Project Highway

I. Introduction

A highway is a type of road designed for the movement of a large number of vehicles at high speed. Because of this, cars traveling on highways need to maintain proper control over their **speed, lane position, and road boundaries** to ensure safety. Highways are usually built with multiple lanes, smooth surfaces, and traffic management systems to reduce accidents and allow vehicles to move efficiently.

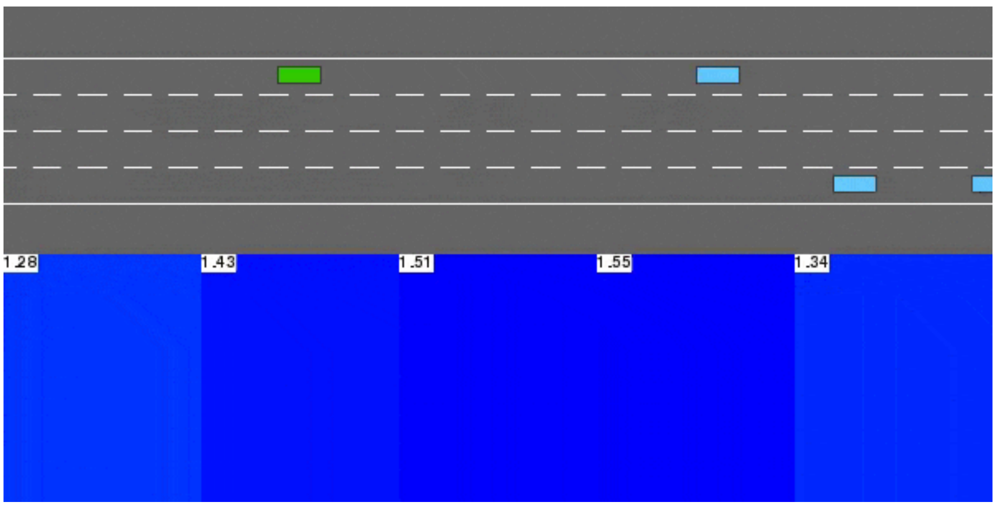

Figure 1. Blue car is an agent of this environment.

https://github.com/Farama-Foundation/HighwayEnv

The study of highways is important because it involves not only the design and construction of the road itself, but also the rules, systems, and technologies that help vehicles operate safely at high speeds. Key aspects include **speed control, lane discipline, traffic flow management, and safe navigation of curves and intersections**.

II. Objectives & Goal

Train multiple cooperative/competitive drivers that
(1) Avoid collisions,
(2) Maintain efficient speeds and lane discipline under dense traffic
(3) Reach destinations within a time budget—evaluated under varied traffic patterns and driver behaviors.
We'll use **highway-env**'s multi-agent setup by increasing controlled_vehicles and switching the env to **MultiAgentAction**/**MultiAgentObservation** so each agent gets its own obs and sends its own action.

III. Scope & Limitation

Scope

- The project focuses on developing an AI model for ***highway driving simulation***.
- The model will be trained to ***detect vehicles, avoid collisions, and maintain safe lane positioning***.
- It will handle **multi-lane highway scenarios** with varying traffic density.
- The system will support **multi-agent RL**, where each car acts as an independent agent that learns to control its own driving behavior.
- The AI will simulate **interactions among multiple vehicles** on a highway, including lane changes, overtaking, and safe distance maintenance.
- The scope covers **simulation-based testing only**, not real-world deployment.

Limitations

- The model will be limited to **highway environments only**; it does not cover city roads, intersections, or rural areas.
- Environmental conditions is simple road, dry.
- The project will focus mainly on **vehicle-to-vehicle interaction**, and may not fully address pedestrians, animals, or unexpected roadblocks.
- The AI model's performance will depend on the **quality and quantity of training data** available.
- Real-world implementation will require additional hardware (sensors, cameras, LiDAR, etc.) and regulatory approval, which are beyond the current project scope.

IV. Methodology

A) Overview

We adopt a **hierarchical LLM-guided RL** approach. A **Large Language Model (LLM)** selects high-level **maneuvers/constraints** (e.g., *keep lane*, *change left*, target speed), while a **low-level RL policy** (PPO/SAC) executes continuous control (steer/throttle/brake) to satisfy those constraints safely and efficiently.

