**Course/Section:** CSC 580 AI 2 Final Project

**Assignment Name:** Reinforcement Learning Analysis in Highway Environments

**Group:** 40 **| Names:** Om Prakash Gunja (2131025), Raju Meesala ( 2119844 )

**Word Count:**

**Final Project Report – Reinforcement Learning Analysis in Highway Environments**

**1. Introduction**

Autonomous driving relies on precise decision-making for tasks such as lane navigation, collision avoidance, and merging in dynamic traffic environments. Reinforcement Learning (RL) offers a powerful framework for training autonomous agents by enabling them to learn optimal driving policies through interaction with simulated environments.

This project investigates Deep Q-Network (DQN) variants within Stable-Baselines3 to optimize RL-driven autonomous navigation in Highway-Env. The primary focus is on developing a robust policy in “highway-fast-v0”, a high-speed, multi-lane driving scenario, and evaluating its transferability to “merge-v0”, a more complex environment requiring lane-changing and traffic adaptation. Additionally, we extend the evaluation to “roundabout-v0”, which presents unique challenges such as non-linear trajectories, dynamic negotiation, and multi agent interactions.

**2. Methodology**

**2.1 Environment**

We conducted our experiments in three primary environments provided by the **highway-env** library:

1. **Highway-Fast-v0:**

• Simplified version of highway driving with high-speed traffic.

• Allows the RL agent to learn fundamental driving maneuvers (lane following, speed control) and collision avoidance with fewer vehicles.

2. **Merge-v0:**

• More complex environment where the agent must merge onto a highway.

• Involves additional vehicles, varying speeds, and lane-change complexities.

• Challenges the agent to anticipate other cars’ behaviors and plan merges effectively.

3. **Roundabout-v0:**

• More complex environment where the agent must take round path onto a highway.

• Involves additional vehicles, varying speeds, and lane-change complexities.

• Challenges the agent to anticipate other cars’ behaviors and plan merges effectively.

**DQN Algorithm**

**3. Highway Environment Results**

**3.1 Baseline Model (DQN\_1)**

Baseline DQN Model: Hypothesis

Based on the baseline DQN model creation and training, the hypothesis is as follows:

* The baseline DQN model uses standard hyperparameters:
  + **Learning rate:** 5e-4
  + **Buffer size:** 15000
  + **Gamma:** 0.8
* It employs a basic neural network architecture with layers [256, 256].

**Hypothesis:**  
This configuration is expected to establish foundational performance metrics in the highway environment. However, due to the complex nature of multi-agent driving interactions, further hyperparameter tuning may be necessary for optimal results.

A graph on a screen

AI-generated content may be incorrect.

**A screenshot of a graph

AI-generated content may be incorrect.**

By analysing above graphs we believe these changes would yeild better training

1. Train Longer: Increase total\_timesteps from 2k to at least 20k, to make use of the bigger buffer zone and exploration

2. Adjust Gamma : Higher gamma might encourage longterm planning, which might be beneficial

3. Refine Exploration Fraction: If reward remains low after 70% of training, the agent may be spending too much time in a high exploration phase.

4. Higher Learning Rate Schedule: May help stabilize loss and improve late stage performance.

**3.2 DQN\_hyper1**

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

Observations

1. Episode Length (rollout/ep\_len\_mean)
   * New Run: ~15–17 (with fluctuations)
   * Old Run: ~11–12
   * Interpretation: Longer episodes may indicate improved survival or more complex (possibly indecisive) policies.
2. Episode Reward (rollout/ep\_rew\_mean)
   * New Run: Peaks around 13–14
   * Old Run: Peaked around ~9
   * Interpretation: Higher rewards suggest that the updated model is learning a better policy over a longer training horizon.
3. Exploration Rate (rollout/exploration\_rate)
   * New Run: Decays from 1.0 to ~0.05 over ~14k steps
   * Old Run: Quickly decayed within 2k steps
   * Interpretation: Extended training supports prolonged exploration followed by effective exploitation, contributing to higher rewards.
4. Time/FPS (time/fps)
   * Both runs remain stable at ~57 FPS, indicating no performance bottleneck.
5. Learning Rate (train/learning\_rate)
   * New Run: Around 0.01 (or following a schedule toward that)
   * Old Run: Constant at 5e-4
   * Interpretation: A jump to 1e-2 can be aggressive and may explain the more volatile loss; a proper decay schedule is critical.
6. Training Loss (train/loss)
   * New Run: Averages around 0.98+ with larger spikes
   * Old Run: Fluctuated around 0.18–0.4
   * Interpretation: Despite higher loss spikes (likely due to the higher learning rate), the better rewards indicate the agent is learning, albeit in a more turbulent manner.

