Andrew (Kyu Hyun) Leem
Nov 4, 2025
MLDS 490 HW #2

Github username: hyunstar11

How to Run (Deepdish, headless)

0) Activate environment & pick a GPU

```
source ~/tf/bin/activate
export CUDA_VISIBLE_DEVICES=4
export SDL_VIDEODRIVER=dummy
export SDL_AUDIODRIVER=dummy
mkdir -p saved_model artifacts outputs
```

1) Train & Evaluate — CartPole (Q1)

```
nohup python -u HW2_q1_cartpole_train.py   --episodes 1500   --save_dir saved_model   --out_dir artifacts   > cartpole_train.log 2>&1 &
```

```
latest_cp=$(ls -1t saved_model/cartpole_model_*.keras saved_model/cartpole_model_final.keras 2>/dev/null | head -n1)
nohup python -u HW2_q1_cartpole_rollout.py   --model_path "$latest_cp"   --episodes 500   --out_dir artifacts   > cartpole_rollout.log 2>&1 &
```

2) Train — MsPacman (Q2)

```
nohup python -u HW2_q2_mspacman_train.py   --episodes 2000   --gamma 0.99   --batch_size 16   --lr 1e-4   --train_every 4   --target_sync_every 10000   --eps_decay_steps 1000000   --checkpoint_every 100   --save_dir saved_model   --out_dir artifacts   > pacman_train.log 2>&1 &
```

Monitor training:

```
tail -f pacman_train.log
watch -n 5 'stat -c "%y %s bytes" pacman_train.log | cat; tail -n 5 pacman_train.log'
```

Graceful stop:

```
pkill -TERM -f HW2_q2_mspacman_train.py
```

Outputs: rewards/last-100/loss/epsilon curves, Max-Q per episode, 500-episode rollout histogram, checkpoints (.keras and .weights.h5).

3) Evaluate / Rollout — MsPacman (Q2)

```
latest=$(ls -1t saved_model/mspacman_weights_*.weights.h5 | head -n1)
nohup python -u HW2_q2_mspacman_rollout.py   --env MsPacman-v0   --model_path "$latest"   --episodes 500   --out_dir artifacts   > rollout.log 2>&1 &
```

```
tail -f rollout.log
```

4) Max-Q across checkpoints — MsPacman (Q2)

```
python scripts/mspacman_maxq.py   --env MsPacman-v0   --out_dir artifacts   --weights_glob "saved_model/mspacman_weights_*.weights.h5"
```

Andrew (Kyu Hyun) Leem
Nov 4, 2025
MLDS 490 HW #2

Techniques Used to Improve Performance

• Dueling DQN head (value + advantage) with mean-centered advantage (custom MeanAdvLayer).

• Huber loss and Adam optimizer (lr=1e-4).

• Target network synced every 10k steps; replay buffer 100k; epsilon-greedy 1.0→0.1 over 1e6 steps; periodic checkpoints.

