AI4R: Full Scale Autonomous Driving Project

Kangyu Zhu (Robin) (ID# 1160170)
Tanzid Sultan (ID# 1430660)

1. Task Introduction

In this project, we tackle the challenge of enabling a full-sized passenger vehicle to drive on high speed with tight corners autonomously. Our objective is to design, synthesize and compare different policies which ensure safe and comfortable autonomous driving under challenging conditions. To achieve this, **Proximal Policy Optimization (PPO)** and **Soft Actor-Critic (SAC)** algorithms are implemented and evaluated.

Autonomous driving on high-speed roads with tight corners presents significant challenges due to the need for precise control, rapid decision-making. The vehicle must maintain appropriate speeds, handle sharp turns smoothly. Performance metrics are carefully designed to quantitatively assess the effectiveness of the policies. These metrics include various aspects of driving performance.

To enhance the robustness and generalization capabilities of the policies, we use domain randomization during training. This technique varies environmental parameters such as road curvature, vehicle dynamics and surface conditions to expose the policies to a diverse set of scenarios. As a result, the trained policies are able to handle different conditions. Rewards functions are designed to guide the learning process toward desirable behaviors. The reward function encourages maintaining lane position while adhering to speed limits and navigating efficiently. By implementing these reward components, policies can drive safely, comfortably and autonomously on challenging roads.

For evaluation, both policies are tested on a single, more challenging road that was not encountered during training. This road has tighter curves and varying speeds to test policies' ability to generalize and perform under unfamiliar conditions.

2. Performance Metrics

Given the overarching goal of our autonomous vehicle driving on a high speed road with tight corners, with the additional requirement for robust transferability, we have identified the following set of performance metrics, addressing each requirement.

Requirement	Performance Metric	Equation	Justification
Avoid exceeding speed limit	Maximum exceedance over speed limit	$\max_{k=0,..,N_{sim}} (v_k - v_{\lim})$	Both metrics measure how well the policy is able to obey the speed limit. A better policy is expected to have a lower maximum speed exceedance and lower fraction of time exceeding the speed limit.
	Fraction of time exceeding the speed limit	$\frac{1}{N_{sim}} \sum_{k=0}^{N_{sim}} 1_{v_k > v_{\lim}}$	
Lane keeping	Max deviation from center of lane	$\max_{k=0,..,N_{sim}} d_k$	Measures how well the policy can perform lane keeping, by always maintaining a small distance to the path.
	Average heading angle relative to lane	$\frac{1}{N_{sim}} \sum_{k=0}^{N_{sim}} \mu_k$	Measures how well the policy can follow the trajectory of the path, by always maintaining a small heading angle relative to lane.
Smooth steering	Total variation of heading angle	$\sum_{k=1}^{N_{sim}} \lvert \mu_k - \mu_{k-1} \rvert$	Measures the degree of oscillatory perturbations in heading angle over small timescales, which gives us a direct measure of the smoothness of the lane-keeping policy/steering. A smaller total variation means the steering is smoother and less shaky/jittery.
Driving slower around tight corners	Maximum centripetal acceleration	$\max_{k=0,..,N_{sim}} \left(\frac{v_k^2}{r_k} \right)$	Measures ability of the policy to slow the car down near tight corners to avoid slipping and

	Average centripetal acceleration	$\frac{1}{N_{sim}} \sum_{k=0}^{N_{sim}} \left(\frac{v_k^2}{r_k} \right)$	losing control. Smaller values of max and average centripetal acceleration and lower speed (relative to speed limit) at the point of highest curvature are desirable and indicate that the policy is able to slow down appropriately as needed near corners.
	Relative speed at highest curvature point on lane	$\frac{v_k}{v_{\text{lim}}}$ *s.t.* $k = \text{argmin}_j \ r_j$	

where,

$v_k \ = \ speed \ at \ timestep \ k$
$d_k \ = \ dist \ from \ closest \ point \ on \ path \ at \ timestep \ k$
$\mu_k \ = \ heading \ angle \ relative \ to \ path \ at \ timestep \ k$
$r_k \ = \ path \ radius \ (i.e. \ inverse \ curvature) \ at \ closest \ point \ at \ timestep \ k$
$v_{\text{lim}} \ = \ speed \ limit$
$N_{sim} \ = \ total \ number \ of \ simulation \ timesteps$

