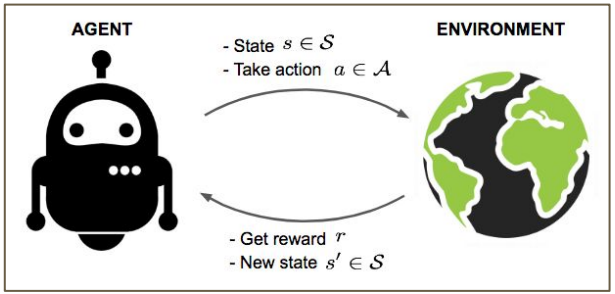
How to Make Reinforcement Learning Agents Learn Quicker?

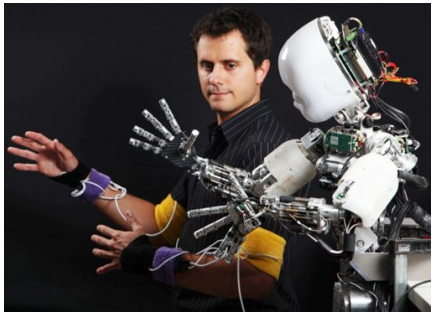
Team 7 - CS 7648
Vitaly V. Marin, Kritika Mittal, Sai Prasath

Georgia Institute of Technology

Motivation



Reinforcement Learning (RL)



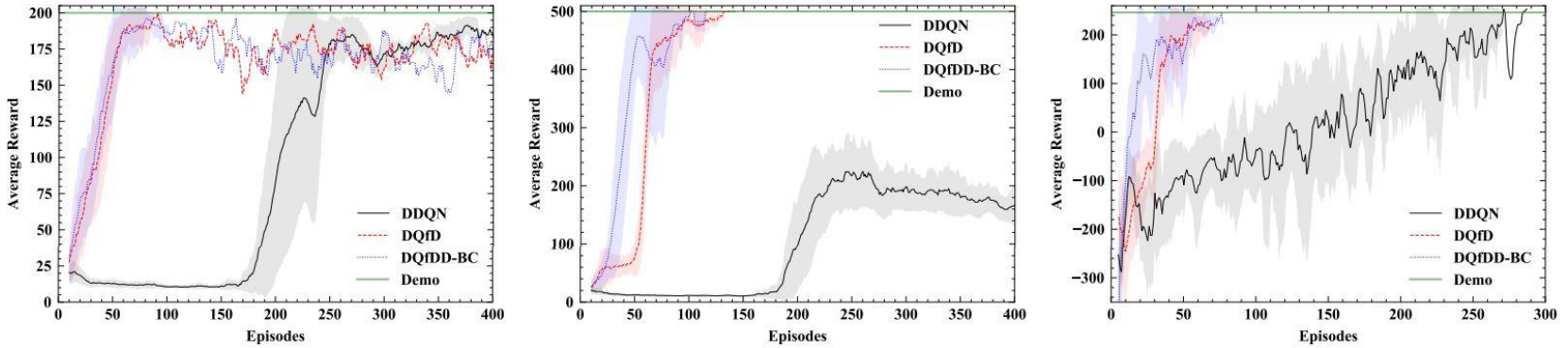
Learning from Demonstrations (LfD)

❌	Expert Demonstrations	✅
❌	Sample Efficiency	✅
❌	Quick?	✅
✅	Exploration	❌
✅	Better Policy?	❌

How to get the best of both RL and LfD?

Existing Literature

Bootstrap RL using demonstrations **before** online RL → Better Performance + Sample Efficiency [1][2]



(a) CartPole-v0 (b) CartPole-v1 (c) LunarLander-v2

Boost Initial Model Performance → Skip initial poor performance

Insufficiency in Literature

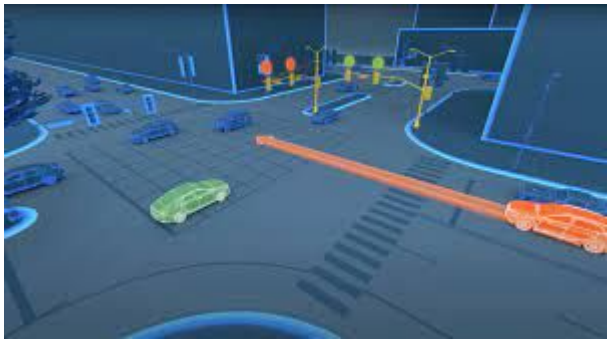
What to do with expert demonstrations after bootstrapping?