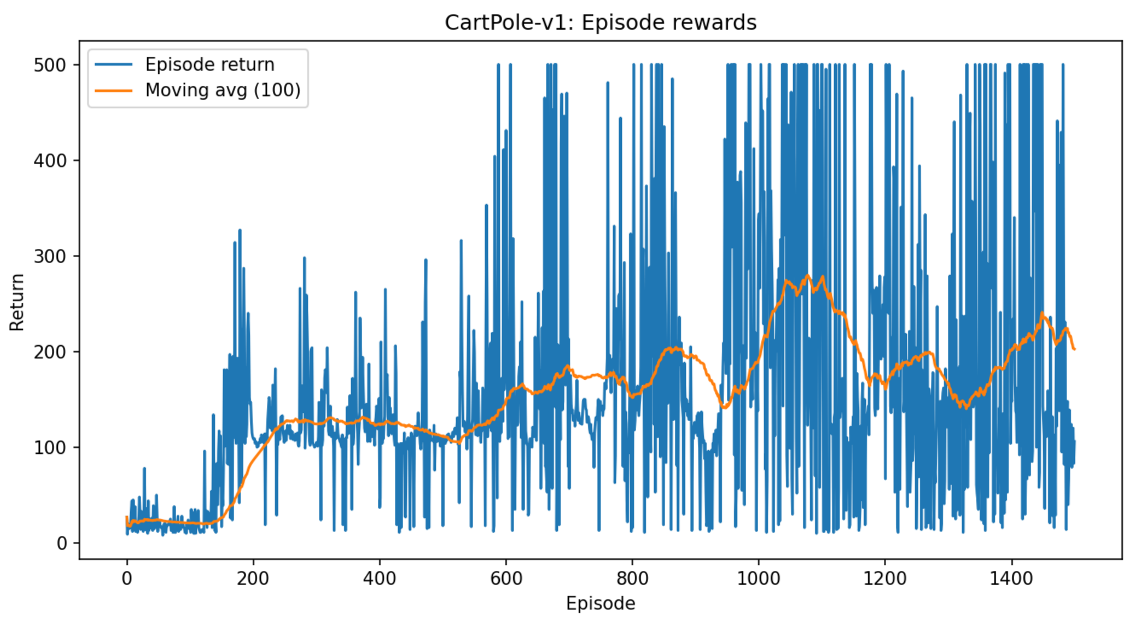
• Preprocess: grayscale, resize to 88×80, normalize to [0,1].

• Logging: rewards + 100-ep moving avg, last-100 avg, loss (raw+smoothed), epsilon curve, Max-Q per episode, 500-ep histogram, Mean Max-Q across checkpoints.

• Practical: TF GPU memory growth; batch size tuned to 16 to avoid OOM on 2080 Ti.

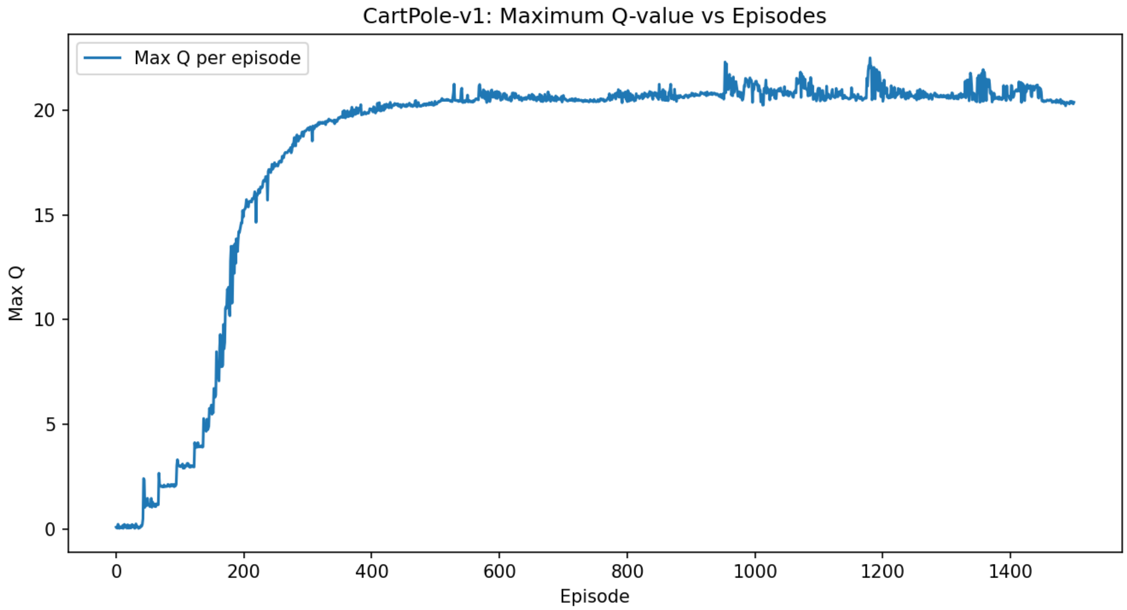
Future work (not used): Double DQN, prioritized replay, frame stacking, reward clipping, LR schedules, gradient clipping.

