Final Project Phase 1 Report

Antonio Jimenez and Rohan Damani
CS 378: Geometric Foundations of Data Science
Professor Chandrajit Bajaj
3 May 2025

1 Base Paper Exploration

The base paper [1] addresses the challenge of designing safe control policies for large-scale nonlinear systems operating in uncertain environments. In the standard optimal control framework, the goal is to find a policy that maximizes expected performance. In contrast, the robust control framework used in the paper seeks a policy that performs well under the worst-case scenario, maximizing the minimum expected return across a set of plausible models (the ambiguity set). This guarantees a lower-bound performance when the policy is deployed on the true system.

The paper introduces two algorithms: Deterministic Robust Optimistic Planning and Interval-Based Robust Control. The first algorithm is suitable for problems with a finite ambiguity set and a discrete action space. It extends optimistic planning methods by exploring action sequences and evaluating their worst-case returns. The algorithm provides a guarantee of convergence: As the number of planning iterations increases, the worst-case return of the policy approaches that of the true robust optimal policy. An upper bound on regret (the performance gap between the learned and optimal robust policy) is also established.

The second algorithm, Interval-Based Robust Control, handles settings where the ambiguity set is continuous. It uses interval arithmetic to compute a conservative over-approximation (interval hull) of the set of reachable states under all possible dynamics. This enables the evaluation of a surrogate robust objective, which is easier to optimize. The resulting policy is guaranteed to achieve at least the surrogate's value, offering a certified safety guarantee, though potentially at the cost of some optimality.

The loss function optimized in the paper is the worst-case expected return. This can be formulated as a min-max problem:

$$\max_{\pi} \min_{P \in \mathcal{P}} \mathbb{E}_{\pi,P} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right]$$

where π is the policy, $\mathcal{P}$ is the set of plausible transition models, and $r(s_t, a_t)$ is the reward function. This formulation ensures the policy is robust to model uncertainty, making it well-suited for safety-critical applications.

1.1 Highway Environment

The experiment described in the base paper is made accessible via a Jupyter Notebook hosted on Google Colab. The core package utilized is the highway-environment, which interacts with OpenAI Gym's API in order to provide realistic driving behavior, multi-agent interaction, setup environments, and other tools for enabling training and evaluation of different driving policies in varying environment configurations. The notebook also uses PyTorch for building and training neural networks and incorporates the rl-agents repository to access various other agents which are generated using a collection of reinforcement learning algorithms such as Value Iteration, Deep Q-Networks (DQN), and Robust MDP solvers.

The environment utilized in the base paper, roundabout-v0, presents a complex multi-agent navigation scenario requiring continuous control on a roundabout. The observation space utilized is the kinematics observation space, which is characterized by a VxF matrix where V represents the number of nearby vehicles and F represents the features of those vehicles. The features are the vehicles' x and y positions and their velocity in the x and y direction. The action space selected is Discrete Meta-Actions which allows for the agent to change to the left or right lane, move faster or slower, or maintain the current state.

The behavior of the other vehicles in the environment is represented by the IDMVehicle class where the vehicle's longitudinal behavior is given by the Intelligent Driver Model found in [2] which provides a balance between reaching the desired velocity and maintaining a safe time gap. The lane change decisions are given by the Minimizing Overall Braking Induced by Lane change (MOBIL) model from [3]. The behavior parameters for each vehicle are randomly sampled from a predefined set. Through this approach the agent is exposed to varying levels of aggressive driving, while still experiencing realistic human driving behavior. Lastly, the base paper's setup does not account for uncertainty resulting from the various sensors.

Having configured the environment, we see the agent produce a batch of experiences consisting of the current state, the action taken, and the resulting state where the action taken is randomly selected. The reward function is setup such that the agent is rewarded for driving fast along a planned route while avoiding collisions. The configuration within the Colab is as follows:

$$R(s_t, a_t) = -\mathbf{1}_{\text{collision}} + 0.2 \cdot \mathbf{1}_{\text{high-speed}} - 0.05 \cdot \mathbf{1}_{\text{lane-change}} \tag{1}$$

The authors of the base paper create a python script to compare the performance of various agents across 100 episodes, where they can find the episode with the worst reward and the average reward across all episodes for each agent.

2 Method Exploration

This section will systematically examine some potential strategies proposed by various sources of literature. These investigated methods vary in terms of their degrees of complexity, feasibility, and the specific areas of uncertainty that they address.