- **LLM (planner):** interprets a compact, textual summary of the scene → outputs a maneuver and parameters (JSON).

- **RL (controller):** receives simulator observations + LLM's subgoal (one-hot + target speed) and outputs continuous actions.
- **Safety shield:** simple rule checks (e.g., TTC/headway) override unsafe actions.

B) Observation → Language (state summarizer)

Use your `highway-env` **Kinematics** observation and convert to traffic facts the LLM can reason about.

- **Computed features per step:**
  - Front gap & relative speed (ego vs lead), left/right lane gaps, **TTC** (time-to-collision), headway, lane index, curvature ahead, speed limit & ego speed.
- **Text template (example):**

  "Ego lane=2/3, speed=95 km/h (limit=100). Lead gap=36 m, rel_speed=−5 km/h, TTC=4.8 s. Left lane gap front/back = 22 m / 18 m. Right lane not available (barrier). Traffic density=high, road=straight."

V. Dataset

This project uses a **simulation-generated dataset** tailored for highway scenarios and multi-agent RL. Data consists of time-stamped state–action–reward transitions for the ego car and nearby vehicles, plus scenario metadata (traffic density, lane count, speed limits). Dataset will collect base on

- Highway OpenAI https://highway-env.farama.org/dynamics/road/lane/
- Existing: https://github.com/Farama-Foundation/HighwayEnv

1) Environments & modes

- **Base env:** highway-v0 (straight multilane highway). Default config emphasizes **high speed + no collisions** and rewards keeping to right lanes; start from defaults (e.g., 4 lanes, ~50 vehicles, 40s episodes), then vary density/speeds for curricula. A faster variant highway-fast-v0 exists for large-scale training. highway-env.farama.org
- **Multi-agent switch:** set controlled_vehicles ≥ 2, action → MultiAgentAction with action_config: DiscreteMetaAction, and observation → MultiAgentObservation (e.g., Kinematics per agent). highway-env.farama.org

2) State (observation) design

Pick one (or mix during curriculum):

- **Kinematics (tabular):** fixed V×F feature table (e.g., presence, x, y, vx, vy, cos_h, sin_h) for nearby vehicles; normalized and optionally ego-relative. Stable and sample-efficient. highway-env.farama.org
- **Occupancy grid (spatial):** W×H×F grid encoding presence/kinematics around ego; good for dense traffic topology. highway-env.farama.org
- **(Optional) Time-to-Collision or grayscale image** stacks for advanced probes. highway-env.farama.org

3) Action space

Use **Discrete Meta-Actions** (simple, robust): {0: LANE_LEFT, 1: IDLE, 2: LANE_RIGHT, 3: FASTER, 4: SLOWER}. The env masks unavailable actions (e.g., can't change lane at road edge). highway-env.farama.org

4) Reward shape (per agent)

Start minimalist (mirrors the library's philosophy) and only add shaping if needed:

- **Speed term:** normalized progress toward target speed band.
- **Collision term:** large penalty on impact; keep rewards in [0,1].
- Optional gentle shaping: right-lane bonus, jerk penalty (comfort), headway penalty (follow-distance). highway-env.farama.org

5) Scenario set (for training & logging)

Use a curriculum from easy → hard; 3–5 lanes; episode length 30–60 s.

1. **Free-flow cruising (baseline)**
   Low density; mixed speeds; goal = reach/hold target speed without collisions. Defaults close to library's base config. highway-env.farama.org
2. **Dense commuting**
   vehicles_count high; small gaps; frequent merges; reward emphasizes safe gaps + minimal lane changes while maintaining speed band. highway-env.farama.org
3. **Stop-and-go waves**
   Periodic slowdowns ahead (scripted speed perturbations) to test anticipation and headway control.
4. **Aggressive neighbors**
   A subset of background cars use more "edgy" lane-change policies to induce cut-ins; evaluate collision avoidance & comfort (jerk).

5. **Lane closure / incident**
   One lane blocked mid-episode → requires early merge decisions and cooperative etiquette.
6. **Time-budget runs**
   Add a soft deadline: reward slight bonus for earlier arrival without increasing collision/comfort penalties.

VI. Expected Result