RECURRENT CONTROL NEURAL NETWORK FOR THE MOUNTAIN CAR PROBLEM

YIWEI LI
RUOXIA QI
JINGCHENG WU

GROUP V RCNN

INTRODUCTION

Main Idea

- Schäfer et al. (2007)
- Based on RNN structure – dynamics simulator
- Extended by a control neural network – action selector
- Deal with high dimensional and continuous RL problems
- Applicable to multiple RL environments

INTRODUCTION

Research Questions

1. How well is the dynamics of the mountain car problem simulated?
2. Is the car able to reach the goal, following the policy learned by the RCNN?
3. How well does a successful policy perform?
4. Is RCNN a data efficient method?

DATA PREPARATION

Data Sampling

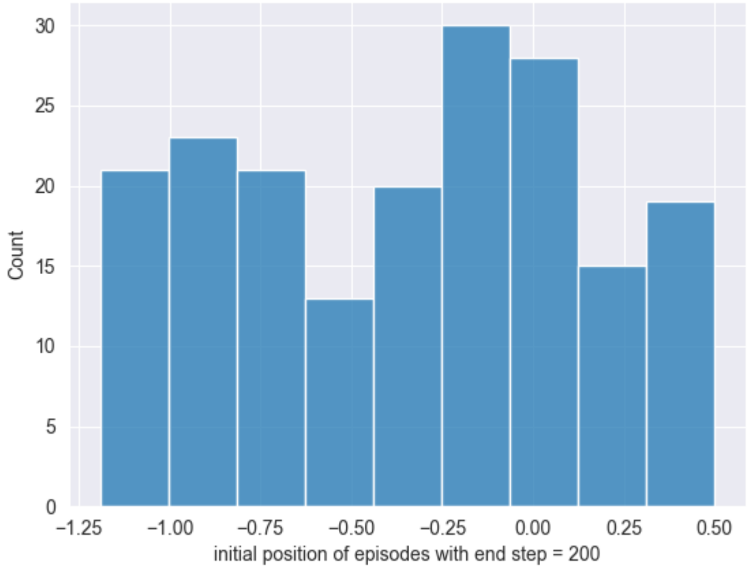
- 200 episodes starting from [-1.2, 0.6] with random selected actions (0, 1, 2)
- Reward = 1 / 0
- Split 7:1:2 into training, validation and test sets

DATA PREPARATION

Data Distribution

- Most episodes have sufficient observations
- Episodes with steps = 200 cover a wide range of initial positions
- Dataset covers sufficient patterns