**3.3 DQN\_hyper2**

In the next tuning phase, we will implement several adjustments based on our observations to improve the agent's performance and learning stability.

* **Expand Network Architecture to [512, 256, 128]:**  
  We are increasing the network size to capture more complex patterns, which should lead to a more robust policy. This adjustment is aimed at enhancing the model’s capacity to understand intricate features in the data, even though it may require more data and training time.
* **Set Learning Rate to 1e-3:**  
  By choosing a moderate learning rate, we aim to speed up early learning without introducing the instability seen with higher rates. This setting is intended to provide a good balance between learning speed and stability.
* **Adjust Buffer Size to 10,000:**  
  A buffer of this size will allow us to store a diverse range of experiences while still ensuring that fresh data is frequently sampled. This should help maintain a balance between learning from recent events and preserving a variety of past experiences.
* **Reduce Batch Size to 64:**  
  A smaller batch size is expected to stabilize gradient updates and reduce noise during training. This change will help the agent learn more consistently, even if each training step might take slightly longer.
* **Increase Gamma to 0.95:**  
  Raising gamma will place more emphasis on future rewards, which is important for tasks where long-term planning is crucial. This adjustment is designed to help the agent better anticipate and act on future outcomes.
* **Update Target Network Every 10 Steps:**  
  More frequent updates of the target network will keep it closely aligned with the online network. This is expected to improve learning stability by ensuring that the target estimates remain relevant to the current policy.
* **Maintain Exploration Fraction at 0.7:**  
  Keeping the exploration fraction high during the early training phases encourages the agent to try a wide range of strategies. This approach is intended to help the agent discover more effective behaviors before settling on a more exploitative strategy.

A graph on a white background

AI-generated content may be incorrect.

**A screenshot of a graph

AI-generated content may be incorrect.**

Observations

1. Better Exploration-Exploitation Balance • With a more moderate learning rate, the agent doesn’t overwrite its Q-function as aggressively, allowing it to gradually refine a strong policy once it has gathered enough experiences.
2. Capacity to Represent Complex States • Highway driving can involve multiple vehicles, lane changes, and merges. A bigger network (hyper2) can learn more nuanced value estimates, leading to more optimal decisions over time.
3. Extended Training • Both hyper1 and hyper2 ran for ~20k steps (unlike the 2k-step baseline), but hyper2’s architecture and LR combination leveraged that extended training more effectively, reaching a higher reward peak.

**3.4 DQN\_hyper3**

**1. Learning Rate:**

* **Hyper2:** 1e-3
* **Hyper3:** 1e-2
* Increased to potentially speed up learning. While a lower learning rate (like in hyper2) helps stabilize training, increasing it in hyper3 might have aimed to explore the benefits of quicker adaptation, especially with a larger dataset or more complex environment.

**2. Exploration Fraction:**

* **Hyper2:** 0.7
* **Hyper3:** 0.6
* Decreased to reduce exploration and exploit learned behavior sooner. By lowering the exploration fraction, hyper3 potentially shifts the agent's strategy towards exploiting what it has already learned, leading to potentially faster convergence and more stable performance.

**3. Gamma:**

* **Hyper2:** 0.95
* **Hyper3:** 0.8
* Decreased to focus more on immediate rewards potentially better suited for highway task. A higher gamma value (like in hyper2) emphasizes future rewards, which is important for tasks requiring long-term planning. Lowering it in hyper3 suggests a focus on immediate rewards, which might have been deemed more relevant for the specific characteristics of the highway task.

**A screenshot of a graph

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

Observations

• Stable Final Performance: Hyper3 ended with a higher and more consistent final reward (~20+) compared to hyper2, which, despite its higher peak, dropped off later. A stable policy is essential for reliable performance, especially when transferring to a new environment like merge.

