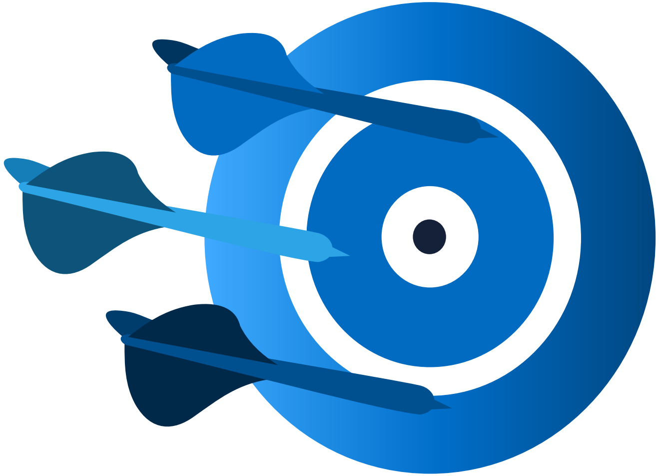
- MLP
- MLP LSTM
- Deep Meta RL

On different action spaces:

- break, gear, throttle, steering
- break, throttle, steering
- movement, steering

On various inputs:

- LIDAR



Environment

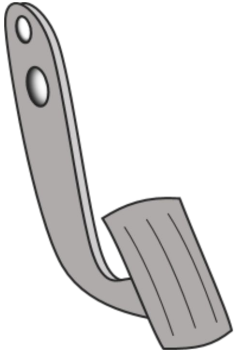
- created wrapper class for OpenAI Gym[1]
- Goal: reach waypoints (sequence in random)
- Conditions:
  - Agent car initially spawns at the middle of the map
  - Episode ends when all waypoints are reached or time runs out

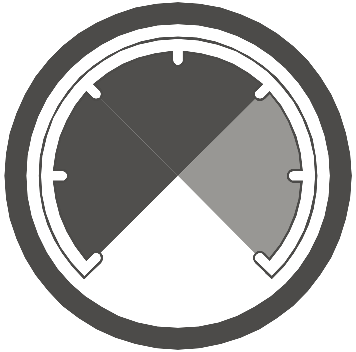
1. https://gym.openai.com/

Action Space

On different action spaces:

- break, gear, throttle, steering
- break, throttle, steering
- movement, steering





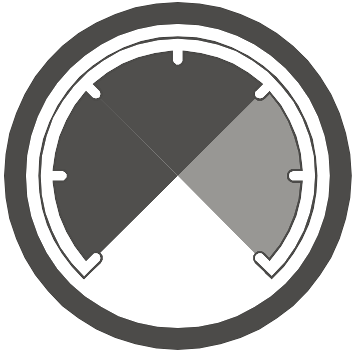
break
soft to hard break

throttle
accelerometer

gear
drive, neutral, reverse
(probability distribution)

steering
angle

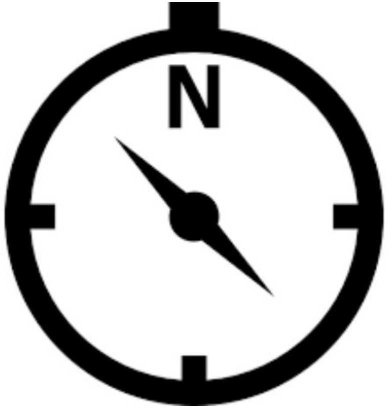
Observation Space



Speed /
Linear and
Angular
Velocity

Light Detection
And Ranging
(LIDAR)
reading per 1°

Range : 10
Rotations Per Second: 10
Points Per Second: 10000



Orientation
(yaw)

Reward Scheme

rewards (+)

Euclidean distance from current waypoint

$dist = | car_{pos} - target_{pos} |$

Orientation of car wrt current waypoint

$\cos(yaw_{target\ wrt\ car} - yaw_{car})$

penalties (-)

collision

$collision\ detected * w_{collision}$

reverse

$car\ reverses * w_{reverse}$

lack of movement speed

if $car_{vel} < car_{vel\_threshold}$

*reward and penalties are computed every timestep

Reinforcement Learning Algorithms

Used Stable Baselines[1]

Advantage Actor Critic (A2C)

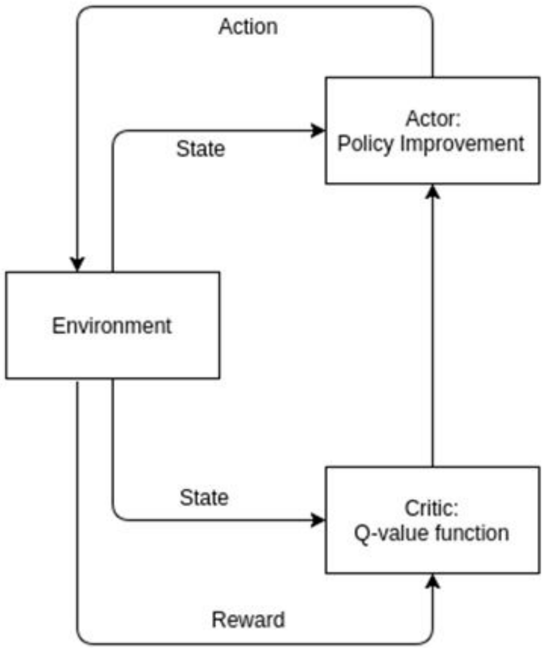
- Actor - controls the behavior of the agent
- Critic - measures the value of the action taken. Learns the advantage function instead of Q value function

