Enhancing PPO for Safe Driving under Noise & Uncertainty

CS378: Arnav Bagad & Jacob Villanueva

As real time perception and control becomes increasingly more important for autonomous driving, systems need to have the ability to make safe decisions under times of uncertainty, however **this is challenging.**

Problem Motivation – 4 Key Areas

Noisy perceptions: Real-world sensors (cameras, lidar) deliver imperfect position/velocity estimates

Unpredictable traffic: Other drivers may behave erratically or suboptimally

Safety vs. performance: Pure RL policies can exploit simulation fidelity, leading to unsafe real-world maneuvers

Need : A decision system that stays close to PPO's efficient learning while guarding against risky actions when inputs are unreliable or traffic is chaotic

Our Objectives

Build noise-aware policies
– Teach PPO to recognize and weigh uncertain observations via confidence scores.

Anticipate other drivers
– Inject simple intent predictions (lane-change hints) into the state for proactive decisions.

Enforce safety online
– Wrap the PPO policy with a light-weight veto that blocks risky lane changes when clearance is low.

Maintain learning efficiency & Modularity
– All enhancements layer on top of standard PPO so we retain fast, stable training while boosting real-world robustness.

Approximate Robust Control

(Leurent et al., 2019)

Two planning algorithms that maximize worst-case return under model uncertainty—one via an optimistic tree search, the other via interval bounds

Only works with known dynamics sets and offline planning; it doesn't learn a policy that adapts online

Highway-env Simulator

(Farama-Foundation, 2020)

A flexible 2D driving environment with kinematic observations and discrete meta-actions

No built-in support for sensor noise or safety filters—agents assume perfect state information

Proximal Policy Optimization
(Schulman et al., 2017)

A clipped surrogate objective that stabilizes policy-gradient updates

PPO alone doesn’t account for observation uncertainty or enforce safety constraints during deployment

Prediction-Augmented DQN
(Yildirim et al., 2022)

Injects predicted lane-change intentions into the DQN state to improve merging decisions

Tailored to a specific merging task and assumes accurate intent predictions—no handling of noisy sensor data

Hybrid RL + MPC

(Kimura et al., IFAC 2022)

Combines a learned policy with an MPC safety filter that vetoes dangerous maneuvers

MPC adds runtime overhead and isn’t integrated into the learning loop, so the policy itself remains unaware of safety constraints when training

Parameterized Skill Priors

(Wang et al., RSS 2023)

Trains over a small set of high-level skills (e.g., “overtake,” “follow”) to speed up learning and improve interpretability

Skills are hand-designed and fixed; there’s no mechanism to discover or adapt new skills based on data

Step 1:
Clean PPO Baseline

Step 2:
Observation Noise

Step 3:
Intention-Augmented DQN

Step 4:
Hybrid Safety Override

Step 5:
Skill-Level Abstraction

Step 6:
Visualization & Reporting

Results

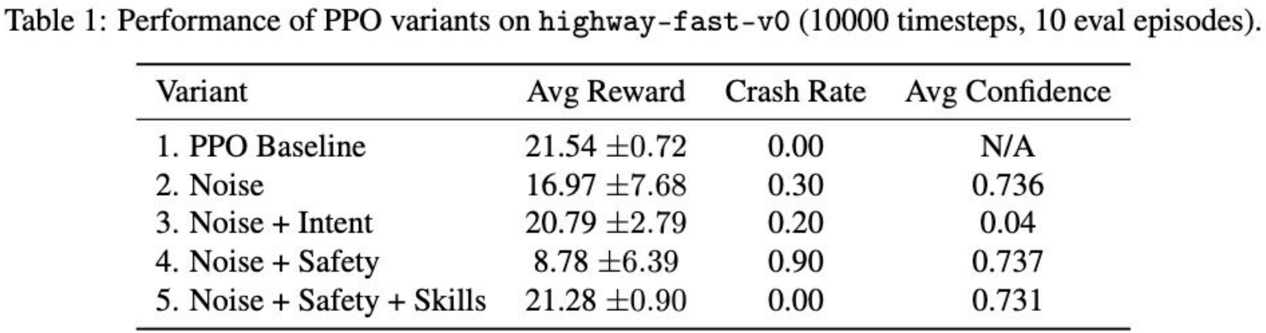


Table 1: Performance of PPO variants on `highway-fast-v0` (10000 timesteps, 10 eval episodes).