3. Policy Design and Synthesis

3.1 Reward Function Design

The bulk of our RL policy synthesis efforts were dedicated towards designing the **reward function**. This is the most critical step and is ultimately responsible for the policy's ability to achieve the overarching goal of driving on a high speed road with tight corners. We construct our reward function as a combination of three different **components**. Each of these components have been designed to specifically address a particular requirement for our policy (the same requirements which have been specified in our performance metrics table). Each reward component is shaped separately and the final reward is obtained by taking a weighted sum of the components. The values of the weights need to be specified appropriately to ensure balanced contributions from the different components. A successful rule of thumb which we employed was to design the reward components such that they all vary over a similar range. This is followed by an empirical adjustment of the weights, i.e. the weights are initially set close to 1, then adjusted iteratively through trial and error and observing training results. (Note that these weights are defined as constant multiplicative factors inside each reward component.)

We now provide a detailed description of each reward component and explain its purpose:

1. **Road progress reward**

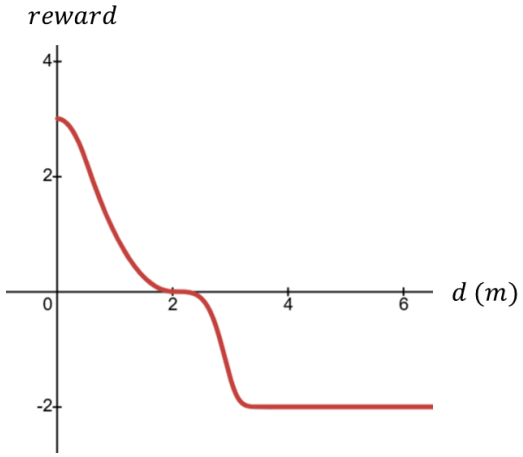
   The road progress reward is designed to encourage the car to make progress along the road, i.e. a positive reward is given when the car progresses along the road. The numerical value of this component is directly proportional to the amount of progress made during the current timestep and can be described as follows:

   $$\text{road progress reward} = c_1(p_k - p_k - 1)$$

   where p_k denotes road progress in time step k and c_1 is a constant.

2. **Line proximity reward**

   The line proximity reward is designed to partially address the lane keeping requirement, by encouraging the car to maintain a small distance from the center line. This is achieved via a piecewise-continuous reward function, which takes on large positive values when the car is very close to the center line and drops off sharply, eventually becoming negative as the car deviates away from the line. More specifically, this reward is designed to be positive over the range $0 \leq d < 2.0$ m and negative for distances larger than 2 m. We also use a $tanh$ function to prevent the negative rewards from undergoing unbounded growth which could potentially make the training process unstable.



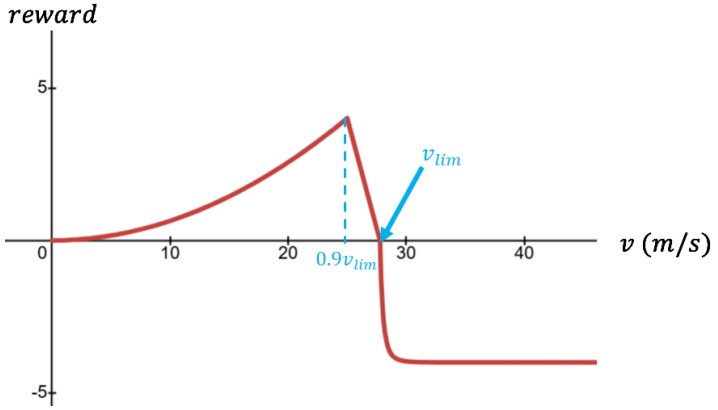
$$\text{line proximity reward} = \begin{cases} 3c_2(1-d^2) & d < 0.5 \\ c_2(2-d)^2 & 0.5 \leq d < 2.0 \\ -c_3 \tanh((2-d)^4) & d \geq 2.0 \end{cases}$$

where d denotes the distance to the closest observed point on the line and c_2, c_3 are constants.