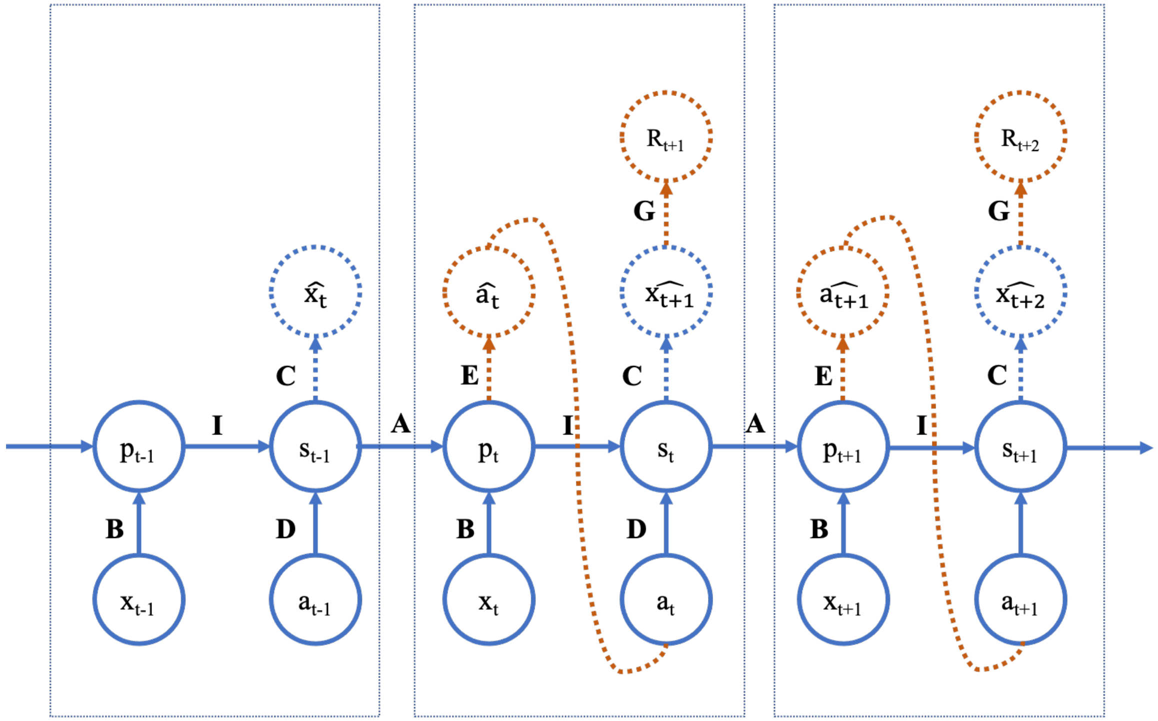
steps	# episode	initial position
1	4	0.515 - 0.6
2	1	0.53
4	3	0.521 - 0.579
5	1	0.556
11	1	0.558
200	190	-1.192 - 0.5



IMPLEMENTATION

Stage 1 - RNN

- Input: (pos, vel, action) -> (w, 3)
- RNN layer with customized RNN cell
- Dropout layer and Dense layer built upon
- Objective: MSE

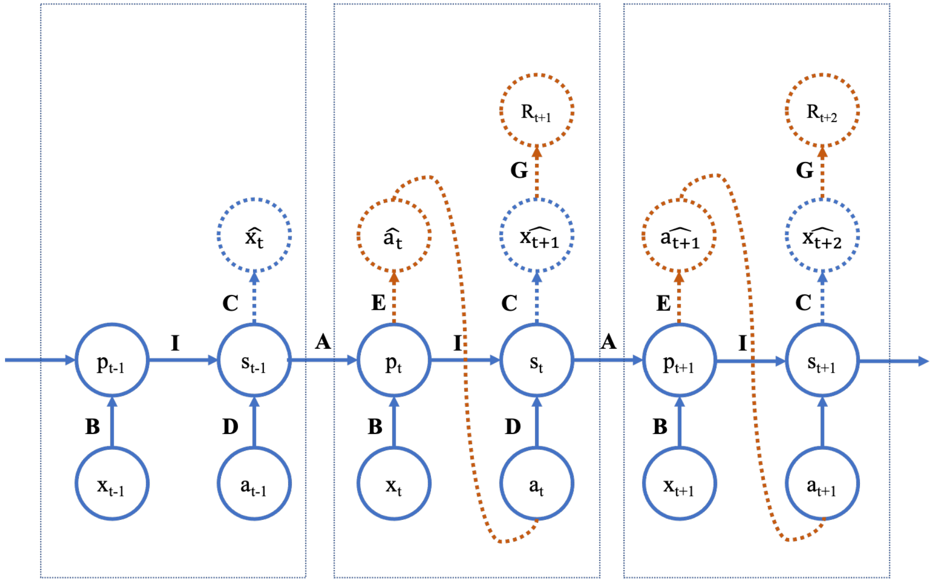


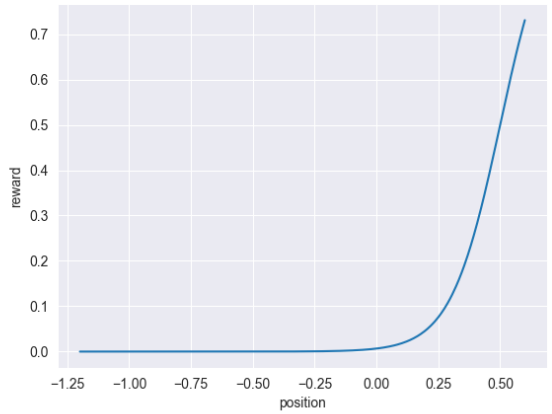
$$
\begin{aligned}
p_t &= As_{t-1} + Bx_t \\
s_t &= tanh(p_t + Da_t + \theta) \\
\hat{x}_{t+1} &= Cs_t + \tau,
\end{aligned}
$$