Proximal Policy Optimization (PPO)

- an Actor Critic method by limiting how far we can change the policy each iteration and adding a soft constraint to the objective function

Soft Actor Critic (SAC)

- Off-Policy Maximum Entropy Deep Reinforcement Learning
- Has a stochastic actor to introduce robustness



1. https://github.com/hill-a/stable-baselines

Policy Networks

MLP

- 2 layers of 64 neurons

MLP-LSTM

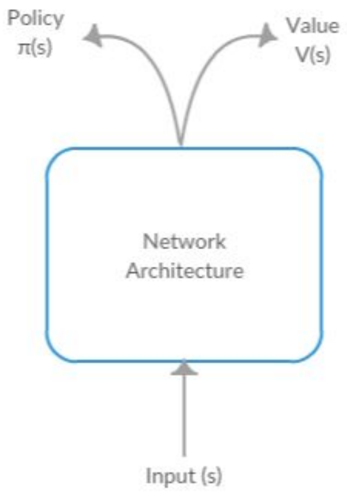
- 2 layers of 64 neurons followed by LSTM layer with 256 cells

Deep Meta-RL[1]

- 2 layers of 64 neurons followed by LSTM layer with 256 cells
- Reward and action from previous timestep as input

Custom MLP

- 3 normalized layer of 512 neurons



1. Learning to Reinforcement Learn, Jane X. Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z. Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, Matthew Botvinick 2016, (arXiv:1611.05763)

Framework

build the various action and observations spaces for the environment

connect to environment and train

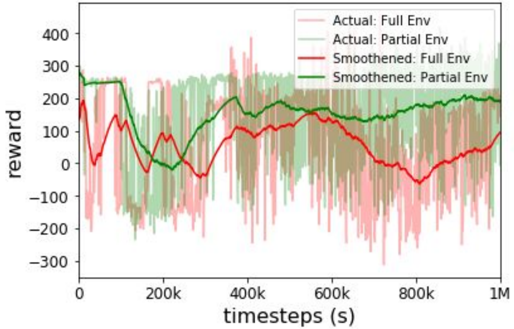
analyze

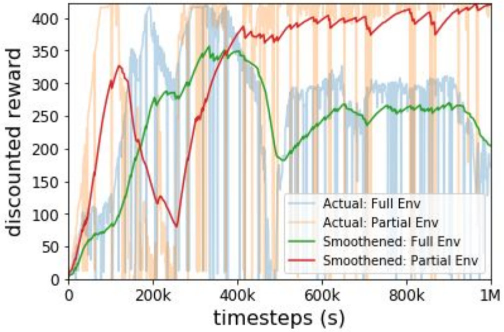
Experiments

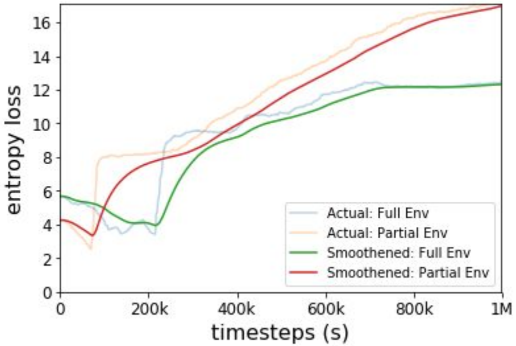
Algorithm-Policy	Action Space	Rewards/Penalties
A2C-MLP	Steering, throttle, break, reverse	Euclidean Distance Collision Reverse Lack of Movement Speed
	Steering, throttle, break, reverse (vary reward weight)	
	Steering, throttle, break	
	Steering, throttle, break (vary reward weight)	
A2C-MLP+LSTM	Steering, throttle, break, reverse	
	Steering, throttle, break	
PPO2-MLP	Steering, movement (Throttle, Brake, Reverse)	Orientation Euclidean Distance Collision Reverse Lack of Movement Speed
PPO2-MLP+LSTM	Steering, movement (Throttle, Brake, Reverse)	
PPO2-MetaRL	Steering, movement (Throttle, Brake, Reverse)	
SAC-LnMLP	Steering, movement (Throttle, Brake, Reverse)	