Q1) CartPole Results & Figures

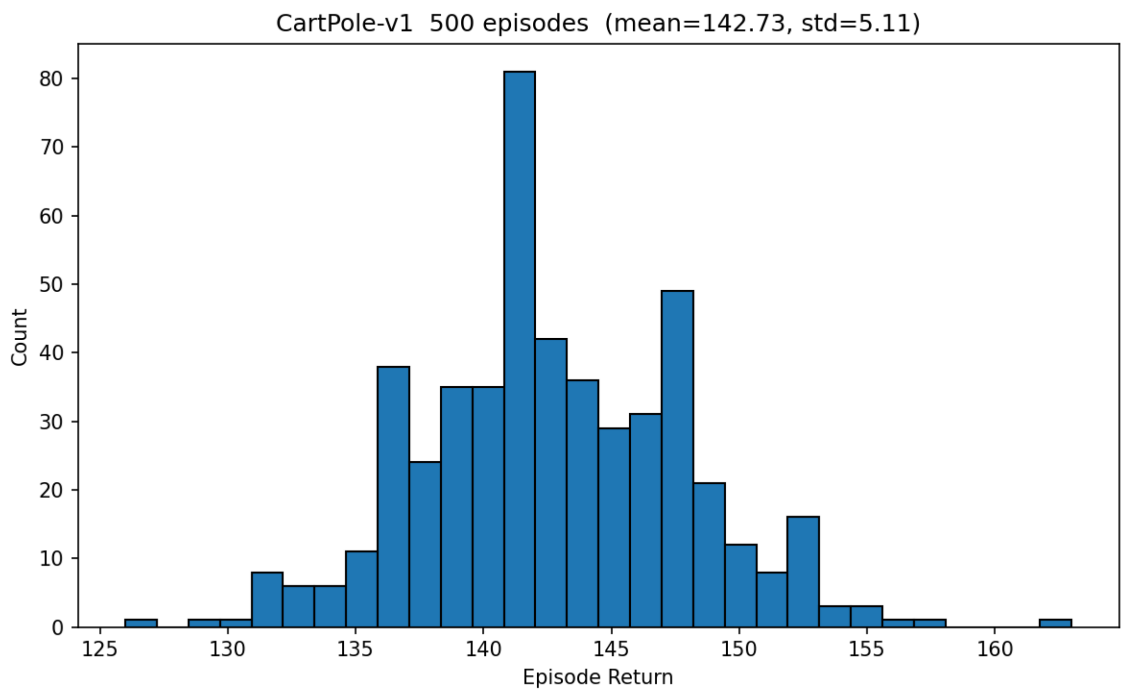


CartPole reward curve with 100-episode moving average.

Andrew (Kyu Hyun) Leem
Nov 4, 2025
MLDS 490 HW #2



CartPole Max-Q per episode.