IMPLEMENTATION

Stage 2 - RCNN

- Input: (pos, vel, action) -> (w, 3)
- Copy RNN weights
- Build Dense layer to predict action
- Replace true action , calculate R
- Output next state and action



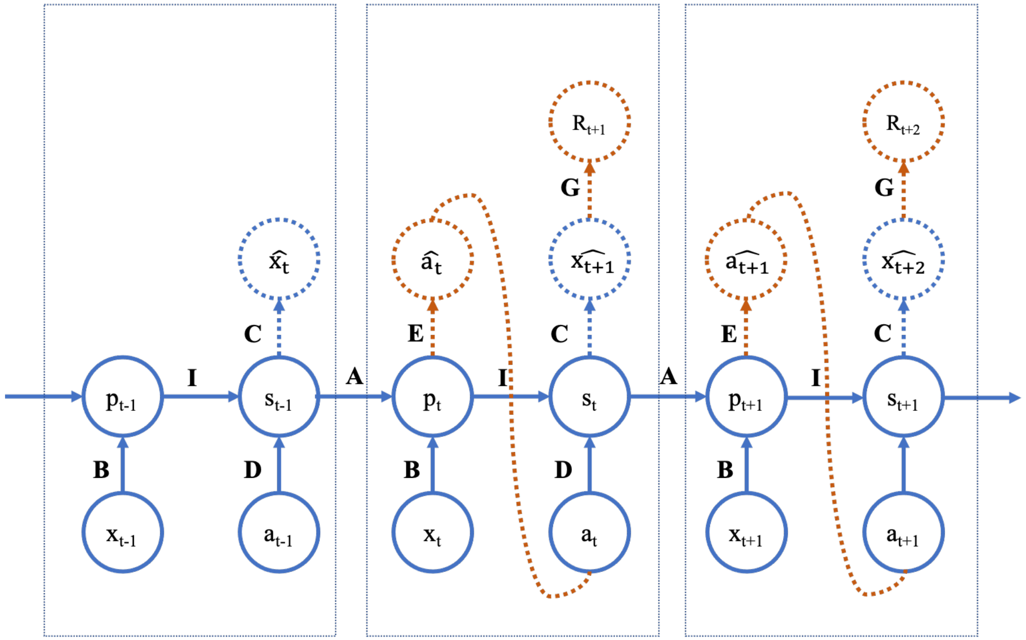


$$\hat{a}_t = tanh(Ep_t + b) \qquad R_t = \frac{1}{1 + \exp -10 * (o_t - 0.5)}$$

IMPLEMENTATION

General Setting

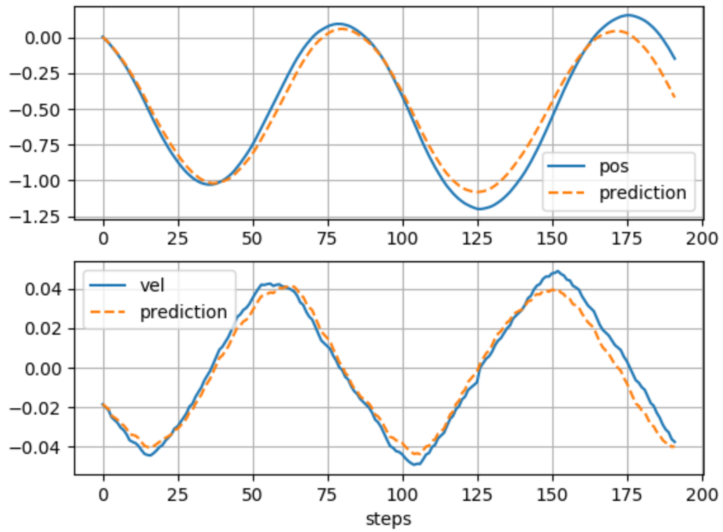
- Adam optimizer with lr = 10^{-3}
- lr reduced automatically
- Early stopping with patience = 20
- Train each model for 10 times