• Consistent Episode Behavior: The parameter settings in hyper3 (e.g., lower exploration fraction and gamma at 0.8) resulted in longer, more consistent episodes, suggesting that the agent’s decisions are robust. This is often more valuable than sporadic peaks in reward.

• Immediate Reward Focus: With a gamma of 0.8, hyper3 might be better tuned for environments where immediate actions have a significant impact, providing a solid baseline for further fine-tuning in the merge or roundabout tasks.

• Potential for Further Tuning: A stable model like hyper3 gives you a reliable starting point. You can experiment with slight adjustments (like tweaking the learning rate schedule or exploration settings) and be more confident that changes are improving a solid baseline, rather than trying to stabilize a model that already shows signs of volatility.

**4. Transfer Learning to Merge-v0**

**4.1 Rationale for Transfer**

The **merge-v0** environment introduces lane-change and merging complexities. Rather than training from scratch, we aimed to leverage a policy pre-trained on highway-fast-v0, hypothesizing that the learned features (e.g., collision avoidance, speed control) would transfer and reduce training time.

**4.2 Selecting a Model for Transfer**

While **DQN\_hyper2** had the highest peak reward, it exhibited a decline in the later stages. **DQN\_hyper3** ended with a more consistent performance (~20+). For a stable baseline, we opted to **copy the weights from DQN\_hyper3** into the merge environment.

**4.3 Implementation Details**

A screenshot of a computer program

AI-generated content may be incorrect.

**5. Merge Environment Results**

**5.1 Initial Performance**

Upon loading **DQN\_hyper3** weights, the agent started in the merge environment with a baseline policy already familiar with highway dynamics. Early observations showed:

• **Base Model**: Loads the DQN model trained on highway-v0 (dqn\_highway\_hyper3.zip).

• **Hyperparameter Adjustments**:

• learning\_rate = 1e-3: Lower learning rate for fine-tuning.

• gamma = 0.7: Slightly higher discount factor for long-term rewards.

• exploration\_fraction = 0.5: Reduces exploration to speed up exploitation.

• **Training Steps**: total\_timesteps = 2000 (short fine-tuning period).

• **Logging**: tensorboard\_log = tb\_log\_dir+'/DQN\_merge\_hyper1'

• **Model Save Path**: dqn\_merge\_hyper1.zip

**A screenshot of a graph

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

The model is learning effectively, improving survival time and rewards, but further tuning (longer training, reward shaping, or adjusted gamma) may be needed for optimal performance.

**5.2 Merge 2**

• **Changes from Merge1**:

• Increased training steps: total\_timesteps = 20000 (from 2000 in Merge1).

• Updated TensorBoard log directory: DQN\_merge\_hyper2

•: The longer training duration helps the model adapt better to the merge task.

**A screenshot of a graph

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

• **Better Performance**: Increased survival time and reward compared to Hyper1.

• **Stable Learning**: Loss fluctuates but trends lower.

• **Further Tuning?**: Additional reward shaping or extending training may enhance results further.

**5.3 Merge 3**

**Merge3: Environment Customization for Reward Reshaping**

• **New Environment Adjustments**:

• collision\_reward = -5: Heavier penalty for crashes.

• reward\_speed\_range = [5, 30]: Adjust reward distribution for speed.

• offroad\_terminal = True: Episode ends on off-road incidents.

• Environment reconfigured using env\_merge.unwrapped.configure().

• **Hyperparameter Adjustments**:

• gamma = 0.8: Higher discount factor for long-term planning.

• exploration\_fraction = 0.7: More focused exploration.

• **Training Steps**: total\_timesteps = 20000 (same as Merge2).

• **Logging & Saving**:

• TensorBoard: DQN\_merge\_hyper3

• Model save path: dqn\_merge\_hyper3.zip

• These changes encourage safe but efficient merging by penalizing crashes and modifying speed-based rewards.