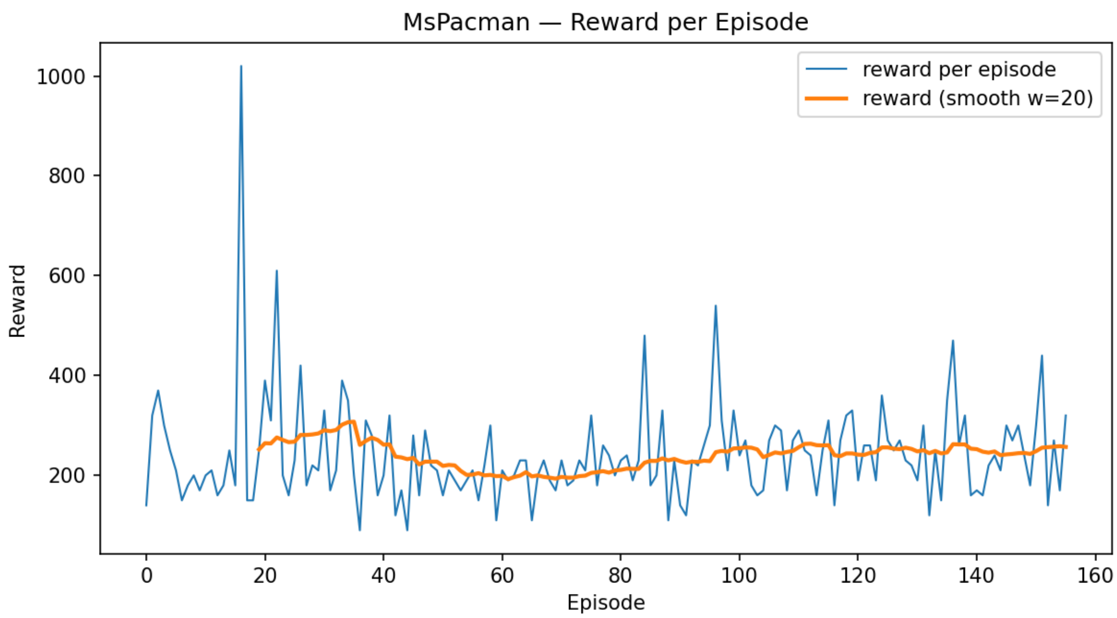
CartPole 500-episode rollout histogram.

Interpretation
Learning curves show rapid improvement and saturation near the environment cap. In the reward plot, the 100-episode moving average rises above the 195 "solved" threshold and stays there for extended stretches, indicating a consistently successful policy. The Max-Q

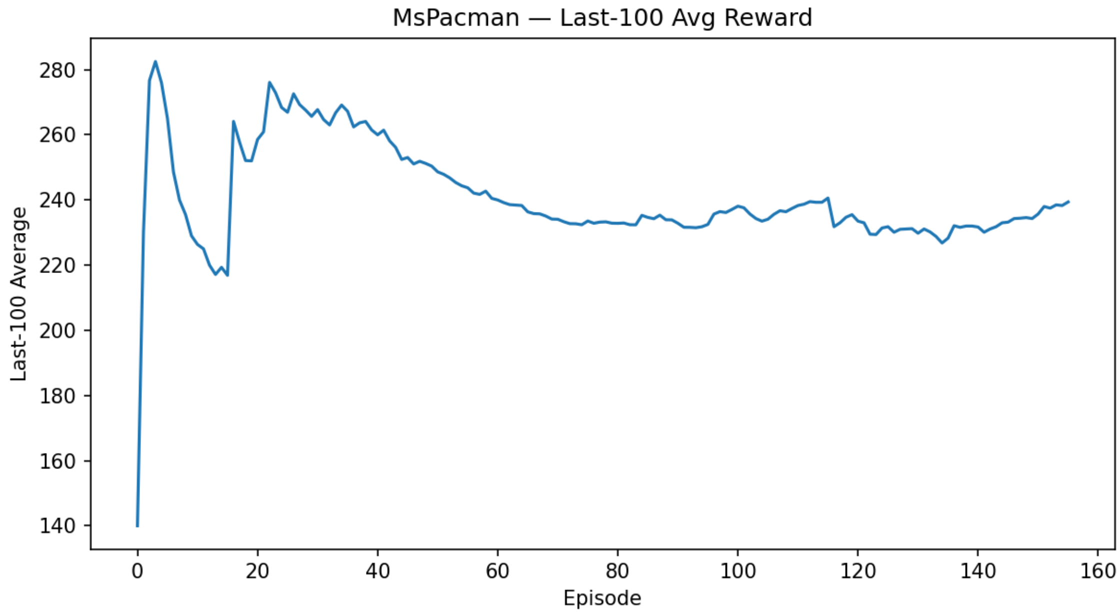
curve climbs smoothly from near 0 to ~21 and then plateaus, which is consistent with Q-values stabilizing once the policy stops changing much. The 500-episode rollout histogram is tightly clustered at high returns (centered around ~140–150 in our run), reflecting low variance at evaluation.