3. Speed reward

Our speed reward component is specifically designed to handle both the speed limit requirement while also encouraging the car to slow down near tight corners. We first describe our process for achieving the speed limit and then show how an appropriate modification allows us to achieve the requirement of slowing down near tight corners.

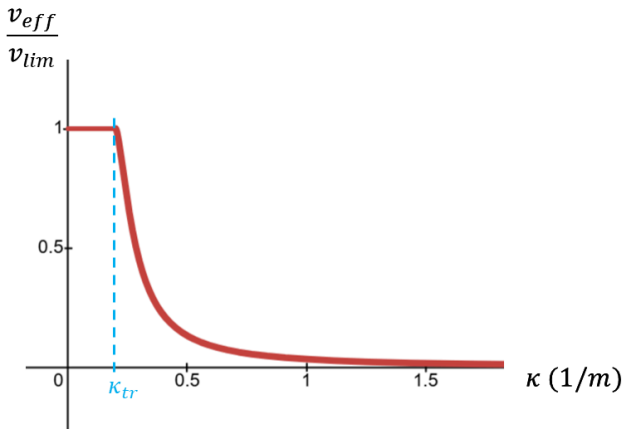
Similar to the line proximity component, we implement the speed reward component using a piecewise-continuous function which takes on large positive values at speeds close to and below the speed limit, then sharply drops off to zero and becomes negative beyond the speed limit. Once again, we prevent the negative rewards from being unbounded.



$$\text{speed reward} = \begin{cases} c_4(\frac{v}{0.9v_{lim}})^2 & v < 0.9\,v_{lim} \\ -c_4(\frac{v-v_{lim}}{0.1v_{lim}}) & 0.9\,v_{lim} \le v < v_{lim} \\ 0.5\,c_4\left(\frac{1}{(v-v_{lim}+1)^4} - 1\right) & v \ge v_{lim} \end{cases}$$

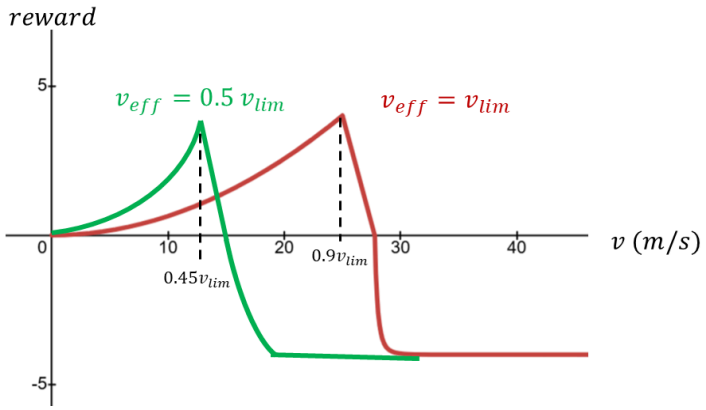
where v is the car speed, v_{lim} is the speed limit and c_4 is a constant.

To encourage the car to slow down near tight corners, we define the concept of an "**effective speed limit**" (v_{eff}) which depends on the road curvature observed at the closest point. The effective speed limit has a maximum value of v_{lim}, with lower values as curvature (κ) increases beyond a fixed threshold value (κ_{tr}). Any curvature under the threshold value is considered small enough, and therefore safe for the car to drive at high speeds. More concretely, we describe the effective speed limit as follows:



$$v_{eff} = v_{lim}\begin{cases} 1 & \kappa < \kappa_{tr} \\ \frac{1}{1+400(\kappa-\kappa_{tr})^{1.2}} & \kappa \ge \kappa_{tr} \end{cases}$$

To modify the speed reward component, we simply replace all instances of v_{lim} with v_{eff}. Thus, as curvature increases v_{eff} becomes smaller, resulting in a horizontal scaling and shifting of the reward function, as illustrated below.