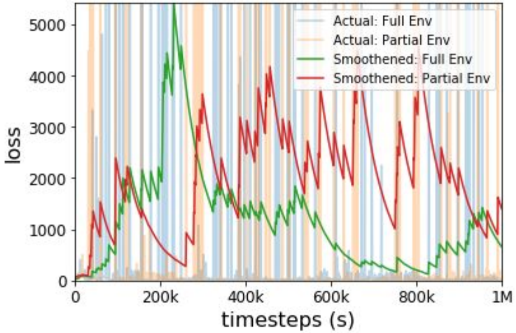
Results and Discussion

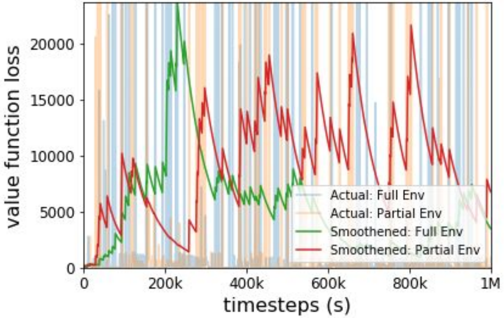
A2C
-
MLP

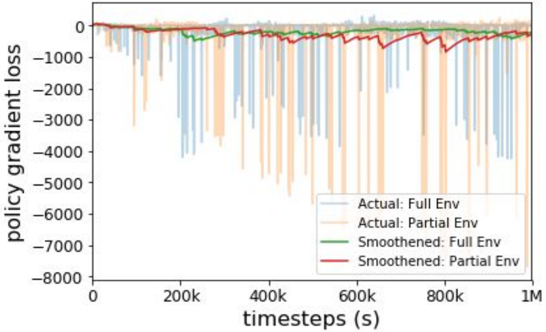






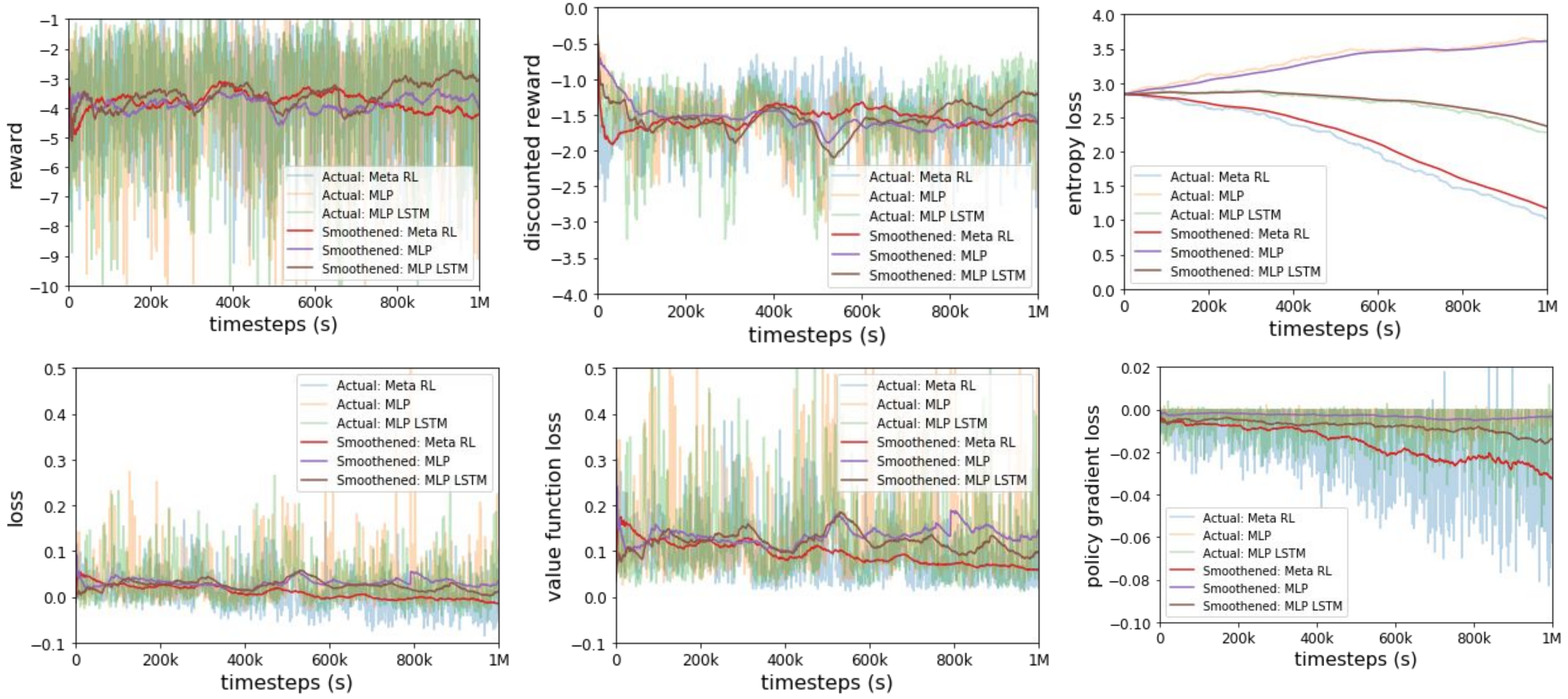






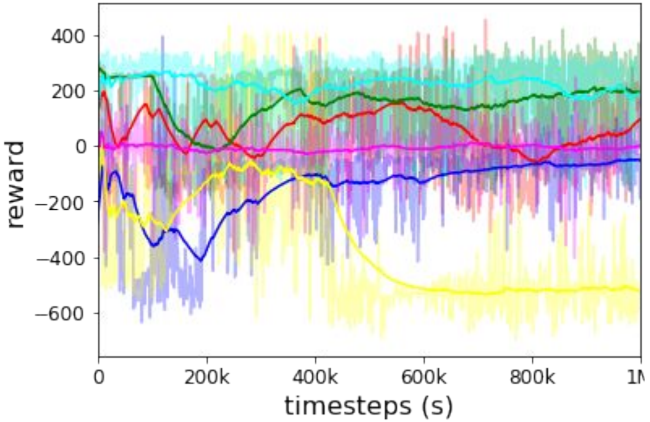
Results and Discussion

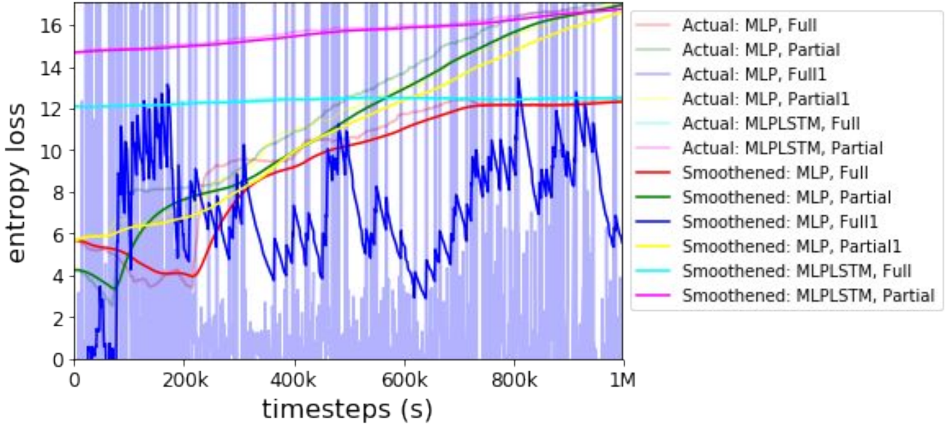
PPO2



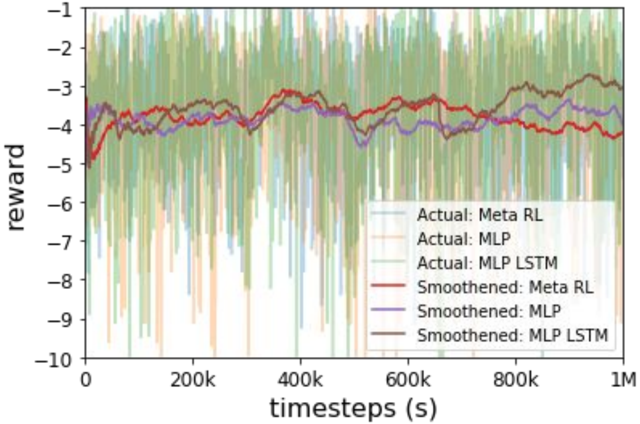
Results and Discussion

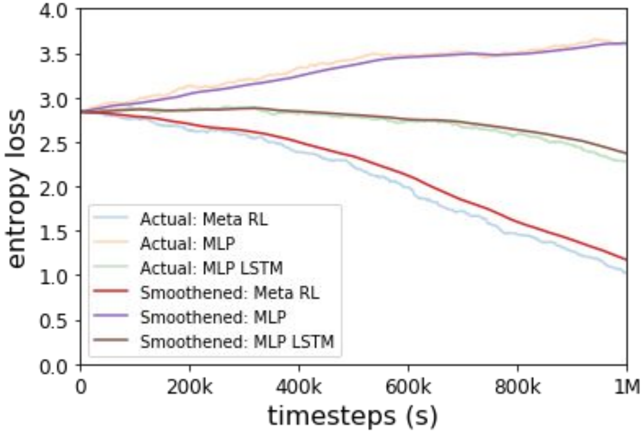
A2C





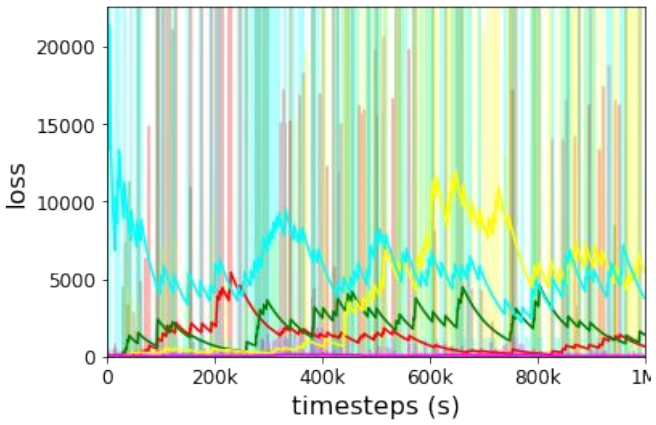
PPO2

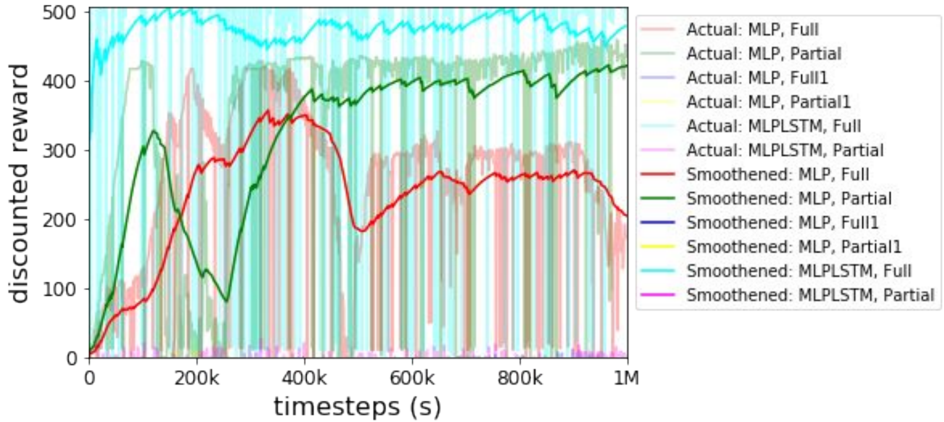