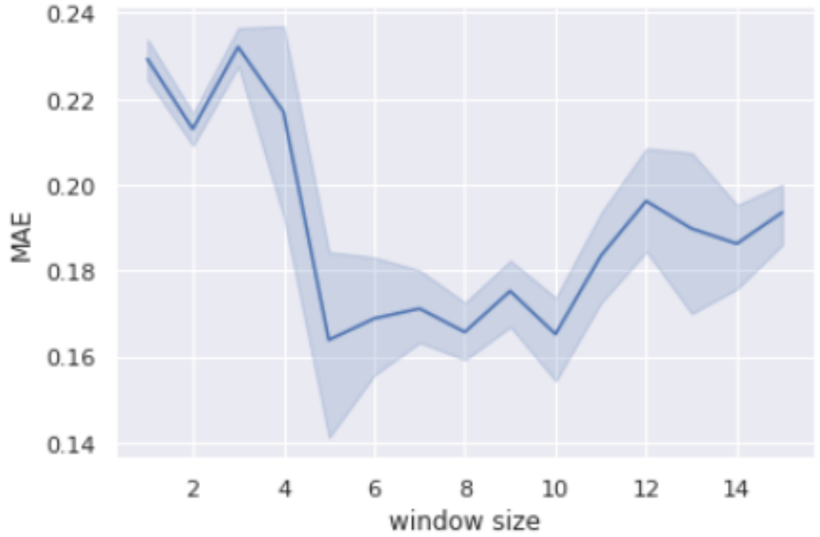


RESULTS

Dynamics Simulation (RNN)

- Test set: episodes with steps > w + 10
- Autoregressively
- MAE as objective measure
- Plotting as intuitive understanding
- Explore window size – MAE: similar behavior for w in 5-10