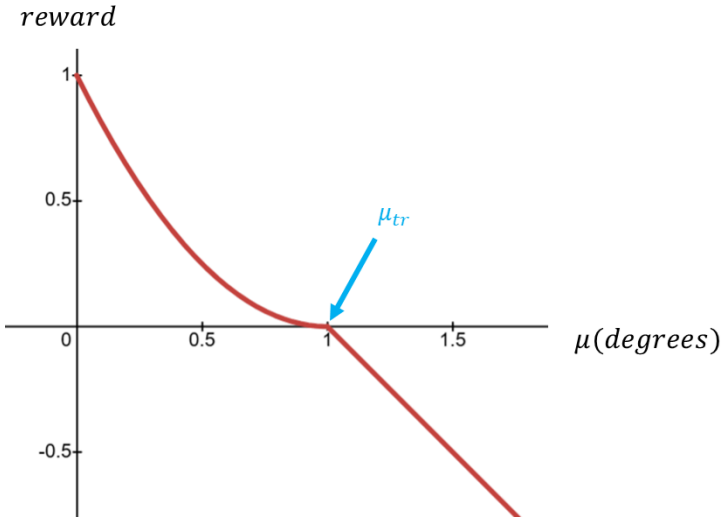


$$\text{modified speed reward} = \begin{cases} c_4\left(\frac{v}{0.9 v_{eff}}\right)^2 & v < 0.9\, v_{eff} \\ -c_4\left(\frac{v - v_{eff}}{0.1 v_{eff}}\right) & 0.9\, v_{eff} \le v < v_{eff} \\ 0.5\, c_4\left(\frac{1}{(v - v_{eff} + 1)^4} - 1\right) & v \ge v_{eff} \end{cases}$$

4. **Steering direction reward**

   The heading angle reward is designed to encourage the vehicle to maintain a small heading angle relative to the road's centerline. This component ensures that the vehicle aligns its orientation with the direction of the road, enhancing lane-keeping performance and passenger comfort.

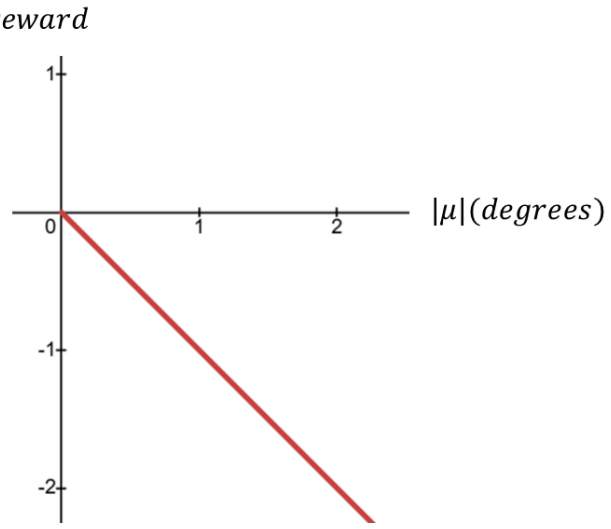
   We implemented the steering direction reward with a piecewise function which gives positive rewards when the heading angle error is small, i.e. less than a degree, and negative rewards when the error exceeds this threshold.



$$\text{heading reward} = \begin{cases} c_5 \frac{1}{4}(\mu - 1)^2 & |\mu| < 1 \\ -c_6 (|\mu| - 1) & |\mu| \geq 1 \end{cases}$$

where c_5, c_6 are constants and μ is the heading angle relative to the center line and the threshold value is set to $\mu_{tr} = 1$.

5. **Steering smoothness reward**
   Steering smoothness reward is designed to encourage the vehicle to make smooth steering. It aims to improve the comfortness of the passenger and vehicle stability by penalizing sudden changes in the vehicle's heading angle.



$$\text{steering smoothness reward} = -c_{av} \ \Delta\mu_{\text{deg}}$$

where μ_k is the heading angle relative to line, $\Delta\mu_k = \mu_k - \mu_{k-1}$ is the change in heading angle between time steps k-1 and k, $\Delta\mu_{deg} = |\Delta\mu_t| \times \frac{180^{circ}}{\pi}$ converts the change from radians to degrees and C_{av} is an constant that controls the magnitude of the penalty

3.2 Training with Domain Randomization

Roads are generated randomly by setting the road parameters.