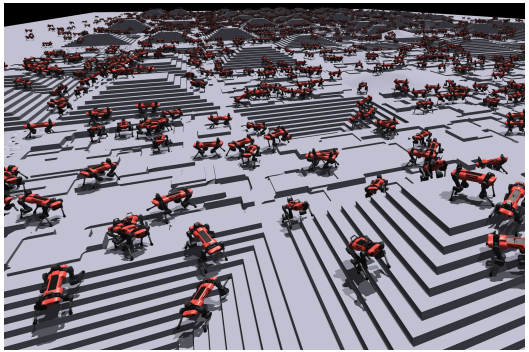
- Assign more priority to expert demonstrations over online RL trajectories [1]
- Sample 25% or 50% batch size from expert demonstrations for each model update [3][4]
- Analyze how the sampling rate from expert demonstration affects the performance on the model for a movie ticket booking task [5]

Why use these techniques? Are they generalizable across games and environments?

Why is it necessary?

- Building real world simulators is difficult and expensive
- **Requires RL agents to be deployed in real world → Poor Initial Performance → Bad Actions → Real consequences**
- Saves time and cost
- Building simulators >> Access to expert demonstrations

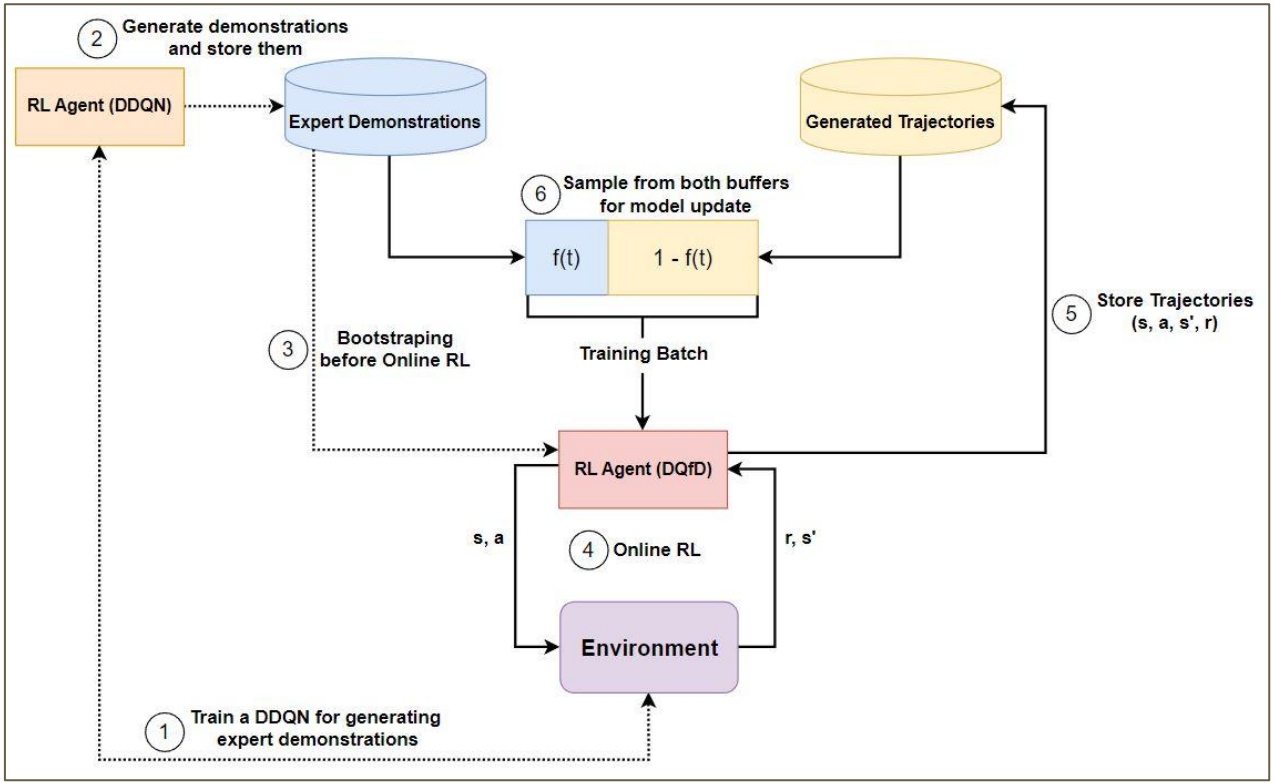




Problem Definition

In our project we will analyze the effects of different strategies for using expert demonstrations during online RL.