**A screenshot of a graph

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

The results of DQN\_merge\_hyper3 indicate substantial improvements in the agent’s performance, both in terms of survival and reward optimization. The episode length has increased to 14.87, showing that the agent has learned more efficient merging strategies, likely by better timing lane changes and avoiding collisions. The episode reward has also significantly improved to 14.23, suggesting that the agent is consistently making high-reward decisions, likely influenced by the refined reward structure, which includes stronger penalties for collisions and better incentives for maintaining safe, high-speed merging behavior. The exploration rate successfully decays to 0.05, confirming the agent has fully transitioned from exploration to exploitation, relying on a learned policy instead of random actions. The training loss, while initially volatile, stabilizes to 0.0288, showing that the model has converged well with minimal Q-value fluctuations, indicating effective learning. Compared to Hyper2, the model in Hyper3 demonstrates stronger generalization and more stable decision-making, with reduced loss variability and an overall more refined policy. However, further fine-tuning, such as longer training durations, increased traffic complexity, and testing across different traffic densities, could help validate the robustness of the agent’s learned behavior. These results suggest that Hyper3 is a significantly optimized version of the model, but further experiments could explore if additional refinements can lead to even greater efficiency in merging scenarios.

**5.4 Merge 4**

• **New Features**:

• **Custom Reward Wrapper (SpeedRewardWrapper)**:

• Penalizes low speeds (reward -= 0.5 if speed < 20).

• Adds speed-based positive rewards (reward += speed / 30).

• **Modified Environment Configuration**:

• vehicles\_count = 50: Increased traffic for more realistic merging.

• duration = 40: Extended episode length.

• collision\_reward = -5.0: Reinforces safety.

• right\_lane\_reward = 0.2: Slightly encourages right-lane driving.

• high\_speed\_reward = 0.5: More aggressive encouragement for speed.

• lane\_change\_reward = 0.3: Incentivizes lane changes.

• offroad\_terminal = False: Off-road driving no longer ends the episode.

• **Same Hyperparameters as Merge3**:

• gamma = 0.8

• exploration\_fraction = 0.7

• **Training Steps**: total\_timesteps = 20000

• **Logging & Saving**:

• TensorBoard: DQN\_merge\_hyper4

• Model save path: dqn\_merge\_hyper4.zip

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

Merge 4 incorporates reward shaping, increased traffic (50 vehicles), and enhanced speed incentives, but results in slightly lower episode length (13.57) and reward (12.76) compared to Merge 3. The higher loss variability (~0.0747) suggests increased learning complexity, likely due to the new environmental dynamics.

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Key Adjustments** | **Hyperparameters** | **Training Steps** |
| **Merge1** | Basic fine-tuning, minor hyperparameter tweaks | gamma=0.7, exploration\_fraction=0.5 | 2,000 |
| **Merge2** | Extended training duration | Same as Merge1 | 20,000 |
| **Merge3** | Reward reshaping (collision, speed), off-road termination | gamma=0.8, exploration\_fraction=0.7 | 20,000 |
| **Merge4** | Custom reward function, traffic density increased, lane change reward added | Same as Merge3 | 20,000 |

Comparison Across Merge 1-4

Merge 1: Short training (~2K steps), lower episode length (11.55) and reward (9.89), mainly adapting the pre-trained model to the merge task.

Merge 2: Extended training (~20K steps), leading to better survival (12.9) and reward (11.28), improving decision-making.

Merge 3: Best performance, with longest episode length (14.87) and highest reward (14.23) due to collision penalties, better gamma tuning, and refined exploration decay.

Merge 4: Introduced denser traffic and modified speed incentives, making the environment tougher. This led to a slight drop in reward and survival, suggesting that further reward tuning or extended training might be needed to help the agent adapt.

**Roundabout Environment Results**

We tried roundabout env with last model

Based on the video analysis, the agent can enter the roundabout but hesitates to make subsequent moves. During the entrance, it experiences frequent collisions, and once inside, it takes too long to decide on a turn—sometimes even getting stuck between maneuvers.