Policies are trained on these three different type of roads:

Simple Road:

```
simple_road_params = {
    'num_elements_range': (2, 4),               # Fewer elements (2 to 4)
    'straight_length_range': (150.0, 300.0),    # Longer straight segments (150 to 300 meters)
    'curvature_range': (-1/1000.0, 1/1000.0), # Gentle curves (small curvature)
    'angle_range': (10.0, 30.0)                 # Lower curve angles (10 to 30 degrees)
}

medium_road_params = {
    'num_elements_range': (3, 6),               # More elements (3 to 6)
    'straight_length_range': (100.0, 150.0), # Shorter straight segments (100 to 150 meters)
    'curvature_range': (-1/300.0, 1/300.0),  # Sharper curves (higher curvature)
    'angle_range': (20.0, 45.0)                 # Higher curve angles (20 to 45 degrees)
}
SAC_road_params = {
    'num_elements_range': (3, 10),              # More elements (3 to 6)
    'straight_length_range': (50.0, 300.0),  # Shorter straight segments (50 to 150 meters)
    'curvature_range': (-1/50.0, 1/50.0),     # Sharper curves (higher curvature)
    'angle_range': (10.0, 150.0)                # Higher various curve angles (10 to 150 degrees)
}
```

Initially the training was done in two phases: train the PPO for 1000000 steps using simple road parameters and then continue training for an additional 1000000 steps using medium road parameters. However, we observed that the agent's performance degraded after training on the medium road. Training solely on simple road parameters yielded better results, the agent is able to generalize well to more challenging evaluation roads. Simple roads provide a more uniform and less variable environment, the agent can focus on basic controls without being overwhelmed by environmental complexity. The agent receives consistent feedback which enables it to learn fundamental driving skills effectively. Training on medium road may cause the agent to forget previously learned behaviors, this is a phenomenon known as catastrophic forgetting. Additionally another 1000000 steps training on medium roads may not be enough for the agent to adapt to the increased complexity, the agent ends up in a state that is neither well-tuned for simple roads nor trained well for medium roads.

After experimenting with two-phase training, only simple_road_params is used to train the PPO.

SAC is only trained on the SAC_road_params, which contains wider ranges across all the parameters. Due to faster convergence of SAC, it was not necessary to use a multi-phase training schedule with different sets of road parameters.

PPO may not handle high-dimensional or highly variable environments as effectively as some off-policy algorithms like SAC. The on-policy nature of PPO makes it less efficient in diverse environments.

3.3 Policy 1: PPO

PPO is an on-policy, model-free reinforcement learning algorithm, it optimizes the policy directly by following the gradient of expected rewards. It also uses a surrogate objective function that includes a clipping mechanism to limit the size of updates of policy. This prevents drastic changes with stabilized training.

PPO requires fresh data collection for each update as it is an on-policy algorithm, this can be less sample-efficient than SAC. SAC on the other hand is off-policy, it can reuse data from a replay buffer. PPO encourages exploration by including entropy bonuses whereas SAC explicitly maximizes entropy as part of its objective function, this often leads to better exploration. PPO is known for its simplicity and ease of implementation compared to other advanced algorithms, that's why PPO is our first model.

Observations:

- Vx sensor
- Distance to closest point
- Heading angle relative to line
- Road curvature at closest point

Environment Parameters	**Value**	**Motivation**
Speed limit	80 km/h or 22.22m/s	This is used for the maximum speed for the vehicle, ensuring adherence to traffic regulations.
High Road Curvature Threshold	3.0 2.5	This is used to adjust speed and rewards based on road curvature, encouraging slowing down on sharper curves.