2.1 Method 1: Deep Reinforcement Learning with Motion Predictive Safety Controller [4]

This strategy integrates deep reinforcement learning, specifically the Soft Actor-Critic (SAC-Discrete) algorithm with a predictive safety controller. The SAC algorithm is very well suited for dealing with high-dimensional continuous state spaces with efficient exploration and exploitation. The integrated safety controller employs a kinematics-based predictive model to predict the trajectory of nearby vehicles, detecting potential collisions, and replacing unsafe actions dynamically with safer alternatives. This approach is particularly well-suited to handling both unexpected driver behavior (Scenario 1) and vehicle state tracking errors (Scenario 2) since it formally integrates multi-object tracking (MOT) prediction and uncertainty estimation into decision-making. Its real-time computation capability is very helpful for on-road deployment. Nonetheless, the efficiency of this technique mainly depends on the precision and stability of the predictive safety model, however, and it might introduce conservative biases during implementation.

2.2 Method 2: Data-driven CRITICAL Scenario Generation with LLM Integration [5]

The CRITICAL system is the first system that uses a data-driven method for identifying and systematically constructing critical driving scenarios. Initially using clustering techniques to classify and establish diverse driving behaviors from real-world datasets, CRITICAL uses surrogate safety measures to discover highly critical situations. A Large Language Model (LLM) also improves scenario generation from insights learned in the process of learning from prior driving experience. This particular method targets especially the improvement of robust policy learning under uncertain and risky driver behaviors (Scenario 1). Unfortunately, this approach is extremely reliant on the quality and amount of previous driving data. Moreover, the increased complexity from the LLM makes it harder to interpret scenarios, potentially making debugging and tuning during implementation more challenging.

2.3 Method 3: Meta Reinforcement Learning (MRL) [6]

This method involves using Meta Reinforcement Learning with Model-Agnostic Meta-Learning (MAML) and Probabilistic Embeddings for Actor-critic Reinforcement Learning (PEARL) to allow for rapid adaptation in changing traffic conditions. MRL agents are typically trained across a large set of systematically different simulated environments and learn an adaptive policy that shifts quickly to new situations through minimal fine-tuning. This method indirectly responds to unpredictability of driver actions as well as to state estimation inaccuracies (both scenarios 1 and 2) by incorporating the vehicle with inherent capability for high-speed adaptation. While the approach does try to significantly improve generalization, it is very costly in terms of compute and requires a very diverse dataset for true generalization. In addition, tuning the hyperparameters to the level required would be a significant practical challenge when implementing.

2.4 Method 4: SMART Multi-Agent Recurrent Trajectory Prediction [7]

The SMART method uses a Convolutional Long Short-Term Memory (ConvLSTM) network and Conditional Variational Autoencoders (CVAE) for multimodal trajectory prediction of various vehicles interacting with each other. Through this explicit modeling, SMART enhances the accuracy of multi-object tracking (MOT) predictions directly, which is beneficial for Scenario 1. Despite this accurate prediction strength, integrating this within a reinforcement learning pipeline is not as straightforward as other approaches. The output of SMART forecasting also requires other methods to convert these trajectory predictions into effective actions. It is also very computationally intense and complex, making it difficult to use in real-time.

2.5 Method 5: Normalizing Flow-Based Sensor Noise Modeling

Normalizing Flow (NF) models are generative methods for directly modeling sensor noise distributions through transformations of simpler base distributions (e.g., Gaussian). Incorporating NF-based sensor noise modeling into the reinforcement learning itself, autonomous driving agents are provided with greater levels of awareness regarding uncertainties in the estimation of states (Scenario 2), and multi-object tracking and sensor fusion accuracy is greatly improved. The method requires large quantities of accurately labeled sensor data for training, and also large amounts of compute for real-time inference. NF models, however, offer lots of flexibility and accuracy in capturing complex multimodal distributions characteristic of real sensor noise. This makes them extremely suitable to applications demanding tracking accuracy, such as in this project.

2.6 Method 6: Deep Q-Network (DQN) Approach [7]

A Deep Q-Network approach estimates optimal action-value functions with neural network approximations. With the possibility of adding uncertainty-aware prediction architectures inspired by SMART (ConvLSTM + CVAE), the DQN approach would be able to address uncertainty due to driver behavior and inaccurate tracking (Scenarios 1 and 2). However, DQN models tend to suffer from instability and convergence problems, requiring heavy hyperparameter tuning, watchful reward engineering, and potentially additional techniques to ensure robust learning. Also, adding uncertainty modeling could double or triple the complexity.

