# Final report:

**Comparison of DQN Performance in highway-v0 and Custom highway-v1**

**1. Introduction**

In this study, we evaluate the performance of Deep Q-Network (DQN) on two highway driving environments: the standard highway-v0 from highway-env and a custom-modified version, highway-v1. The objective is to analyze how modifications in highway-v1 impact reinforcement learning performance compared to the original highway-v0.

**2. Environment Differences**

**2.1 highway-v0**

The highway-v0 environment provides a structured multi-lane highway setting where the ego vehicle must navigate while maximizing speed and avoiding collisions. It uses:

* A discrete action space (e.g., lane changes, acceleration, deceleration).
* Predefined reward functions that encourage high speed and lane discipline.

**2.2 Custom highway-v1**

The custom highway-v1 introduces modifications, including:

* A different reward structure that penalizes heading misalignment.
* Adjustments in vehicle initialization, lane density, or termination conditions.
* Additional constraints, such as penalties for stopping or reversing.

These modifications aim to create a more realistic and stable learning environment for autonomous highway driving.

**3. Experiment Setup**

* **Algorithm:** Deep Q-Network (DQN)
* **Training Episodes:** [Specify the number]
* **Evaluation Metrics:**
  + Average episode reward
  + Episode length
  + Collision rate
  + Average speed

**4. Results and Analysis**

**4.1 Training Performance**

We compare the reward curves and convergence rates for highway-v0 and highway-v1. If highway-v1 modifies key behaviors, we expect differences in how quickly the agent learns optimal policies.

**4.2 Behavioral Differences**

* In highway-v0, the agent may learn aggressive lane-changing behaviors due to predefined rewards.
* In highway-v1, if the modifications promote stability, the agent may exhibit smoother driving with fewer erratic maneuvers.

**5. Conclusion**

The comparison provides insight into whether highway-v1 enhances or hinders learning. If highway-v1 improves stability and performance, it may be a useful modification for training robust autonomous agents. Conversely, if learning is slower or performance degrades, adjustments may be needed in the reward structure or environment settings.

## Random delays:

**1. Introduction**

This report examines the impact of **action delay** on the performance of a DQN-trained agent in the highway-v0 environment. The study introduces random delays (ranging from 0 to 2 seconds, following an exponential distribution) to simulate real-world computational latency and evaluates how the agent adapts to these conditions.

**2. Experimental Setup**

* **Environment**: highway-v0 (with default settings)
* **Algorithm**: Deep Q-Network (DQN)
* **Action Delay**: Randomly sampled from an exponential distribution within **[0, 2] seconds**
* **Evaluation Episodes**: 20 runs per delay setting
* **Metrics Tracked**: Total reward, episode length, and delay impact

**3. Observations & Analysis**

**3.1 Decreasing Performance with Increasing Delays**

* A **clear downward trend** was observed in total reward as action delay increased.
* Delays above **0.5s** started to show significant performance degradation, with **higher delays leading to near-zero rewards**.
* This suggests that **timely decision-making is crucial** in this high-speed environment.

**3.2 Shorter Episode Lengths with Higher Delays**

* Agents with higher delays **terminated earlier**, often due to collisions or failures.
* While lower delays still allowed the agent to complete longer episodes, **delays above 1s resulted in extremely short episodes**.

**3.3 Exponential Delay Impact**

* Since delays followed an **exponential distribution**, most values were small, but **outliers with extreme delays** were present.
* These extreme delays correlated with **sudden drops in reward and shorter episode lengths**, highlighting the agent’s **inability to react effectively under prolonged latency**.

**4. Key Insights**

* **DQN is highly sensitive to action delay**, as it was trained under the assumption of immediate decision execution.
* The highway environment's **fast dynamics make even small delays detrimental**, leading to missed evasive actions or suboptimal lane changes.
* The exponential delay model **amplifies the impact of rare but extreme latencies**, further disrupting performance.