Model Architecture



Algorithm 1: Deep Q Learning from Demonstration

Input: D_{expert}: Initialized with expert demonstrations, $D_{generated}$: Empty, θ : weights for the model (random), θ' : weights for the target network (random), τ : target model update frequency, k : number of pretraining steps, m : number of training episodes, $f(.)$: sampling rate function

***Bootstrapping Phase*:**

for *steps* $t \in \{1, 2, 3, .., k\}$ **do**
- *Sample a mini-batch of size* n *from* D_{expert}
- *Calculate loss* $J(Q)$
- *Perform a gradient descent step on* θ
- ***if*** $t \bmod \tau = 0$ ***then*** $\theta' \longleftarrow \theta$ ***end if***

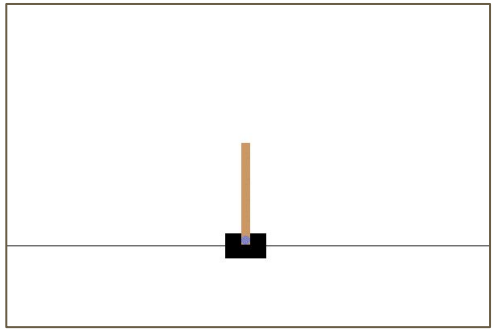
end

Online RL Phase:

for *steps* $t \in \{1, 2, 3, .., m\}$ **do**
- *Sample action* $a \sim \pi_{Q_\theta}$
- *Perform action* a *and observe* r *and* s'
- *Store* (s, a, s', r) *into* $D_{generated}$, *overwriting the oldest data store if exceeded capacity (FIFO)*
- *Sample a mini-batch of size* $n * f(t)$ *from* D_{expert}
- *Sample a mini-batch of size* $n - n * f(t)$ *from* $D_{generated}$
- *Calculate loss* $J(Q)$
- *Perform a gradient descent step on* θ
- ***if*** $t \bmod \tau = 0$ ***then*** $\theta' \longleftarrow \theta$ ***end if***
- $s \longleftarrow s'$

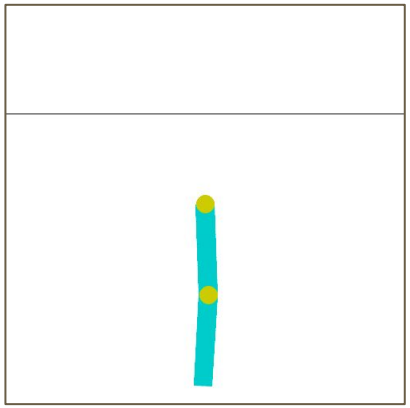
end

Environments [6]



Cart Pole

Action: {Left, Right}
State: {Position, Velocity, Pole Angle, Pole Angular Velocity}
Reward: +1 for each time step
Goal: Don't fall down

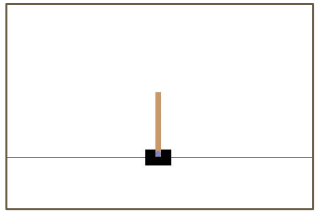


Acrobot

Action: {-1, 0, 1} Apply torque
State: {cos(t1), sin(t1), cos(t2), sin(t2), angular velocity t1, angular velocity t2}
Reward: -1 for each time step
Goal: Bottom end crosses black line

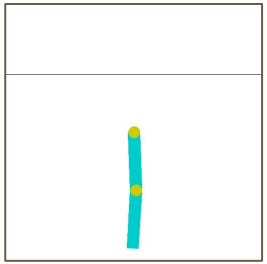
Experiment Setup

Step 1: Create different strategies to use expert demonstrations during online RL



Step 2: Test the strategies on the Cart Pole game

Step 3: Analyze the results and propose hypothesis about the behaviour of various strategies



Step 4: Validate the hypothesis by testing on the Acrobot game

Create different strategies

- **Linear Annealing:** In this technique we start the sample rate from a value $initial\_rate$ and reduce the value by $decay\_rate$ for each episode.

$$f(t) = max(0, initial\_rate - decay\_rate * t) \quad (8)$$

- **Exponential Annealing:** In this technique we start the sample rate from a value $initial\_rate$ and reduce the value exponentially at a $expo\_const$ rate.

$$f(t) = initial\_rate * e^{-expo\_const * t} \quad (9)$$

- **Threshold Based Annealing:** In this technique we use the sample rate from a value $initial\_rate$ if the average 10 episode reward is lesser than $anneal\_threshold$. Else we use a sample rate 0.

$$f(t) = \begin{cases} initial\_rate & \text{if 10-episode avg reward} \\ & < anneal\_threshold \\ 0 & \text{else} \end{cases} \quad (10)$$

- **Constant Sampling:** In this technique we maintain a constant sampling rate of $initial\_rate$ throughout the whole process.

