5.1.2 Methodology Refinement & Evaluation Protocol

Initially, the training was conducted using the standard environment configuration (`train.py`) and evaluated using default metrics. However, an in-depth analysis of these preliminary results revealed inconsistent behaviors. While the agents were often able to complete the maximum episode length (1000 steps), visual inspection via video logs showed that the driving was erratic. The agents frequently survived by spinning in circles or drifting off-track without being penalized sufficiently, exploiting the survival reward rather than learning proper lane-keeping.

Hypothesis: For an autonomous vehicle, safety and road adherence must take precedence over raw velocity. We hypothesized that introducing a strict negative penalty for leaving the designated track would force the agent to learn stability and reduce "cheating" behaviors.

Reward Policy Evolution: To test this hypothesis, we formally defined two distinct reward policies used during the experimentation phase.

A. Baseline Policy (Standard CarRacing-v2) *Used in initial training experiments (`train.py`).*

The default reward structure focuses purely on velocity and track completion. The reward (R_t) at step t is defined as:

> **R_t = (1000 / N × Δ_visited) - 0.1**

Where:

- N: The total number of track tiles in the generated circuit.
- $\Delta_{visited}$: The number of **new** track tiles visited in the current step.
- $1000/N$: The normalized reward points gained for visiting a new tile.
- -0.1: A constant time penalty applied at every frame to encourage faster driving.
- **Deficiency:** There is no explicit negative reward for driving on the grass, allowing the agent to cut corners or survive off-track.

B. Robust Policy (Implementation: Grass Penalty) *Used for the final evaluated models (`train2.py`).*

To address the baseline deficiencies, we implemented a custom `GrassPenaltyWrapper` that modifies the reward structure based on visual feedback. The new reward function is:

$$R'_t = R_t - P$$

Where:

- R_t: The baseline reward defined above.
- P_{grass}: The dynamic penalty term for unsafe driving.

Penalty Logic (*P_grass*): The system analyzes the RGB observation to detect "grass" pixels (Green > 150, Red/Blue < 100) and applies the penalty as follows:

- **If *green_ratio* > 0.25:** The car is considered off-track.
  - **Penalty Applied:** P_{grass} = **0.8**
- **Otherwise:**
  - **Penalty:** P_{grass} = **0**

Safety Termination: The episode is automatically terminated (Fail) if the car remains off-track (*green_ratio* > 0.25) for more than **50 consecutive frames**.

Evaluation Metrics: For the comparative analysis, we use **Mean Reward** to assess driving quality and **Win Rate** (% episodes > 900 points) to determine optimal racing behavior.