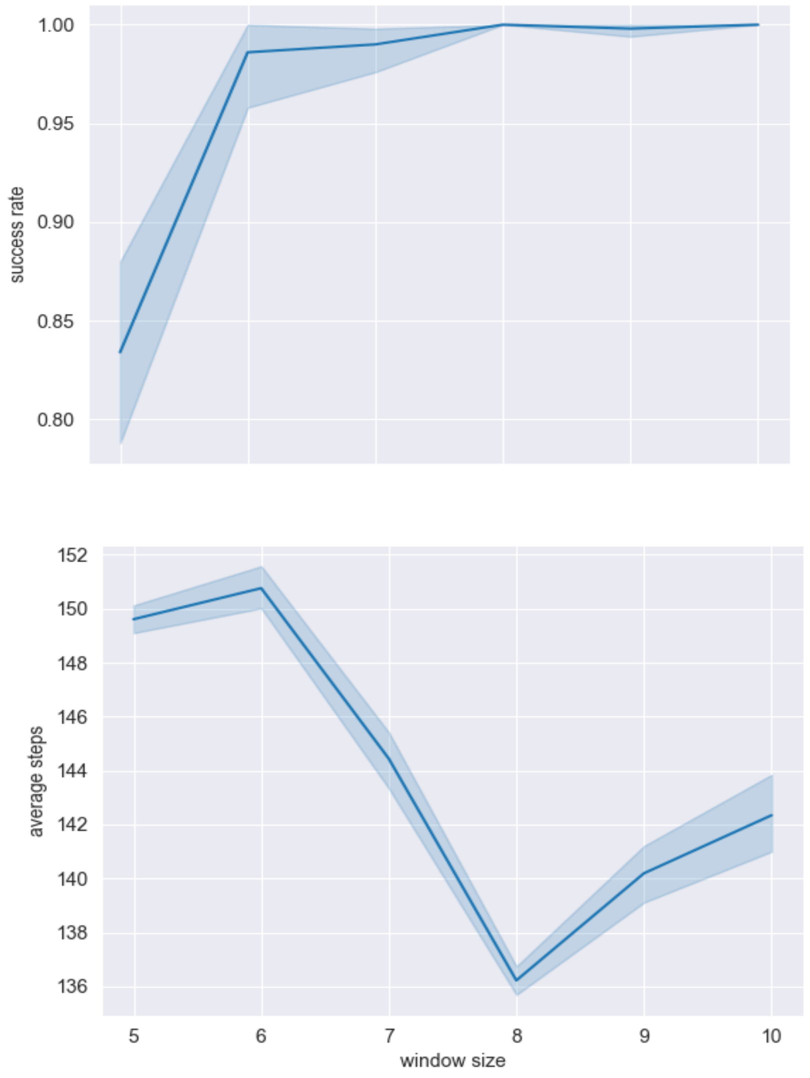




RESULTS

Policy Optimization (RCNN)

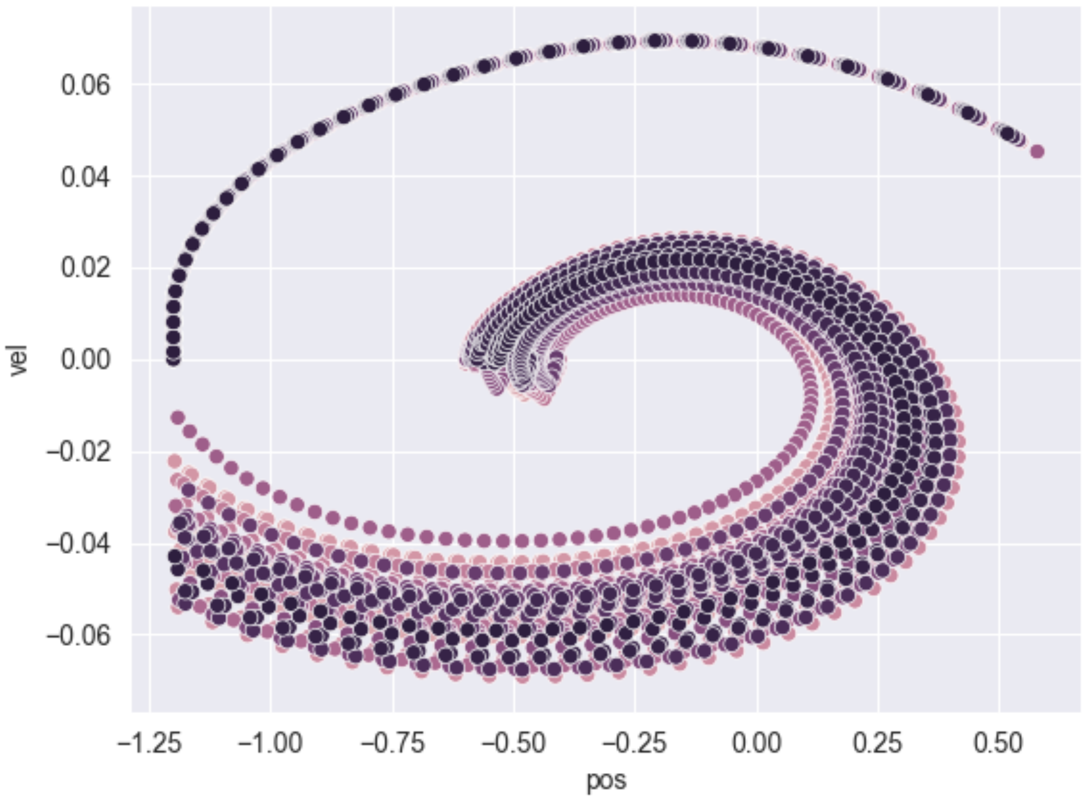
- Generate first window starting from [-0.6, -0.4]
- Autoregressively and repeat for 50 episodes
- Metrics:
  - success rate
  - average steps over successful trials
- Explore window size – performance with w = 5-10