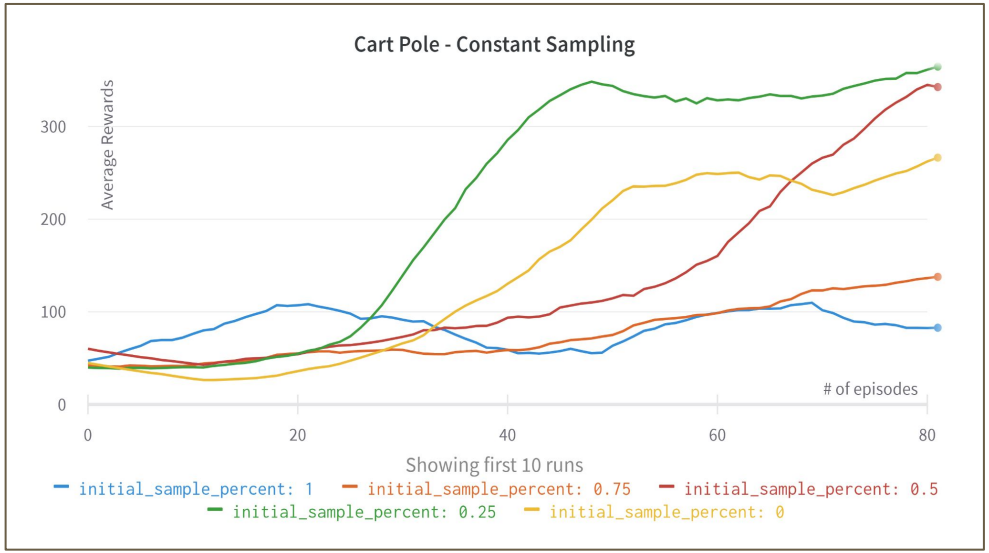
$$f(t) = initial\_rate \quad (11)$$

TABLE I
PARAMETER VALUES FOR CART POLE AND ACROBOT

	Initial Rate	Decay Rate	Expo Const	Anneal Threshold
Linear	0.25, 0.5	0.0025, 0.005, 0.01		
Exponential	0.25, 0.5		0.001, 0.01, 0.1	
Threshold Based	0.25, 0.5			C : 100, 200, 300 / A: -150, -200
Constant	0.0, 0.25, 0.5, 0.75, 1.0			

- Try different sets of parameter of all the annealing techniques
- Run 3 seeds to account for randomness in the results
- Analyze the 20-episode average reward to compare various strategies

Proposing Hypothesis 1

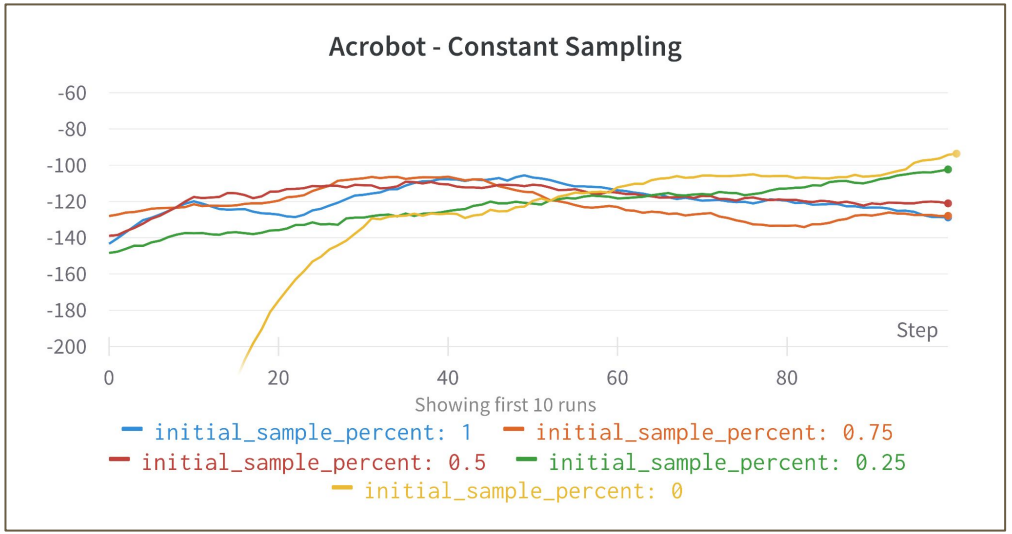


A low percentage of expert demonstrations, with larger space in the buffer for exploration, should be present during training to guide the RL agent to converge faster.