Reward Function Constants	**Value**	**Motivation**

Progress Reward Constant	1.0	Scales the reward for progressing along the road, this is a baseline value that ensures the progress is positively reinforced
Line Proximity Reward Constants	3.0 2.5	Control the magnitude of rewards based on the vehicle's lateral distance from the center of the road. These values encourage the car to stay close to the center of the road
Speed Reward Constant	4.0	Scales the reward to maintain appropriate speeds
Heading Reward Constant	2.25 0.2	This encourages the vehicle to align its heading with the road direction. It gives high reward for small heading errors, and large negative reward for larger errors
Angular Velocity Smoothness Reward Constant	0.2	Penalizes rapid changes in steering to promote smooth driving and provide vehicle stability in order to enhance passenger comfort

PPO Hyperparameters	**value**	**Motivation**
Policy Network Architecture	[64, 64]	[64, 64] architectures are standard for continuous control tasks, by changing the hidden layer to [256, 256], the performance is not increased. Therefore, default setting with sizes [64. 64] is used.
Batch size	64	This allows for stable gradient estimates without requiring excessive memory, the value can be increased if more computable resources are available.
Learning rate	3×10^{-4}	The default value balances convergence speed and stability
Discount Factor(Gamma)	0.99	This balances the importance of immediate rewards versus future rewards, high value emphasis more on long-term gains.

Training Parameters	**value**	**Motivation**
Total Timesteps	1000000	This provides sufficient training iterations for the agent to learn effective policies.

Timesteps per Epoch	50000	This determines how frequently the model is saved and evaluated. By comparing all the Epoches, the model performs the best at timestep around 250000. Therefore the model at that time is used for evaluation.

3.4 Policy 2: SAC

Our second Reinforcement Learning based policy algorithm is the Soft-Actor Critic (SAC) which optimizes a stochastic policy in an off-policy setting. Alongside PPO, SAC is considered to be a state-of-the-art RL algorithm, known for its high sample efficiency and overall robustness. SAC incorporates a maximum entropy framework which encourages better exploration of the action space. We use the SAC implementation available through the Stable Baselines 3 python library.

The following observations were made available to our SAC model:

- Vx_sensor
- Distance to closest point
- Heading angle relative to line
- Road curvature at closest point
- Heading angular rate gyro
- Look ahead road curvature

The following table contains the weights used for the various reward function components:

Reward Function Constants	**Value**	**Motivation**
Progress Reward Constant	2.0	Scales the reward for progressing along the road, this is a baseline value that ensures the progress is positively reinforced
Line Proximity Reward Constants	1.0 2.0	Control the magnitude of rewards based on the vehicle's lateral distance from the center of the road. These values encourage the car to stay close to the center of the road
Speed Reward Constant	10.0	Scales the reward to maintain appropriate speeds. Note that we use a higher weight for this constant, compared to the PPO policy training.
Heading Reward Constant	2.0 1.0	This encourages the vehicle to align its heading with the road direction. It gives high reward for small heading errors, and large negative reward for larger errors.

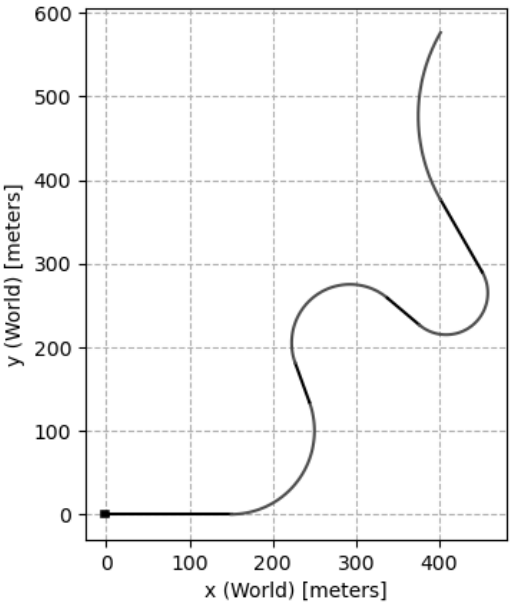
Angular Velocity Smoothness Reward Constant	0.2	Penalizes rapid changes in steering to promote smooth driving and provide vehicle stability in order to enhance passenger comfort.
High Road Curvature Threshold	1/200	Curvature values less than 1/200 are deemed to be safe for high speed driving.

The most relevant hyperparameter settings used for our SAC model are shown in the table below. Note that we also modify the default neural network architecture, by incorporating a 3 hidden-layer network, with 256 units each.