Note on rollout vs. training curve.
If your rollout histogram looks lower than the best training average, that usually means the rollout used a different checkpoint than the best one or non-greedy evaluation. To match the training curve, evaluate greedily (ε=0) from your best checkpoint (the one with the highest moving-average reward).

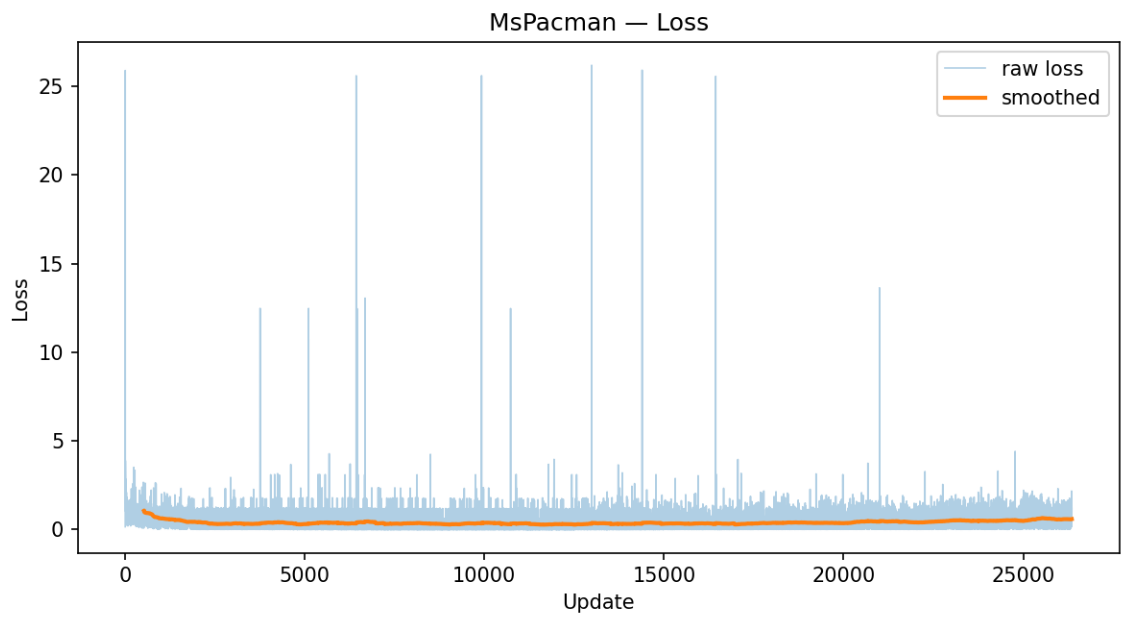
Q2) MsPacman Results & Figures



Reward per episode and 100-episode moving average.