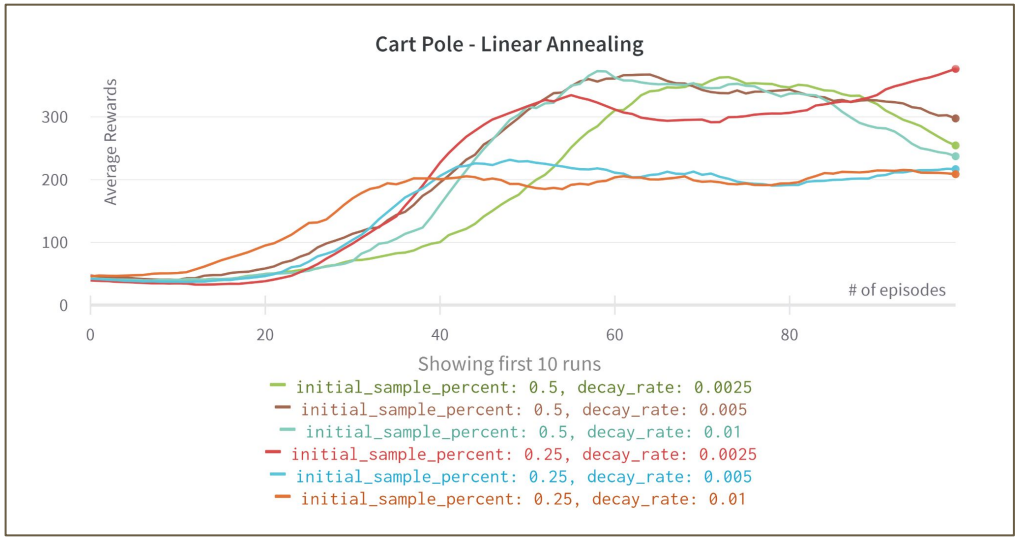
Validating Hypothesis 1

ACCEPT

A low percentage of expert demonstrations, with larger space in the buffer for exploration, should be present during training to guide the RL agent to converge faster.

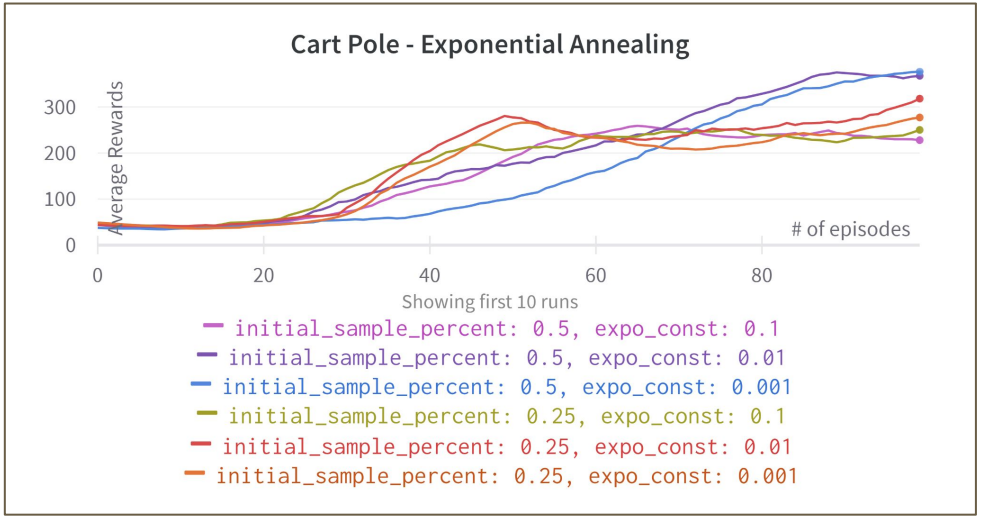


Proposing Hypothesis 2(a)



Using higher initial sampling rate (with decay) or a lower decay rate can help achieve better overall performance at the cost of slower initial learning rate.

Proposing Hypothesis 2(b)

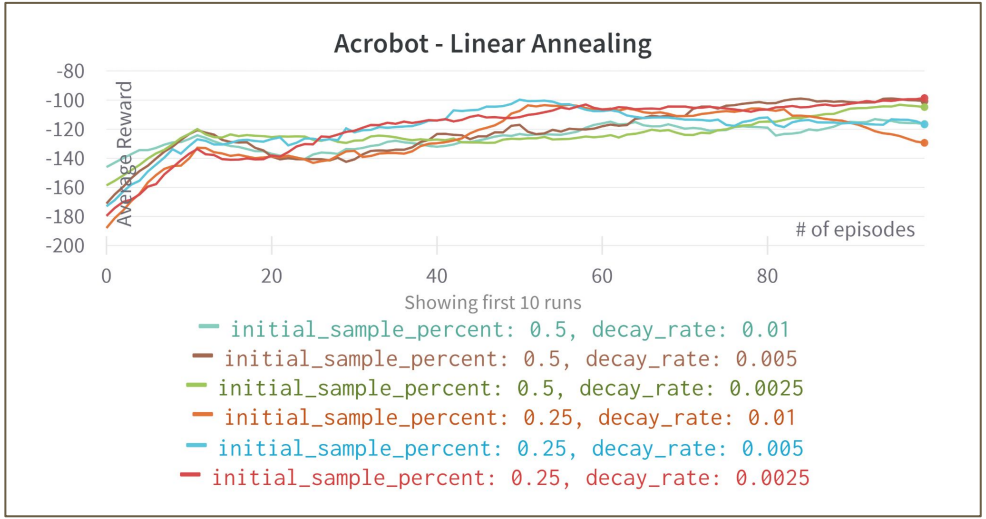


Using higher initial sampling rate (with decay) or a lower decay rate can help achieve better overall performance at the cost of slower initial learning rate.