To optimize performance, we can adjusting the environment parameters through a wrapper function that refines collision handling, rewards, and penalties for staying too long. Specifically: Collision Handling: Reward Adjustments. Penalty for Delay

**PPO – Base Model**

With the baseline model I think it's going to train almost nothing because the first thing the learning rate seems to be really small

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A graph with numbers and lines

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

**Hyperparameter Tuning #1**

Post-Hyperparameter Tuning Analysis & Next Steps

After tuning the hyperparameters, we observed that while the value function improved significantly (higher explained variance and lower value loss), the overall policy performance declined. The hyperparameter-tuned model exhibited \*\*shorter episode lengths and lower rewards\*\* compared to the baseline PPO, suggesting that the agent struggled to generalize its decision-making effectively.

This was likely due to \*\*over-restrictive policy updates\*\*, as evidenced by:

- A \*\*high clip fraction\*\* and more clipped updates.

- \*\*Lower entropy loss\*\*, indicating reduced exploration and premature convergence to suboptimal policies.

**Next Hyperparameter Updates & Hypothesis**

For the next hyperparameter tuning, we will make the following changes:

1. \*\*Reduce `clip\_range` to 0.15\*\* → Allows more flexibility in policy updates and reduces excessive clipping.

2. \*\*Increase `ent\_coef` to 0.01\*\* → Encourages exploration and prevents premature convergence.

3. \*\*Reduce `learning\_rate` to 2e-4\*\* → Slows down aggressive policy updates while maintaining learning stability.

Hypothesis:

With these adjustments, we expect the agent to:

- Learn a \*\*more balanced policy\*\* with better exploration.

- Avoid \*\*premature convergence\*\* to suboptimal strategies.

- Achieve \*\*higher episodic rewards\*\* while maintaining a strong value function.

- Allow \*\*more efficient policy updates\*\* without excessive clipping.

**Hyperparameter tuning #2**

The updated hyperparameter tuning (\*\*“hyper2”\*\*) showed improvements in the \*\*value function\*\*, with:

- \*\*Better explained variance\*\* and \*\*lower value loss\*\*.

- \*\*More controlled policy updates\*\*, as seen in the \*\*moderated KL divergence and clip fraction\*\*.

However, improvements in \*\*episode reward and length\*\* remain modest compared to the baseline. This suggests that:

- \*\*Reducing `clip\_range` to 0.15\*\* and \*\*increasing `ent\_coef` to 0.01\*\* helped the agent learn more stable value estimates and smoother policy updates.

- \*\*Exploration may still be limited\*\*, and aggressive updates might \*\*hinder\*\* the agent from achieving \*\*longer, more rewarding episodes\*\*.

Hyperparameter Tuning: Goals for "hyper3"

For the next tuning iteration (\*\*"hyper3"\*\*), our objective is to \*\*increase lane-switching efficiency, improve speed control, and balance risk-taking\*\*. The following changes have been implemented:

**Expected Changes in "hyper3"**

1. More Decisive Lane Switching\*\*

Increase `ent\_coef` to 0.02:\*\*

Encourages bolder, more exploratory decisions.

Set `vf\_coef` to 0.75:\*\*

Enhances the value function’s estimation, helping the agent trust its lane-switching decisions.

2. \*\*Better Speed Control\*\*

- \*\*Reduce `gamma` to 0.97:\*\*

This reduction shifts the focus toward immediate speed incentives, reducing over-cautious long-term planning.

3. \*\*Encouraging Risk-Taking (While Staying Optimal)\*\*

- The reward function is adjusted to \*\*favor higher speeds and overtaking behavior\*\*, allowing the agent to compete more effectively with other cars rather than lagging behind.

### Additional Hyperparameters

- \*\*Learning Rate:\*\* Set to `2e-4` for stable and gradual updates.

- \*\*Rollout & Batch Size:\*\*

`n\_steps=4096` and `batch\_size=128` support longer rollouts and gradient stability.

- \*\*Training Epochs:\*\* `n\_epochs=20` to allow ample updates per training cycle.

- \*\*GAE Lambda:\*\* `gae\_lambda=0.95` for controlled variance in the advantage estimation.

- \*\*Clip Range:\*\* `clip\_range=0.15` ensures smooth policy updates.

### Hypothesis

With these changes, we expect the agent to:

- \*\*Switch lanes more efficiently\*\* with less hesitation.

- \*\*React better to immediate speed incentives,\*\* optimizing its speed control.

- \*\*Balance risk-taking\*\* by exploring aggressive maneuvers without compromising overall stability.