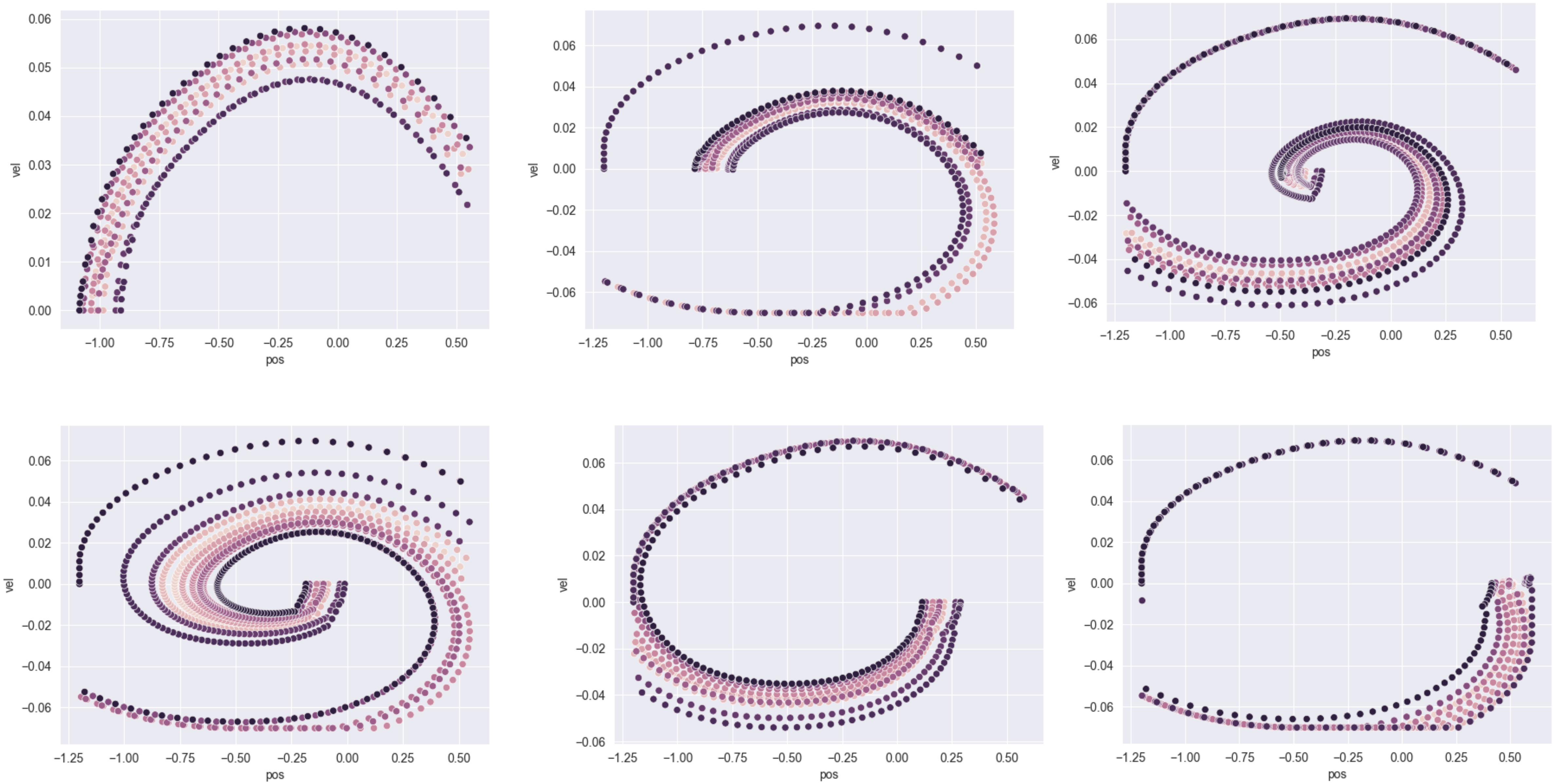


RESULTS

Policy Optimization (RCNN)