Validating Hypothesis 2(a)

ACCEPT

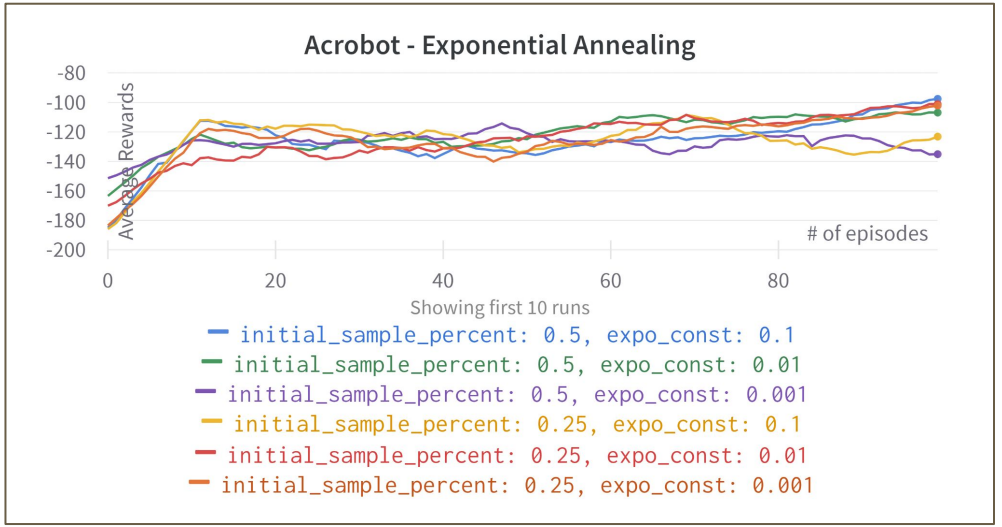
Using higher initial sampling rate (with decay) or a lower decay rate can help achieve better overall performance at the cost of slower initial learning rate.



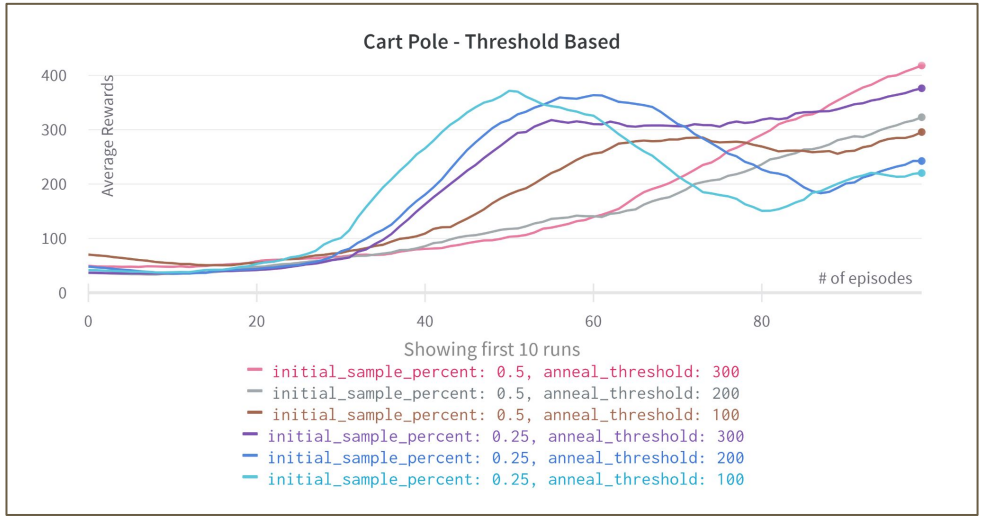
Validating Hypothesis 2(b)

REJECT

Using higher initial sampling rate (with decay) or a lower decay rate can help achieve better overall performance at the cost of slower initial learning rate.