- \*\*Discover improved strategies\*\* for lane switching and speed control, leading to higher episodic rewards and increased survival time in the highway environment.

**Hyperparameter Tuning #3**

**A screenshot of a graph

AI-generated content may be incorrect.**

**A screenshot of a graph

AI-generated content may be incorrect.**

PPO Hyperparameter Tuning: Hyper3 Analysis vs. Previous Models\*\*

Performance Comparison

After tuning the hyperparameters for the third iteration (`Hyper3`), we observed a few notable changes in training dynamics and performance. Compared to the \*\*Baseline PPO (`PPO\_1`), `Hyper1`, and `Hyper2`\*\*, `Hyper3` demonstrates improved \*\*episode length and reward stability\*\*, but its \*\*real-world performance remains suboptimal\*\*, especially in terms of \*\*speed control, lane-switching confidence, and overtaking behavior\*\*.

### \*\*Key Observations\*\*

- \*\*Survival Time Increased\*\*

- `Hyper3` achieves the \*\*longest episode length (`ep\_len\_mean = 29.07`)\*\*, indicating better survival.

- However, longer episodes do not necessarily mean optimal performance, as the agent \*\*might simply be avoiding interactions rather than making efficient driving decisions\*\*.

- \*\*Policy Updates Are Weak\*\*

- `Hyper3` has a \*\*low KL divergence (`approx\_kl = 0.0053`)\*\*, which means \*\*policy updates are too small\*\*.

- The \*\*clip fraction remains relatively high\*\*, meaning \*\*many updates are being rejected\*\*, potentially leading to \*\*hesitation in lane-switching\*\*.

- \*\*Speed Control Remains an Issue\*\*

- Despite increased exploration (`ent\_coef=0.02`), the agent \*\*does not accelerate aggressively\*\* and often lags behind other vehicles.

- \*\*No significant improvement in episodic reward (`ep\_rew\_mean = 20.43`)\*\*, suggesting that \*\*speed and overtaking strategies are not being optimized effectively\*\*.

- \*\*Lack of Risk-Taking for Overtaking\*\*

- The \*\*entropy loss is decreasing\*\*, meaning the policy is becoming \*\*too deterministic too early in training\*\*.

- The agent \*\*is not experimenting enough with overtaking or higher speeds\*\*, leading to inefficient driving behavior.

## \*\*Next Steps: Testing in Merge Environment & Implementing a Custom Reward Wrapper\*\*

Since `Hyper3` does not show clear improvements in lane-switching and overtaking, the next step is to:

1. \*\*Test the model in the Merge Environment\*\*

- This will assess how well the agent handles \*\*entering and exiting highway ramps\*\*.

- If similar hesitation and speed issues persist, it confirms that \*\*the policy itself needs further tuning\*\*.

2. \*\*Introduce a Custom Reward Wrapper to Encourage Speed & Overtaking\*\*

- Add a \*\*speed-based reward\*\* to incentivize higher velocity.

- Implement a \*\*proximity-based overtaking reward\*\*, so the agent learns to \*\*overtake instead of staying behind slower vehicles\*\*.

- Introduce a \*\*penalty for unnecessary lane-switching\*\*, reducing hesitation and encouraging \*\*more decisive maneuvers\*\*.

The next steps will involve analyzing the model's behavior in the \*\*Merge Environment\*\* and designing a \*\*custom reward structure\*\* to improve real-world driving performance.

**A2C Algorithm**

**Base model**

**A screenshot of a computer code

AI-generated content may be incorrect.**

A graph on a screen

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A2C Initial Analysis & Optimization Path

The training executed smoothly for 2000 steps, but performance is below expectations with rewards at 14.27 and episode lengths around 20.36.

A high value loss (~58) suggests the model is struggling with return estimation, leading to suboptimal policy updates.

The explained variance is near zero, meaning the value function isn’t learning well, which impacts overall decision-making.

Low n\_steps=5 is likely causing instability, as frequent updates disrupt long-term learning patterns.

To optimize, increasing n\_steps=32 will improve credit assignment, reducing learning\_rate=2.5e-4 will stabilize training, and raising gamma=0.995 will enhance long-term planning.