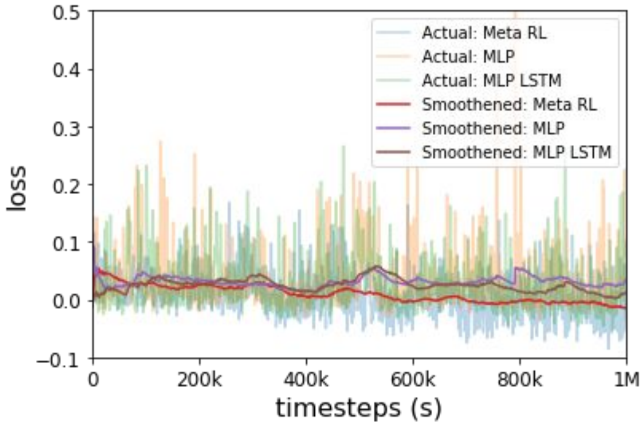
Results and Discussion

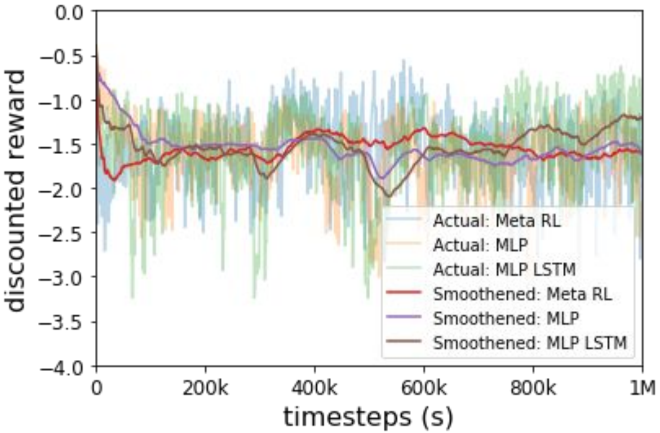
A2C





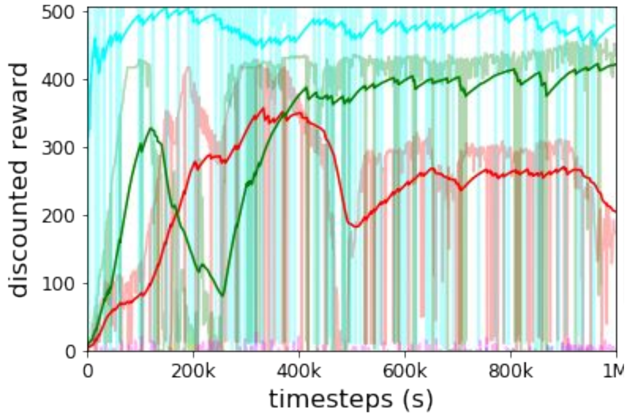
PPO2

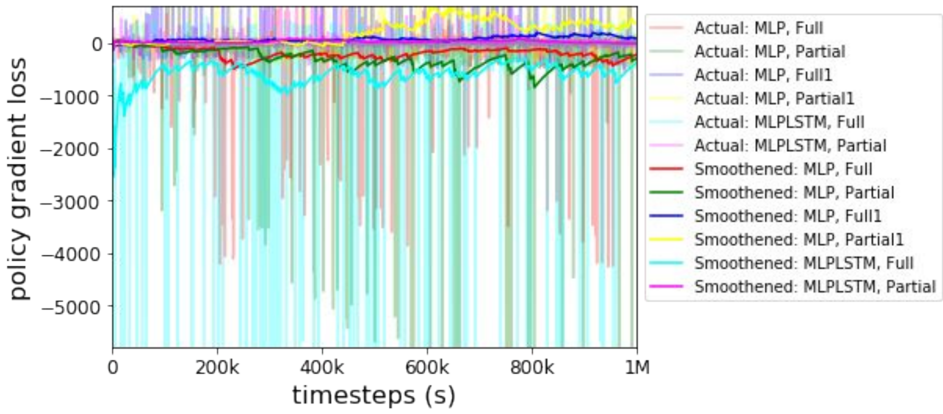




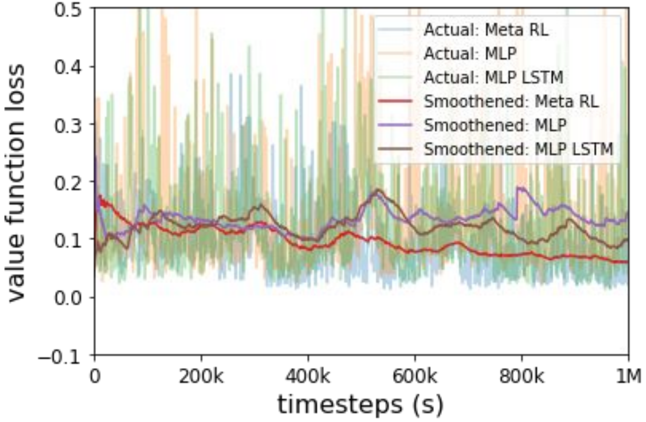
Results and Discussion

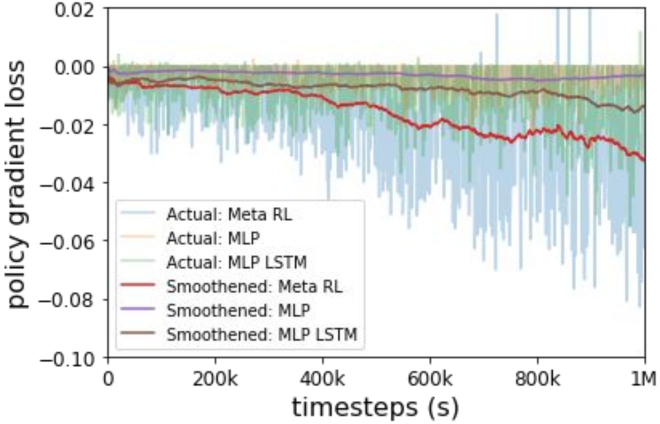
A2C





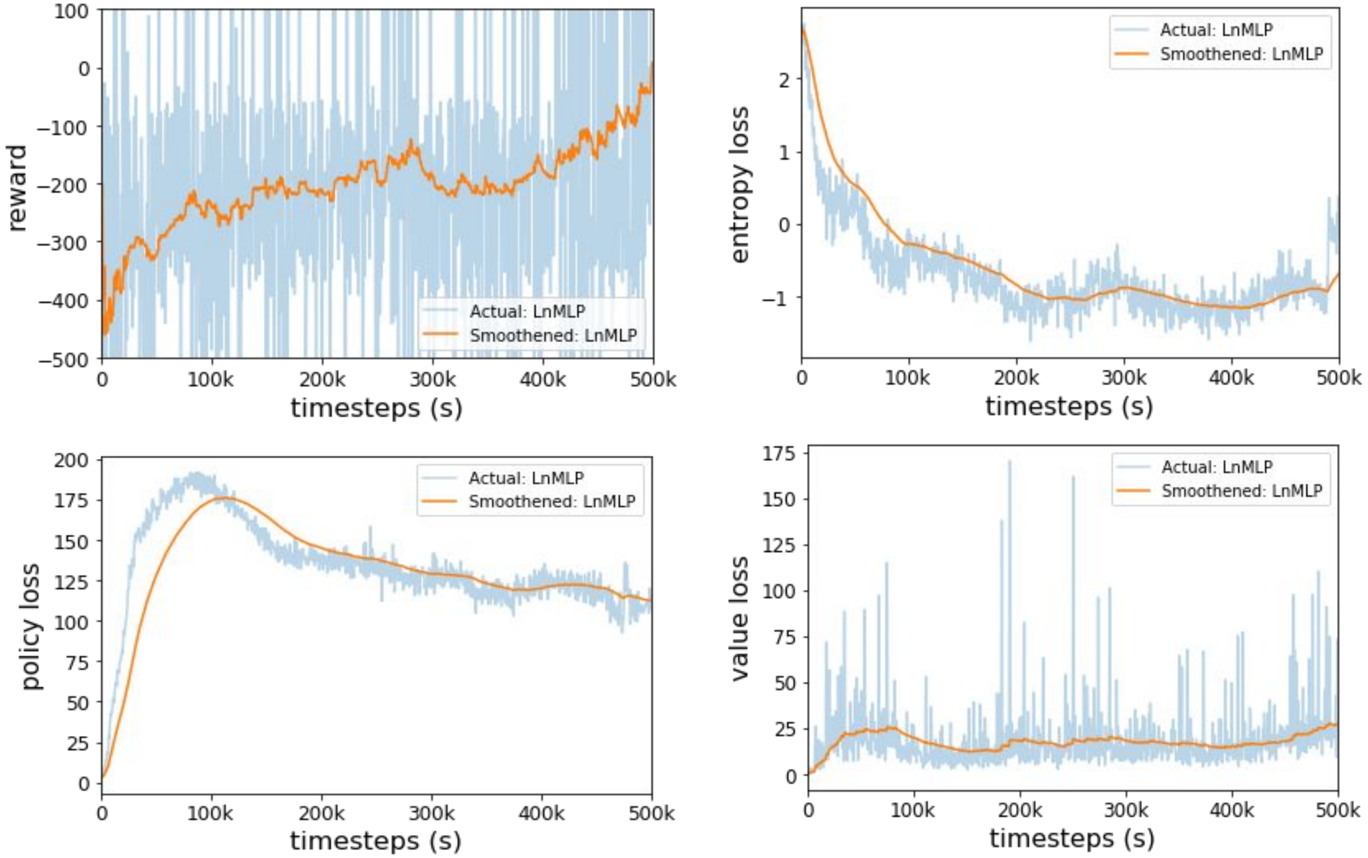
PPO2

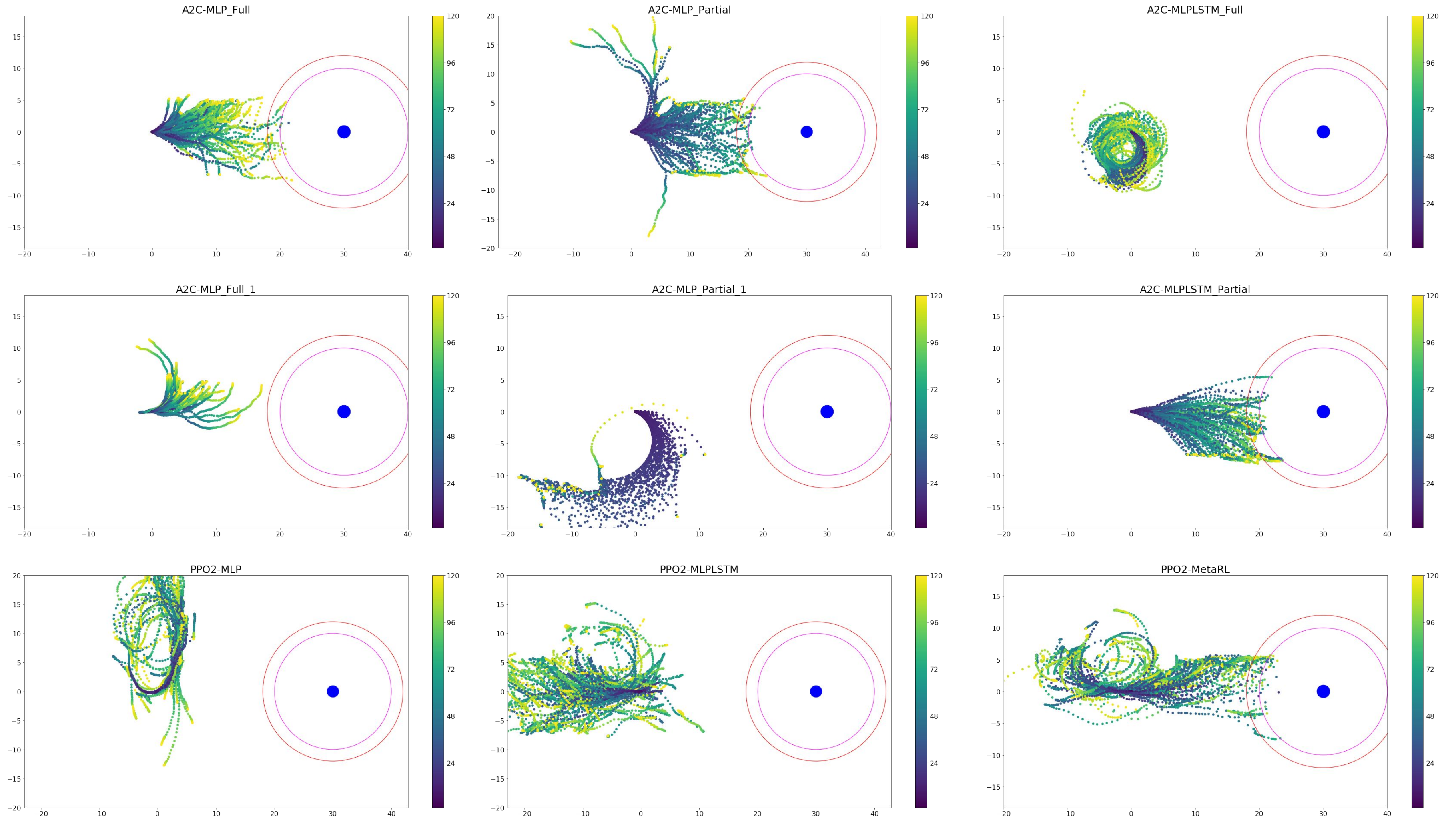


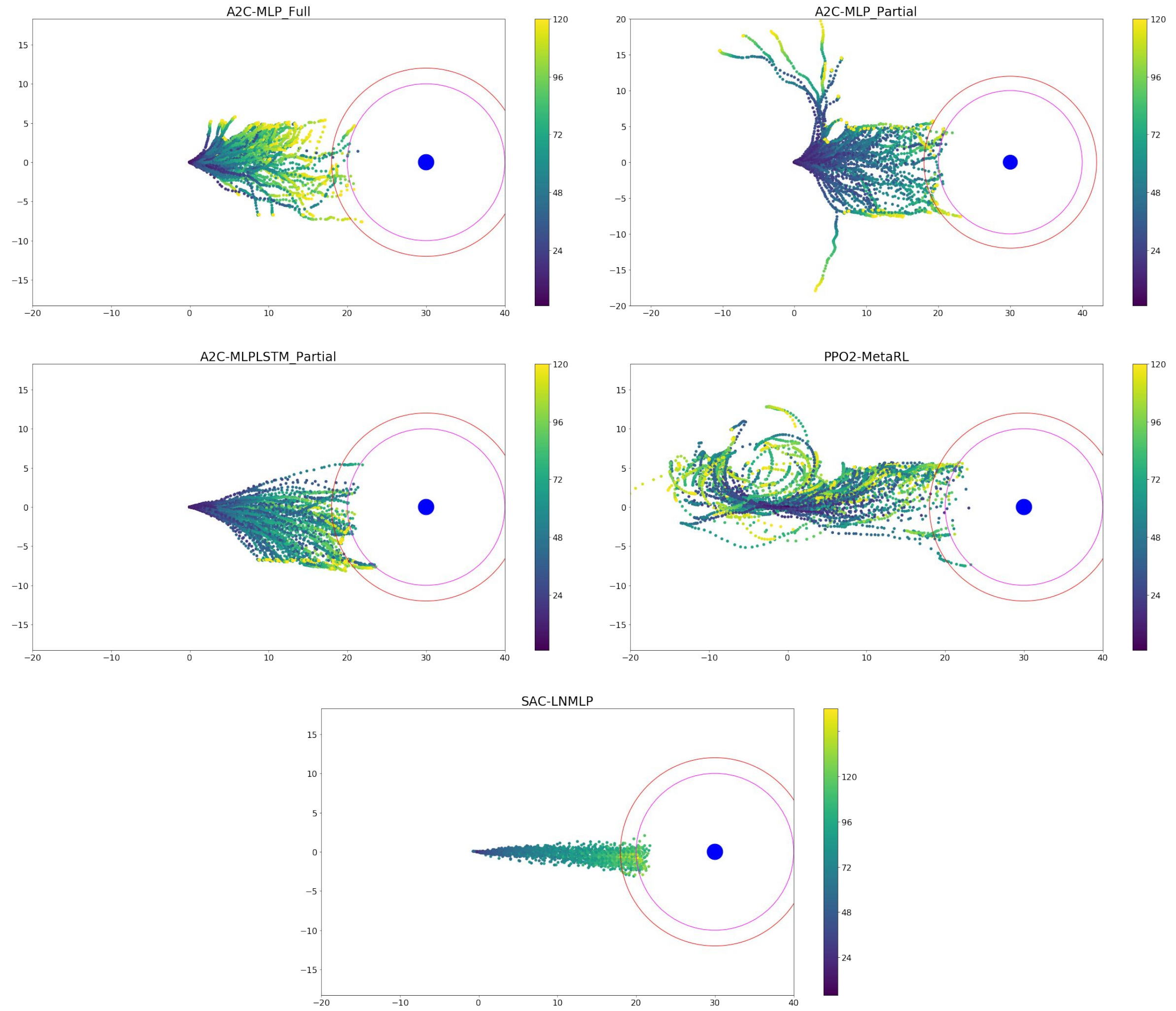


Results and Discussion

SAC




A2C-MLP_Full
A2C-MLP_Partial
A2C-MLPLSTM_Full
A2C-MLP_Full_1
A2C-MLP_Partial_1