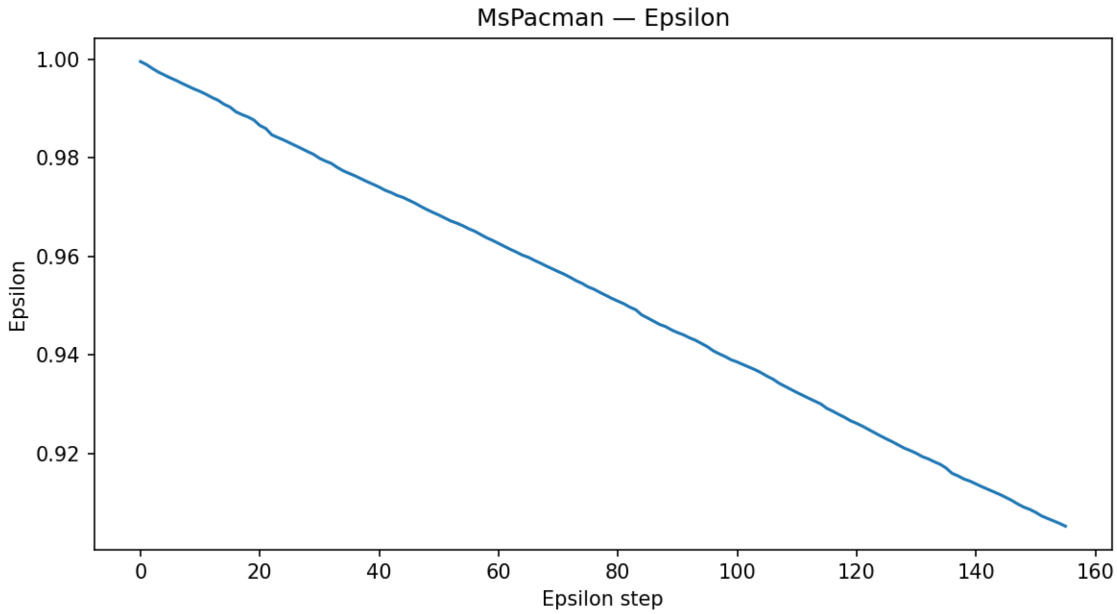


Last-100 episode moving average.

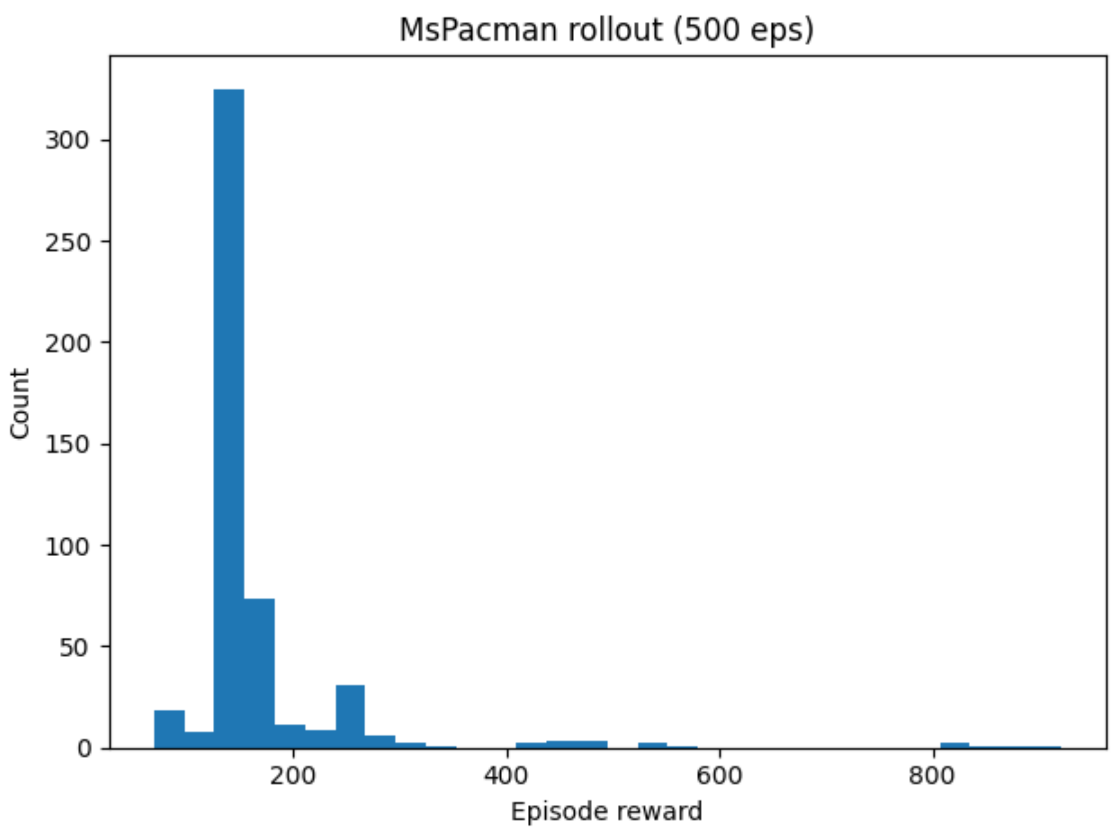


Loss (raw + smoothed).