Hyperparameter	Value
Batch Size	512
Buffer Size	200000
Learning Rate	0.0003
Discount factor	0.99
Total training timesteps	100000
Custom Neural Network Architecture	[256, 256, 256]

4. Policy Performance Comparison

In order to thoroughly evaluate and compare our PPO and SAC policies and determine their robustness, we have carried out a total of 160 simulation runs, in which certain environmental parameters are varied. We start with an evaluation road template, shown in the figure on the right, which we deemed to be quite challenging as it contains corners with very high curvature. This road template was not included in our training set. In each simulation run, we apply random perturbations to every segment of this road, e.g. perturbations in the length of straight segments and curvature/angle of curved segments. The road is also under wet conditions across all simulations and a speed limit of 100km/hr is enforced. Additionally, we also perturb the mass of the car and the steering angle offset. We then computed aggregates, i.e. average and standard deviation, of each performance metric for both policies. We present these aggregate results in the following table:



Performance Metric	PPO	SAC
Max exceedance over speed limit (m/s)	0.00 ± 0.00	0.00 ± 0.00
Fraction of time above speed limit	0.00 ± 0.00	0.00 ± 0.00
Max deviation from center line (m)	**0.89 ± 0.13**	4.13 ± 0.67
Average heading angle relative to line (degrees)	**0.95 ± 0.16**	1.65 ± 0.13
Total Variation in Heading Angle (degrees)	8.74 ± 1.09	**6.89 ± 0.58**
Maximum centripetal acceleration (1/s^2)	3.98 ± 0.25	**1.79 ± 0.11**
Average centripetal acceleration (1/s^2)	1.47 ± 0.07	**0.51 ± 0.02**
Relative speed at highest curvature	0.50 ± 0.00	**0.30 ± 0.00**
Total Travel Time (s)	**74.51 ± 3.70**	121.17 ± 10.79

According to these results, it is clear that Pareto dominance does not occur. In other words, we observe certain trade-offs, where each model performs better than the other across a subset of metrics. First, we observe that both models are able to easily stay under the speed limit at all times, as the car tends to drive at speeds that are well below the speed limit. This can be understood as a consequence of the speed reward component which enforces a “low effective speed limit” in the presence of high curvature road segments, leading to the car traveling at very low speeds under such conditions, to avoid slipping and losing control.

Next, we observe that the PPO policy demonstrates superior lane-keeping ability, achieving significantly lower values for the max deviation from center line and average heading angle metrics, compared to the SAC policy. This could be due to the fact that the PPO policy was trained using fewer observations along with the simple default neural network architecture and is therefore able to generalize better. In contrast, the SAC policy has a more complex neural network architecture and uses several additional observations, which very likely may have caused the model to overfit the training data. Both PPO and SAC policies achieve low and comparable values for the total variation in heading angle metric, indicating both are able to satisfy the smooth steering requirement.

We have also observed that the SAC policy is generally more inclined toward traveling at lower speeds, allowing it to drive better in tight corners, as indicated by the lower values of the max

and average centripetal acceleration and also lower relative speed at the point of highest curvature. Indeed, the total travel time is also significantly higher for SAC, compared to PPO, confirming the SAC policy's tendency towards slower driving. When training the SAC policy, we used a relatively large weight for the speed reward component, as smaller values were preventing the model from learning to slow down near tight corners. This could be the likely cause for the slower driving.

To qualitatively assess the performance of our policies, we have also performed single simulation runs with each model across 3 different road layouts, with varying degrees of complexity, in terms of shape of the road. These include a long mostly-straight road, a slightly curvy road and a more challenging road with tight curves (which was previously used as a template during the aggregated simulation runs). Figures 7-9 show the trajectories of each policy from these simulation runs. In all 3 cases, the most notable difference between the trajectories followed by the two policies is that PPO is clearly able to maintain close proximity to the center line while SAC undergoes slight deviations, especially near curved road segments. This qualitative behavior also supports the superior performance of PPO for the lane-keeping metrics.

Furthermore, the ability of both our policies to successfully drive in the three different evaluation roads also demonstrates their overall transferability to new scenarios which were not encountered during training.