Proposing Hypothesis 3

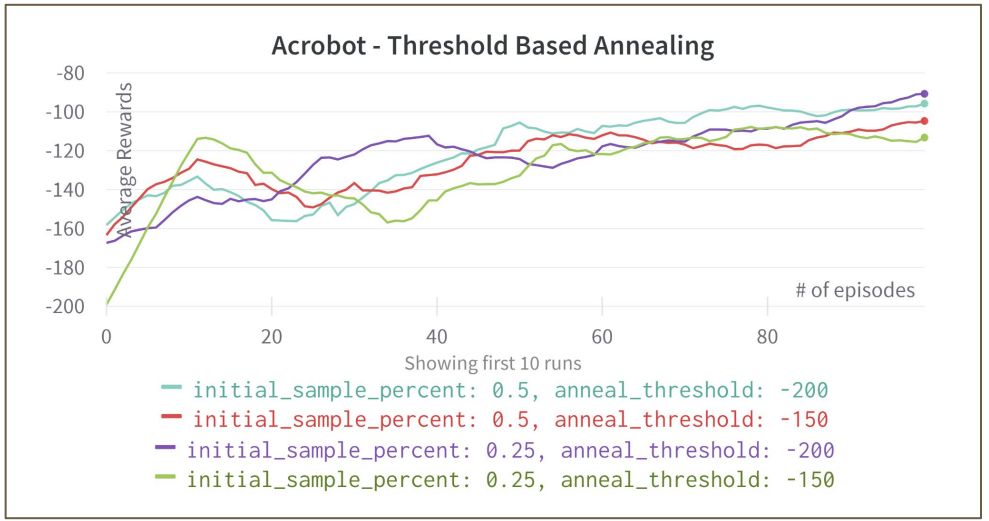


Suddenly dropping the sampling rate to 0% makes the model unstable and makes the model forget, before it re-learns again.

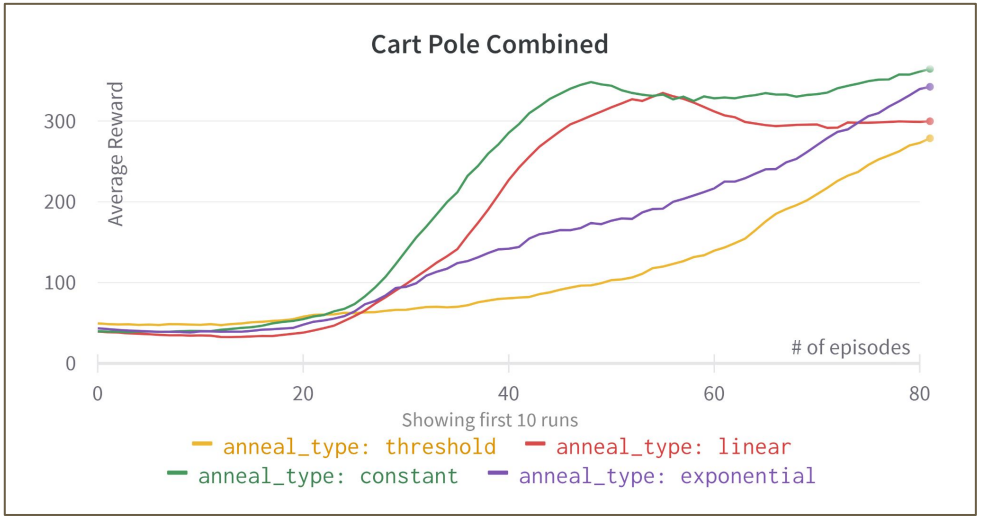
Validating Hypothesis 3

ACCEPT

Suddenly dropping the sampling rate to 0% makes the model unstable and makes the model forget, before it re-learns again.



Proposing Hypothesis 4

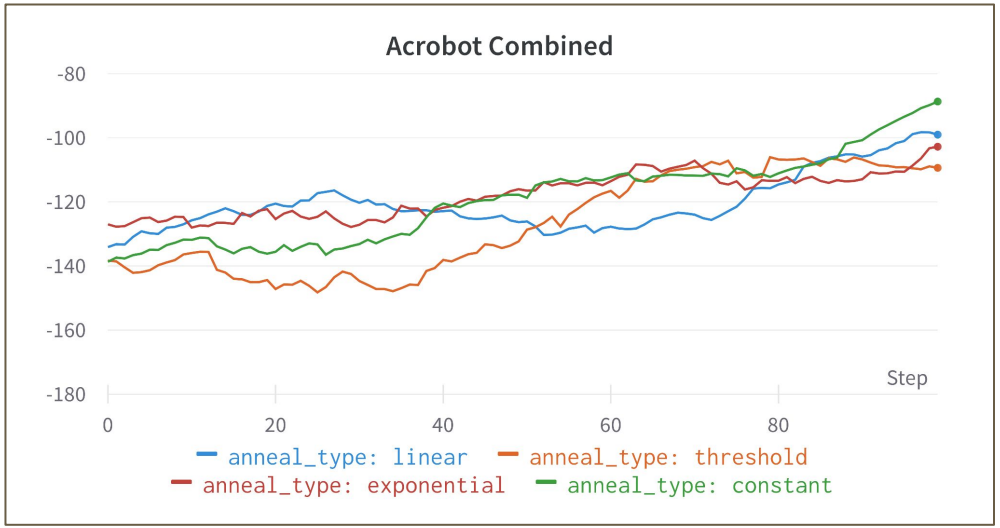


25% constant sampling is the best technique.