2.7 Table Summary

Table 1: Summary of Potential Methods for Robust Autonomous Driving Decision-Making

Method	Model Type	Scenarios Addressed	Complexity
Deep RL with Motion Predictive Safety Controller)	Hybrid (Model-free RL, Model-based Prediction)	Driver Behavior & Tracking Uncertainty	Medium
Critical Scenario Generation with LLM	Model-free (Scenario Generation)	Driver Behavior Uncertainty	Medium/High
Meta Reinforcement Learning for Rapid Adaptation	Model-free (Meta-learning)	Driver Behavior & Tracking Uncertainty	High
SMART Multi-Agent Trajectory Prediction	Model-based (Prediction only)	Driver Behavior Uncertainty	High
Normalizing Flow-Based Sensor Noise Modeling	Model-based (Generative)	Tracking Uncertainty	Medium/High
Deep Q-Network (DQN) Approach	Model-free	Driver Behavior & Tracking Uncertainty	Medium

3 Proposed Solution and Experimental Plan

When searching for model type we focused on model-free approaches as model based approaches result in model bias, which has been shown to drastically impact policy performance [1]. We recognize that there are several limitations within the baseline SAC-Discrete algorithm, especially with regard to sensor accuracy and the uncertainties around predicting vehicle states. We propose extending the original framework of the paper by explicitly modeling sensor noise using a generative approach, specifically through Normalizing Flows (NF). These are advanced probabilistic models capable of representing sophisticated, multimodal distributions present in real-world sensor noise. By using NF to model observational uncertainties, we would have a probabilistic representation of sensor readings directly in our RL process. Now, the agent would be given probabilistically-educated states, significantly improving its ability to make proper decisions in the face of uncertainty.

Our approach aims to integrate these NF-computed uncertainty estimates into the predictive safety controller and the SAC-Discrete reinforcement model directly. Sensor measurements of position and velocity based on the highway environment will first be passed through the previously trained Normalizing Flow model, which will give us uncertainty-aware state distributions. These descriptions will in turn provide the predictive safety controller's collision-prediction model with information to account for uncertainty. Thus, the SAC-Discrete algorithm should perform better in terms of state representation, allowing the policy to deal with uncertainty and learn consistently. This approach also specifically addresses Scenario 2, enhancing dependability in MOT and decision safety.

Regarding an experimental plan, here is a high-level outline:

1. **Data Collection and NF Training:** First, we will collect high-level observational data in the simulated environment, taking in the sensor readings with noise. Then, we can train Normalizing Flow models on this collected data to model sensor noise patterns.
2. **Baseline Testing/Validation:** Prior to implementing the NF uncertainty estimates, we will first take baseline results with the SAC-Discrete algorithm and predictive safety controller to ensure that our implementation is accurate. This will also allow for comparisons in the future.
3. **Integration:** Now, we can incorporate the NF-based representations into the predictive safety controller and RL agent in a systematic manner. We can also compare performance with and without these uncertainty measurements.
4. **Performance Evaluation:** We will run comparative evaluations to look at safety metrics (such as collision rates), driving efficiency metrics (such as average speed or merging), as well as robustness when confronted with high sensor noise levels. Our hypothesis is that the

NF-integrated model is expected to outperform the baseline when confronted with scenarios of greater observational uncertainty.

Overall, our extension with Normalizing Flows to model sensor noise seems to be a realistically implementable addition to the baseline SAC-Discrete and predictive safety controller solution. We hope that this combination can greatly enhance the decisions made by the autonomous vehicle especially when dealing with these realistic scenarios of significant sensor noise and uncertainty.

References

[1] Leurent, E., Blanco, Y., Efimov, D., & Maillard, O.-A. (2019) Approximate Robust Control of Uncertain Dynamical Systems. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, pp. 1–12.

[2] Alexander, J.A. & Mozer, M.C. (1995) Template-based algorithms for connectionist rule extraction. In G. Tesauro, D.S. Touretzky and T.K. Leen (eds.), *Advances in Neural Information Processing Systems 7*, pp. 609–616. Cambridge, MA: MIT Press.

[3] Bower, J.M. & Beeman, D. (1995) *The Book of GENESIS: Exploring Realistic Neural Models with the GEneral NEural SImulation System.* New York: TELOS/Springer–Verlag.

[4] Liu, Q., Dang, F., Wang, X., & Ren, X. (2022) Autonomous Highway Merging in Mixed Traffic Using Reinforcement Learning and Motion Predictive Safety Controller. *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1063–1070.

[5] Tian, H., Reddy, K., Feng, Y., Quddus, M., Demiris, Y., & Angeloudis, P. (2024) Enhancing Autonomous Vehicle Training with Language Model Integration and Critical Scenario Generation. *arXiv preprint arXiv:2404.08570*.

[6] Zhang, S., Wen, L., Peng, H., & Tseng, H.E. (2021) Quick Learner Automated Vehicle Adapting its Roadmanship to Varying Traffic Cultures with Meta Reinforcement Learning. *arXiv preprint arXiv:2104.08876*.

[7] N N, S., Liu, B., Pittaluga, F., & Chandraker, M. (2020) SMART: Simultaneous Multi-Agent Recurrent Trajectory Prediction. *European Conference on Computer Vision (ECCV)*, pp. 1–16.