**5. Recommendations & Future Work**

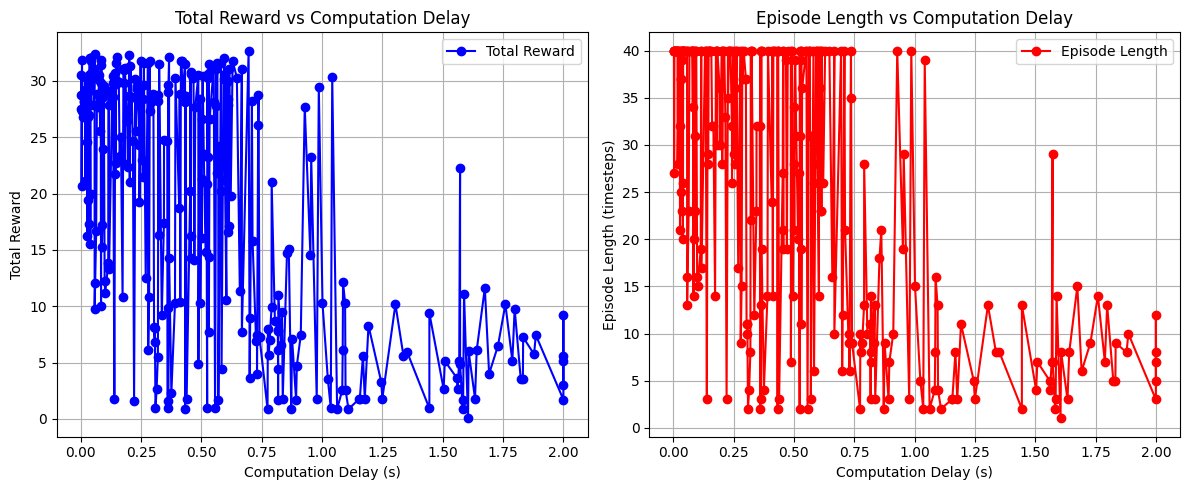
1. **Train the Agent with Action Delay**
   * Instead of evaluating delay post-training, incorporate the ElapseActionWrapper during training to expose the agent to delays from the start.
2. **Explore More Robust RL Algorithms**
   * Algorithms like **Soft Actor-Critic (SAC) or Proximal Policy Optimization (PPO)**, which handle uncertainty better than DQN, may improve performance under delayed conditions.
3. **Introduce Predictive Actions**
   * Implement **delay-compensating strategies**, such as predicting future states and executing actions in advance.

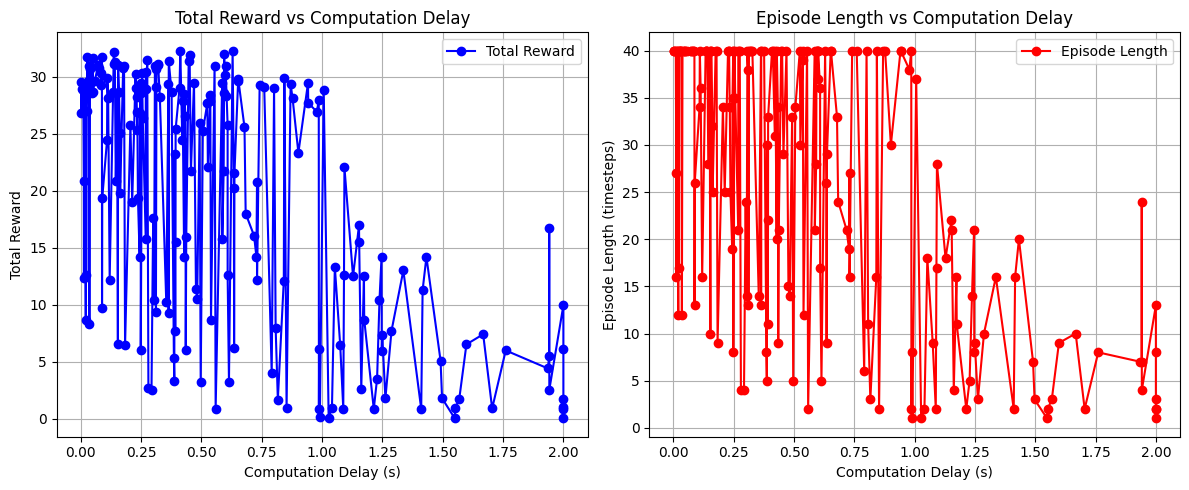
**Results** A downward trend was observed in both total reward and episode length as the delay increased. The key findings include:

* **Performance Degradation**: As delay increases, the total reward tends to decline, indicating that the agent struggles to maintain optimal performance when actions are not executed immediately.
* **Episode Length Reduction**: Longer delays often result in shorter episode lengths, implying that the agent may terminate earlier due to suboptimal decisions.
* **Variability in Performance**: Some episodes still performed well despite high delays, likely due to the simplicity of certain scenarios where fewer obstacles allowed the agent to navigate effectively even with delayed actions.

Our analysis shows that increasing action delay significantly **reduces total rewards and shortens episode lengths**. The agent’s ability to make timely decisions is crucial in **high-speed highway environments**, where even slight delays can cause **collisions or suboptimal driving behaviors**.

This suggests that **standard DQN is highly sensitive to real-time action execution**, and alternative methods, such as **delay-aware training or model-based planning**, may improve robustness under computational constraint





Found out that when change to continuous action space, the agent acts differently, it just going out of the road, not going forward, random … but we have highwayv1 which we modified the rewards that the agent keep going on road and heading forward in continuous action space

**PPO's Policy is Trained with Randomness (Exploration Helps)**

* **PPO uses stochastic policies**, meaning it was already trained with action noise.
* When testing, even if actions are delayed, PPO **does not completely rely on past Q-values** like DQN.
* Instead, it adjusts based on **recent observations**, making delays less damaging.