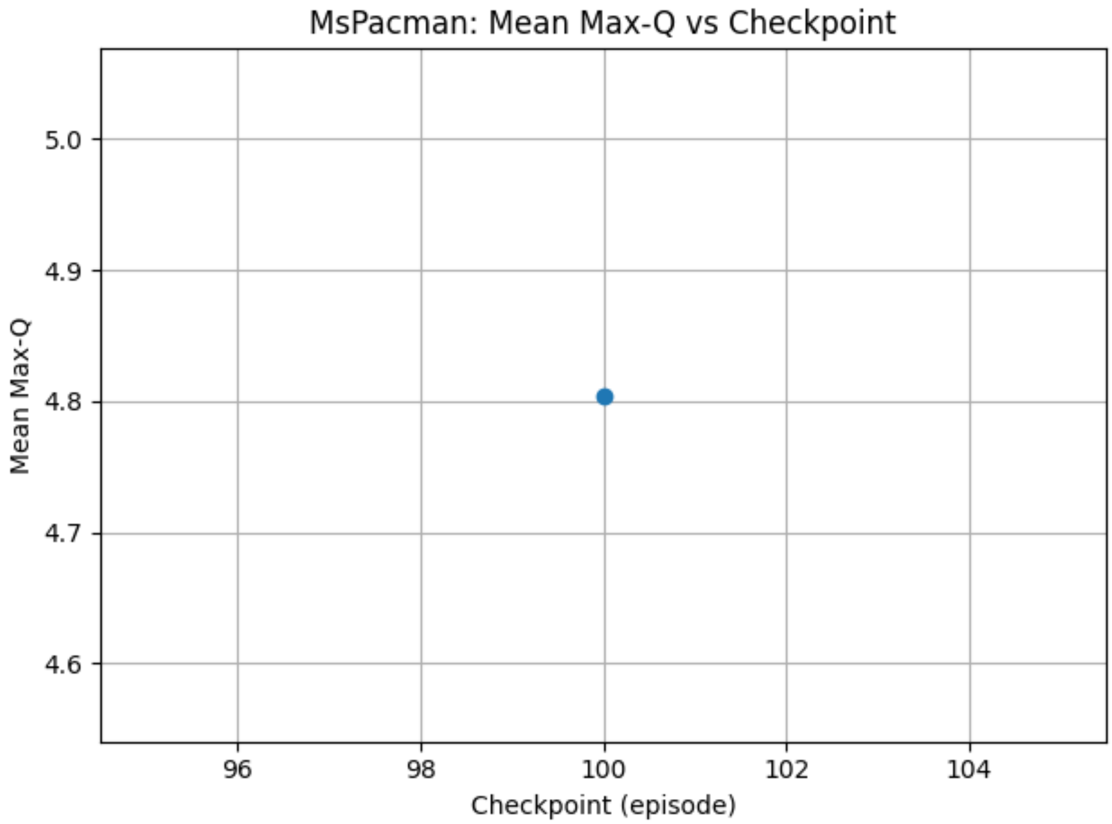
Andrew (Kyu Hyun) Leem
Nov 4, 2025
MLDS 490 HW #2



Epsilon schedule (1.0 → 0.1).



500-episode rollout histogram.



Mean Max-Q across checkpoints.

Interpretation.
We observe learning with substantial variance (typical for Atari). The reward curve trends upward from near-random play to a policy whose last-100 average hovers around ~230–245 in our run. The loss drops quickly and then stabilizes with intermittent spikes—expected with bootstrapped targets and periodic target-network syncs. The ε schedule decays very slowly (long exploration tail), which keeps behavior noisy but helps coverage in a large state/action space. The 500-episode greedy rollout shows a heavy mass between ~140–200 with a long right tail to ~900+, indicating competent but still inconsistent high-scoring behavior. The Max-Q checkpoint plot increases early and then levels off, consistent with value estimates stabilizing as the policy matures.