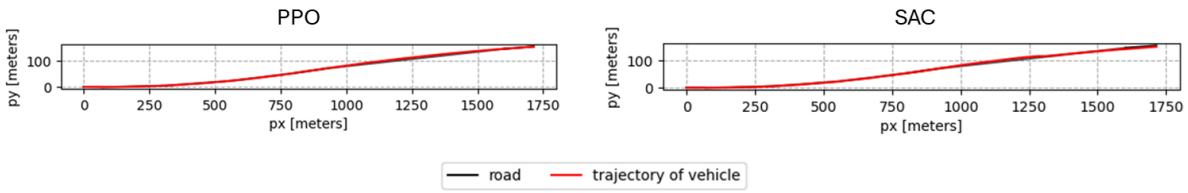


Fig 7: Evaluation road 1 - Long straight-ish road

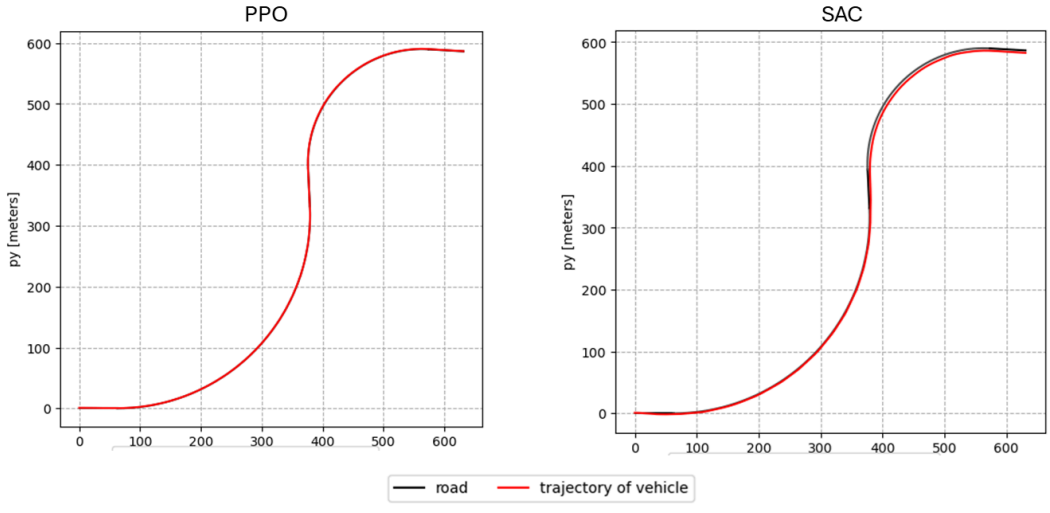


Fig 8: Evaluation road 2 - Slightly curved road

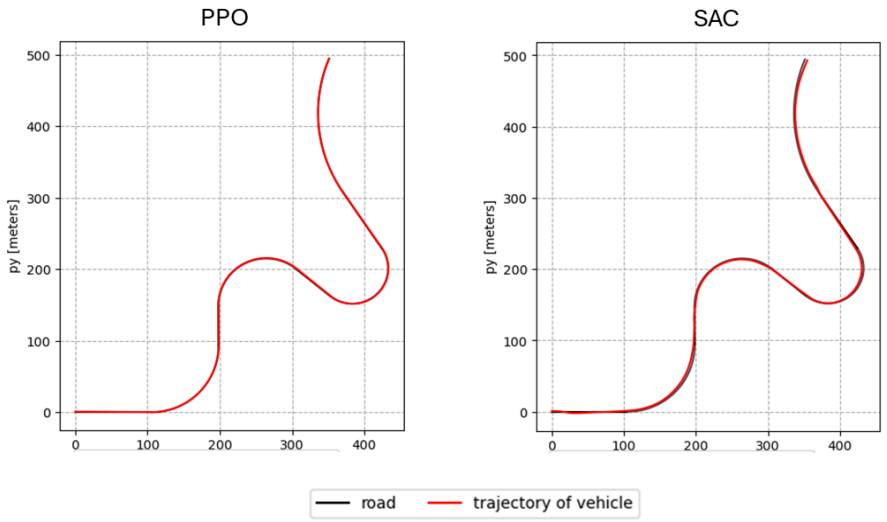


Fig 9: Evaluation road 3 - Challenging road with tight corners