Validating Hypothesis 4

ACCEPT

25% constant sampling is the best technique.



Conclusion

- Initially expectation: The RL agent will not need any expert demonstrations after sufficient learning
- Observed behaviour: Forgetting problem in reinforcement learning → Constant sampling was the most stable
- **Small amount of expert demonstration in the buffer → Faster initial learning + Better overall performance**
- 25% constant sampling technique gave the best overall performance for the classic control environments, Cart Pole and Acrobot.

Limitations and Future Work

- Limited testing due to time and resource constraints.
- Results might not apply to games with continuous action spaces.
- Quality of expert demonstrations may affect the model’s performance.
- Testing on more games and environments for generalizability.

Got Questions ?
We Got Answers!

References

[1] Hester, Todd, et al. "Deep q-learning from demonstrations." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

[2] Li, Xiaoshuang, et al. "Deep Q Learning from Dynamic Demonstration with Behavioral Cloning." (2020).

[3] Hosu, Ionel-Alexandru, and Traian Rebedea. "Playing atari games with deep reinforcement learning and human checkpoint replay." *arXiv preprint arXiv:1607.05077* (2016).

[4] Pohlen, Tobias, et al. "Observe and look further: Achieving consistent performance on atari." *arXiv preprint arXiv:1805.11593* (2018).

[5] Lipton, Zachary, et al. "Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. No. 1. 2018.

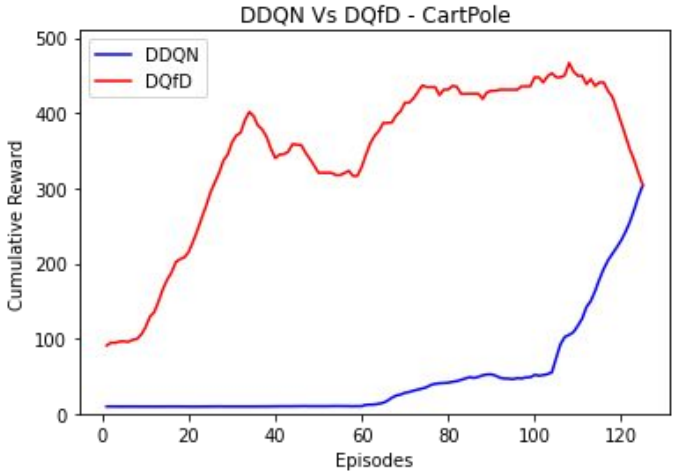
[6] Greg Brockman et al. “Openai gym”. In:arXiv preprint arXiv:1606.01540 (2016).

EXTRA SLIDES

Progress / Results

Baselines :

- **Double DQN** (DDQN) network
- **DQfD** network [1]
  - using 25% of the expert demonstrations



Environments:

- Classic Control Environment: Cart Pole
- MinAtar: Breakout
- Atari*: Breakout

[1] Hester, Todd, et al. "Deep q-learning from demonstrations." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

Proposed Approaches

Problem 1: How to use the demonstrations during online RL?
→ Test different strategies to utilize expert demonstrations: linear annealing, exponential annealing, threshold based annealing

Problem 2: Effects of the quality and quantity of expert demonstrations?
→ **Factors:** Initial learning, overall performance, time taken for convergence and sample efficiency
→ Tradeoff? Generalizable?

Experiment Setup:
→ Classic Control Environment [3] and MinAtar [4]

[3] https://gymnasium.farama.org/environments/classic_control/
[4] Young, Kenny and Tian, Tian (2019). “MinAtar: An Atari-Inspired Testbed for Thorough and Reproducible Reinforcement Learning Experiments.” ArXiv. https://arxiv.org/pdf/1903.03176.pdf

Motivation

- LfD Techniques: **Sample Efficient + Quick** + limited by quality of demonstrations
- RL Algorithms: Not sample efficient + Exploration takes time + **Good Policies**

Existing Literature

- Bootstrap RL using demonstrations → Better Performance + Sample Efficiency [1]

Problem 1: How to use the demonstrations during online RL?

→ Sample 25% during updates (Why?)

- How does quality and quantity of demonstrations affects the overall performance and initial learning?

Problem 2: Analyzed for a movie ticket booking task [2]

→ Is it generalizable?

[1] Hester, Todd, et al. "Deep q-learning from demonstrations." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.

[2] Lipton, Zachary, et al. "Bbq-networks: Efficient exploration in deep reinforcement learning for task-oriented dialogue systems." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 32. No. 1. 2018.