Variant	Avg Reward	Crash Rate	Avg Confidence
1. PPO Baseline	21.54 ±0.72	0.00	N/A
2. Noise	16.97 ±7.68	0.30	0.736
3. Noise + Intent	20.79 ±2.79	0.20	0.04
4. Noise + Safety	8.78 ±6.39	0.90	0.737
5. Noise + Safety + Skills	21.28 ±0.90	0.00	0.731

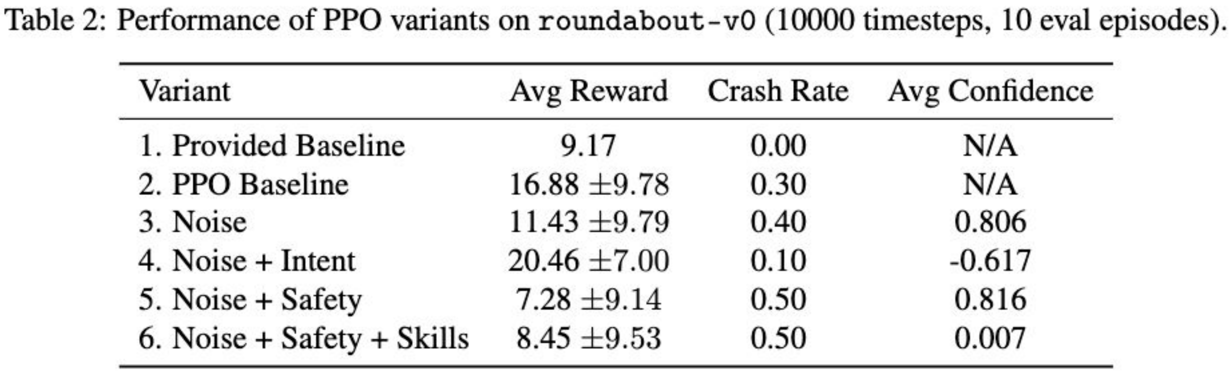
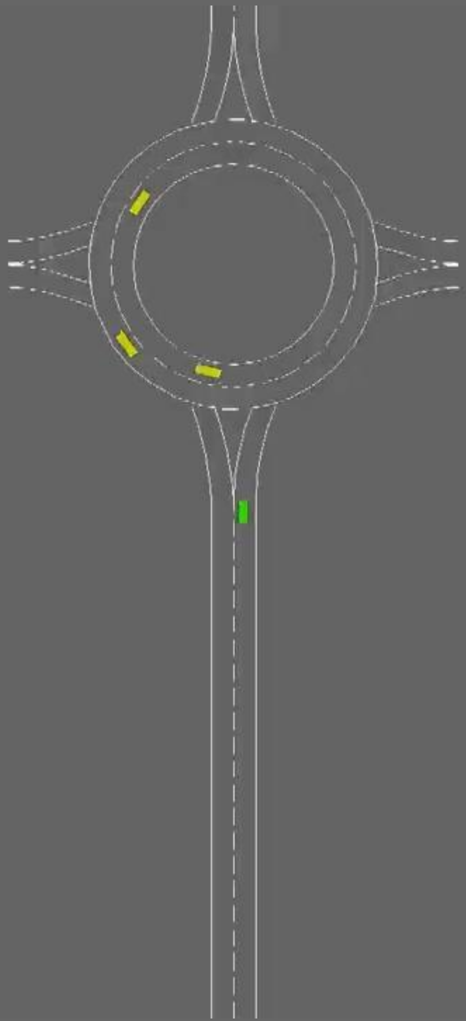
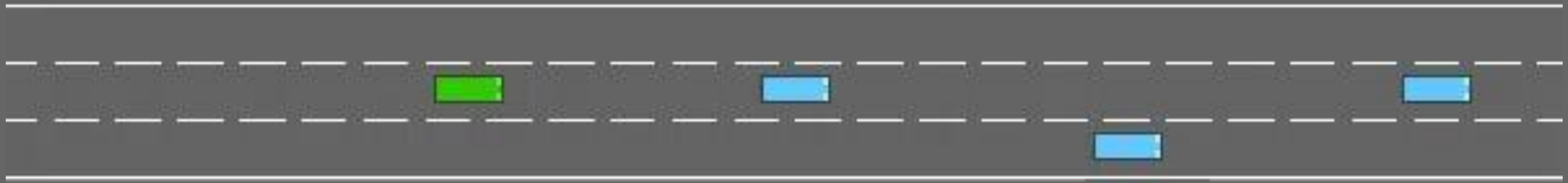


Table 2: Performance of PPO variants on `roundabout-v0` (10000 timesteps, 10 eval episodes).

Variant	Avg Reward	Crash Rate	Avg Confidence
1. Provided Baseline	9.17	0.00	N/A
2. PPO Baseline	16.88 ±9.78	0.30	N/A
3. Noise	11.43 ±9.79	0.40	0.806
4. Noise + Intent	20.46 ±7.00	0.10	-0.617
5. Noise + Safety	7.28 ±9.14	0.50	0.816
6. Noise + Safety + Skills	8.45 ±9.53	0.50	0.007





Your experiences on this project: **BEFORE CS 378**

- RL understanding was mainly textbook-level (Q-learning, basic policy gradients)

- Little hands-on experience with complex simulators or gym environments

- Naive implementations without safety checks or uncertainty modeling

- Minimal familiarity with vectorized training or production-style code organization

Your experiences on this project: **AFTER CS 378**

- Confident using Gymnasium and Stable-Baselines3 for large-scale experiments

- Designed and integrated custom wrappers for noise, intent, and safety overrides

- Learned to structure code for fast parallel training vs. careful evaluation

- Explored advanced techniques like skill abstraction and hybrid RL+rules

Thank You