A2C-MLPLSTM_Partial
PPO2-MLP
PPO2-MLPLSTM
PPO2-MetaRL


A2C-MLP_Full
A2C-MLP_Partial
A2C-MLPLSTM_Partial
PPO2-MetaRL
SAC-LNMLP
−20 −10 0 10 20 30 40
15 10 5 0 −5 −10 −15
24 48 72 96 120

Results and Discussion

	Algorithm-Policy	success rate	timesteps (mean)	total reward (mean)	collisions (mean)	trajectory distance (mean)	goal distance (mean)	goal distance (std)
0	***A2C-MLP_Full***	7%	117.84	408.16	22.64	21.17	19.14	4.95
1	***A2C-MLP_Full_1***	0%	120	-323.08	18.17	10.41	25.11	2.84
2	***A2C-MLPLSTM_Full***	0%	120	432.78	22.90	30.93	32.40	2.82
3	***A2C-MLP_Partial***	15%	112.48	192.32	52.12	21.68	18.92	7.42
4	***A2C-MLP_Partial_1***	0%	120	-1851.52	86.50	33.89	40.93	5.63
5	***A2C-MLPLSTM_Partial***	30%	104.98	239.40	32.36	28.29	13.15	3.67
6	***PPO2-MLP***	0%	120	-2.46	12.42	47.0	32.29	5.56
7	***PPO2-MLPLSTM***	0%	120	-3.12	11.44	34.29	44.69	10.60
8	***PPO2-MetaRL***	32%	105.73	-4.05	17.80	33.16	22.83	12.49
9	***SAC-LnMLP***	100%	28.67	938.42	0.01	23.21	9.46	0.40

Table 2: Performance evaluation during testing at 100 episodes.

Results and Discussion

- A2C used 4 action spaces vs PPO2 and SAC used 2 action spaces
  - steering, throttle, break, reverse
  - steering + movement (throttle, break, reverse)
- Both PPO2 and A2C has an increasing trend of entropy loss for MLP Policy Network
- Meta Reinforcement Learning hastens policy convergence compared to just using LSTM and MLP
- SAC learns relatively quickly compared to A2C and PPO2

Conclusion and Recommendations

- we've conducted a preliminary dive on Autonomous Driving using Deep Reinforcement Learning

Recommendations

- Continue training with tens of millions of time steps
- Further experiment with action space simplification like reducing conversion of range of movements (e.g. angle of steering)
- Add RGB camera view as part of observation space (CNN Policy Network).
- Effects of adding more (strategically placed) sensors
- Try to train first on a simpler map then transfer to more complex ones stabilized
- Explore other policy network architectures

Autonomous driving
via reinforcement learning

Roy Amante Salvador and Maria Isabel Saludares

CS 295 Reinforcement Learning | Project Presentation