# Provably Safe Prediction for Autonomous Vehicles Using Uncertainty Estimations

Michael OBroin Adrian Kager
https://github.com/michael-obroin/424-project

## Description of System

Previously for our project whitepaper, we wanted to do a project related to training reinforcement learning agents to successfully navigate to an objective. However, we were unable to get any code to work for this (IBM VSRL), and we judged that having to implement similar results on our own would have greater complexity than would be feasible for this class project. As a result, we changed our project idea to a deterministic controller with an ML model acting as a predictor on the behavior of another agent. We then make bounds on the uncertainty of the predictions of this model and make verifiably safe decisions to avoid the pedestrian assuming those bounds.

For our project, we want to consider a simplified model of a car traveling down a street using predictions of the time at which a pedestrian will cross the street in order to make choices about when to brake or accelerate to avoid them. In addition to this keymaera model, we wrote python code to generate data and train a neural network model on it to predict the time at which a pedestrian would cross the road, and estimate the uncertainty of the predictions to inform our controller choices. There are two different methods for doing this uncertainty estimation - training several neural networks on the same data and seeing how different their predicted answers are, or a more sophisticated technique involving “dropping out” neurons of the neural network with a certain probability as a kind of monte carlo method, and using the different predictions with the dropout to estimate the uncertainty. An intuition behind this method is that the model output of inputs that have been seen more by the network will be more robust to dropout of nodes and thus a higher variance in outputs when using dropout indicates that the model is less confident about the result.

We think that this project is interesting to pursue because methods for ensuring safety of systems that utilize machine learning is key for more widespread adoption of ML in safety critical applications such as self driving cars. We believe our work is somewhat novel since there is work in the literature on proving controllers that utilize machine learning models for prediction as well as prediction uncertainty estimates to build better controllers, but we believe combining these two aspects with a provably safe ML prediction model using uncertainty estimations is novel. In addition, we don’t believe we’ve seen this kind of combination of machine learning and cyber-physical systems in a previous class project.

## Model

Our first model is included in Proposal_Final_Project_Stopsign_Proof.kyx. We modeled the system as a single-choice, one-dimensional model, where the car and pedestrian are points and the car moves along a one-dimensional line. The pedestrian is located at some distance

CrossWalkPos away from the car, and we have a prediction for the time at which the pedestrian will choose to cross the road, along with an uncertainty. We prove that if the pedestrian crosses the road within ±EstimatedUncertainty of the estimated cross time then the car will not hit the pedestrian (ie be at the same point as the pedestrian).

As a fallback stepping stone, we were going to try a model without non-deterministic choices if the proof for this model proved too difficult, but once we formulated it with the correct assumptions the proof followed fairly easily.

On the machine learning side, we wrote in Python using the PyTorch and NumPy libraries for data generation and neural models. We separated our code into two files: data_generation.py, and networks.py, where we store our helper functions for data generation and the code for defining, training, and evaluating our neural networks respectively. More detail about our data generation and neural models are below in the stepping stones section.

We started with this relatively simple model as a proof of concept for combining a probably safe controller with a ML prediction algorithm with uncertainty estimate and plan to iteratively increase complexity of the data generation method, the neural network and uncertainty estimation techniques and the model of the car and pedestrian interaction itself.

## Stepping Stones

We have a number of different stepping stones that we are interested in trying to accomplish to make our system more complex and interesting. Broadly speaking, these lie under either our ML models or the cyber-physical model itself. To facilitate this kind of development, we are committing all of our progress, both python and keymaera, to a git repository so that we can more easily compare versions and roll back changes if necessary. The repository is currently private, but we can make it public (if that's permissible for academic integrity reasons), or add collaborators if you wish to view it.

- Experiment with more ML models and better training schemes. At present, we have mostly experimented with a few linear layers with and without dropout and ReLU activation functions.
- Create a significantly large number of training and validation results and run multiple iterations of the ML model on it. Currently, we are just generating new data each time which is not what we want in order to have some degree of reproducibility.
- Try both Bayesian inference and ensemble methods for estimating network uncertainty and compare their results.
- Adjust parameters of data generation to be more realistic. Our first data generation function is fairly naive and distributes the attributes independently. In addition, we made some assumptions on the values the attributes could take on, and want to take a closer look to ensure "reasonable" values.
- Incorporate "real-world" constraints on the model assumptions, reflecting things like reasonable car velocities, distances between crosswalks, braking speeds, and pedestrian speeds.

- Weaken our assumptions on the uncertainty of the predictions (specifically that the EstimatedUncertainty is smaller than CrossTime). This will also tie in with iterating on our ML models and refining our uncertainty predictions, as we will be able to give reasonable estimates of uncertainty for a given input. Presently, our model does not have the most realistic assumptions on uncertainty on outlier values of the time prediction (i.e. for small values of CrossTime the EstimatedUncertainty must also be small).
- More closely model the physical world, and make the cars and pedestrians no longer simple point masses. First we want to have a model with a “buffer zone” in 1D motion then “buffer radius” and straight line motion in 2d plane for both pedestrian and car.

## Literature Review and Related Work

Upon searching through the course website for previous projects, we discovered two that are most relevant, "Verified Cruise Control System on RC Vehicle" by Shashank Ojha and Yufei Wang in Fall 2019, and “Formal Verification of V2I aided Autonomous Driving: A Hybrid Systems Approach” by Ishan Pardesi and Dhruv Mahajan in Fall 2018.

The latter project considered a system where a road communicates the positions of obstacles to a car at certain distance intervals, and the car safely changes lanes to avoid them. Our project differs in that it uses bounds on the predictions of the behaviors of other “agents” and makes its decisions based on those. The first project is more closely related to ours as it also makes decisions under some amount of uncertainty (due to using actual sensor values). Our model will be using ML models in a similar manner as their real-world sensors, and estimate the uncertainty there with different methods, but will not be running on actual hardware.

Elsewhere in the literature, we discovered a number of approaches similar to our project proposal, however their scope is definitely outside that of this project. We chose to concentrate on a smaller, simplified scenario as we have to do much of the groundwork ourselves. Previously for the project whitepaper, we had investigated a number of papers on the subject of reinforcement learning of agents under various safety constraints. We aren’t including these as they are no longer that relevant to our project.

Probabilistically Safe Robot Planning with Confidence-Based Human Predictions

Towards safe machine learning for CPS: Infer uncertainty from training data

Trusted Confidence Bounds for Learning Enabled Cyber-Physical Systems

Safe Control Under Uncertainty

Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning

A General Framework for Uncertainty Estimation in Deep Learning