Additionally, expanding the network size (256,256) and extending training to 20,000 steps will support better function approximation and meaningful convergence.

**Hyperparameter tuning #1**

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a graph

AI-generated content may be incorrect.

A2C Hyper1 Analysis: Addressing the Performance Drop

* While episode rewards (~20.38) and length (~27.7) improved numerically, the agent’s real-world behavior worsened, indicating inefficiencies in learning.
* The explained variance is near zero or negative, suggesting poor generalization, meaning the value function fails to estimate long-term returns correctly.
* A high value loss (~695) signals severe instability, making the agent struggle with reward prediction and proper return estimation.
* Entropy loss is dropping too fast, which means the agent isn’t exploring enough, leading to overfitting to an inefficient lane selection strategy.
* Frequent policy loss fluctuations suggest erratic decision-making, potentially causing the agent to stick to fixed top or bottom lanes instead of dynamically adapting.

Planned Fixes Before A2C Hyper2

* Increase entropy coefficient (ent\_coef=0.03) to encourage exploration and reduce overfitting.
* Reduce learning\_rate=1.5e-4 for smoother policy updates, preventing instability.
* Maintain gamma=0.995, as reducing it could worsen short-term fixation on specific lanes.
* Increase n\_steps=128 for more stable updates and better long-term decision-making.
* Manually analyze logs for lane-selection biases, ensuring the agent learns adaptive merging behavior.

Expected Outcome: More balanced policy updates, less lane fixation, and better adaptation to changing traffic conditions for a more flexible and generalized merging strategy.

**Hyperparameter tuning #2**

A screenshot of a computer program

AI-generated content may be incorrect.

**Observations from Hyperparameter Tuning (A2C Hyper2)**

• Entropy loss in Hyper2 is lower than in Hyper1, making the agent more deterministic but reducing its exploration ability. A slight increase in the entropy coefficient may be needed.

• The explained variance in the base A2C model was near zero, while Hyper1 showed large fluctuations. Hyper2 has improved stability but still experiences occasional spikes, likely due to aggressive learning rates.

• Policy loss in Hyper1 was highly erratic, whereas Hyper2 follows a more stable downward trend, suggesting improved gradient updates. Value loss in Hyper1 had large peaks, showing instability, while Hyper2 is more controlled, but further refinements could enhance training consistency.

**Issues in Real-Time Performance**

• The agent remains in the same lanes for extended periods and fails to explore new routes dynamically.

• There is hesitation in proactive lane-switching and overtaking decisions, limiting adaptability in traffic.

• While learning efficiency has improved, the model lacks aggression in overtaking and responding to dynamic lane changes.

**Next Steps: A2C Hyper3 Adjustments**

• Increase entropy coefficient (ent\_coef=0.05) to encourage exploration.

• Lower learning rate (learning\_rate=1e-4) for more stable updates.

• Increase n\_steps (n\_steps=256) to allow better long-term decision-making.

• Expand the policy network (pi=[256, 256, 128]) to refine complex maneuvers.

• Test a lower gamma value (gamma=0.98) to prevent excessive long-term dependencies.

**Hyperparameter tuning #3**

**A screenshot of a computer

AI-generated content may be incorrect.**

Conclusion and Future Directions

The reinforcement learning experiments across DQN, PPO, and A2C in Highway-Env have provided valuable insights into optimizing autonomous driving policies for complex traffic scenarios. The DQN models demonstrated strong adaptability, particularly in Merge-v0, where reward shaping and extended training led to better survival times and decision-making. The PPO models showed promise but struggled with excessive clipping and limited exploration, requiring further tuning to balance policy updates. The A2C models, despite initial instability, improved with adjustments to entropy, learning rates, and n\_steps, though lane fixation and hesitancy in overtaking remain challenges.

For future improvements, reward engineering should be refined to encourage proactive lane-switching and overtaking while maintaining safety. Longer training durations with progressive exploration decay could enhance generalization across different driving scenarios. Testing models in diverse environments, such as roundabouts and heavy traffic densities, will further assess their robustness. Integrating a hybrid RL approach, combining model-free methods with rule-based constraints, could improve interpretability and stability. With these refinements, the trained RL agents will be better suited for real-world autonomous navigation challenges.