- Plot Trajectory starting from
  - the bottom
  - an arbitrary position

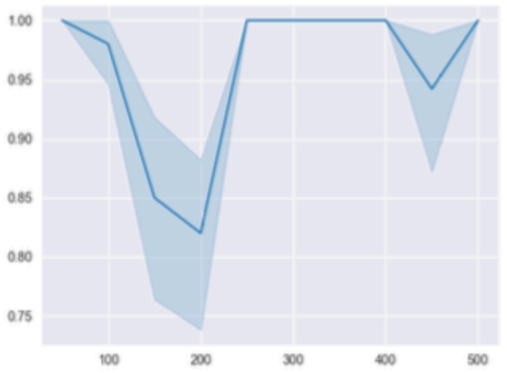



vel
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
pos
vel
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−1.25 −1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
pos
vel
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−1.25 −1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
pos
vel
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−1.25 −1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
pos
vel
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−1.25 −1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
pos
vel
0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
−1.25 −1.00 −0.75 −0.50 −0.25 0.00 0.25 0.50
pos

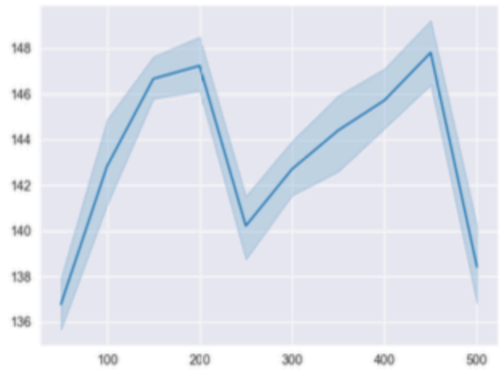
RESULTS

Policy Optimization (RCNN)

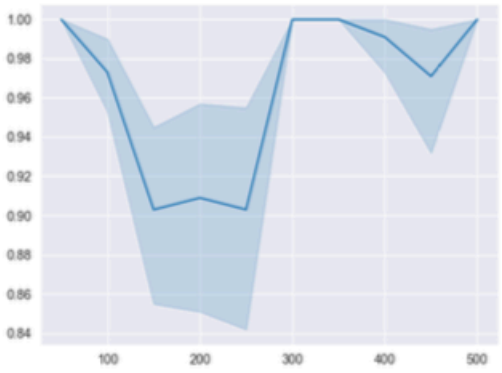
- Data amount – performance
- Unstable



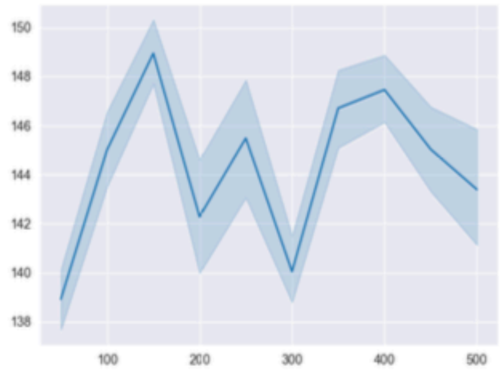
(a) size-rate (10 episodes)



(b) size-steps (10 episodes)



(c) size-rate (20 episodes)



(d) size-steps (20 episodes)

CONCLUSION

Research Questions

1. How well is the dynamics of the mountain car problem simulated? ✓
2. Is the car able to reach the goal, following the policy learned by the RCNN? ✓
3. How well does a successful policy perform? -- success rate = 1, average steps = 137
4. Is RCNN a data efficient method? --?

FUTURE WORK

1. Explore data amount and model performance
2. Compare our results to related works
3. Replace RNN by LSTM

THANK YOU