The Stochastic Road Network Environment for Robust Reinforcement Learning

John Martin[1] Paul Szenher[2] Xi Lin[2] Brendan Englot[2]
[1]University of Alberta [2]Stevens Institute of Technology

Introduction

We introduce the Stochastic Road Network Environment: a simulated domain aimed at modeling stochasticity that influences route-level decision making. Traffic and roadway interactions can be modeled with a simple stochastic reward representative of the travel time across segments of the network. The proposed environment maintains a necessary amount of physical realism by streaming high-dimensional LIDAR scans of complex urban surroundings. Experiments show that robust navigation policies can be learned with modest amounts of data, further suggesting the proposed domain could be a useful platform for autonomous driving research with reinforcement learning. All code needed to use the environment is available in an open source repository at:

https://github.com/RobustFieldAutonomyLab/Stochastic_Road_Network

Robust Reinforcement Learning

This work distinguishes between two kinds of target policies. The *target learning policy* is the policy used in the update rule, and its outcomes are reflected in the learned return distribution, e.g. the standard greedy policy. In embodied learning settings, there can be an additional policy that is used after the learning process stops or is temporarily paused. We call this the *target deployment policy* $\psi \in \Pi$; it is defined as a function of the target learning policy $\psi \triangleq f(\pi)$, which we denote with the shorthand $\psi(\pi)$ in Fig. 1. The deployment policy can be the same as the learning policy—in which case, the robot would learn for some period of time, stop executing its behavior policy, then start executing its target learning policy. However, in settings where robustness is desired, it can be beneficial for the deployment policy to selectively choose which outcomes from the learning policy it would like to realize.

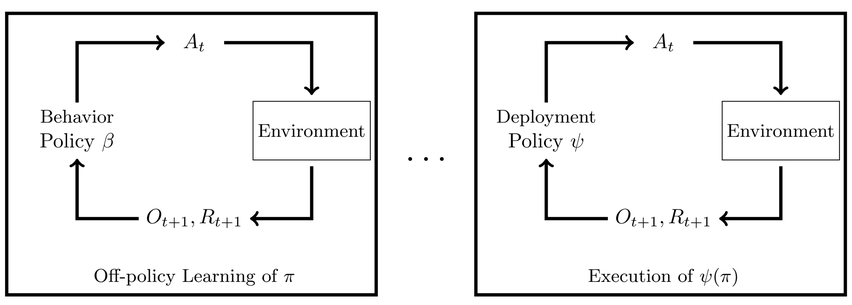


Figure 1: Robust Learning Procedure: Data gathered with a behavior policy β is used to update a target learning policy π. At some point learning will stop, then the system will be deployed with the target deployment policy ψ.

Second-order stochastic dominance

The second-order stochastic dominance (SSD) relation is defined using distribution functions and compared over the set of their realizable values. Consider two returns, G and G', respectively induced by actions a and a'. We say that G stochastically dominates G' in the second order when their integrated CDFs, $F^{(2)}(z) \triangleq \int_{-\infty}^{z} F(x)dx$, satisfy the following equation, and we denote the relation $G \succeq_{(2)} G'$:

$$G \succeq_{(2)} G' \iff F_G^{(2)}(z) \leq F_{G'}^{(2)}(z),\ \forall\ z \in \mathbb{R}.$$

The SSD policy prefers outcome G to G' when $G \succeq_{(2)} G'$. In settings where multiple outcomes do not result in an exact tie between expected values, the SSD policy will be equivalent to the standard greedy policy. Here we also introduce a relaxation to the SSD policy that permits uncertainty to be considered even when there is not an exact tie. The thresholded SSD policy computes the difference in the expected return values of the top two optimal actions. When the difference is less than a certain threshold they will be considered equivalent, and the tie will be broken with a comparison of their second moments which is consistent with the exact SSD policy—namely, the action that induces the smaller second moment will be preferred.

The Stochastic Road Network Environment

The Stochastic Road Network Environment is built upon map structure and simulated sensor data originating from the CARLA autonomous vehicle simulator version 0.9.6, Five such maps are available by default, named in the sequencing of `Town01` to `Town05`. These maps vary in size, with multiple kilometers of roadway defined in each. The maps also vary with respect to the road features present, including complex intersections, multi-lane roadways (allowing for lane changes), and roundabouts. A breakdown of map action spaces, state count, and features is provided in Table 1.

Map	No. Actions	No. States	Features
`Town01`	2	319	Basic
`Town02`	2	156	Basic
`Town03`	4	665	Lane changes
`Town04`	4	1679	Lane changes, Roundabouts
`Town05`	3	1220	Lane changes, Roundabouts

Table 1: Environment map state and action spaces. The action space at each state is a discrete map of candidate state transitions. At states for which the graph topology does not allow for use of the full action space (i.e., in the case of a straightaway, where there is only one valid direction of travel) the remainder of the action space corresponds to a loopback action, whereby the agent re-transitions into its current state. State transitions are fully deterministic, meaning every state-action pair yields a transition to a deterministic subsequent state.

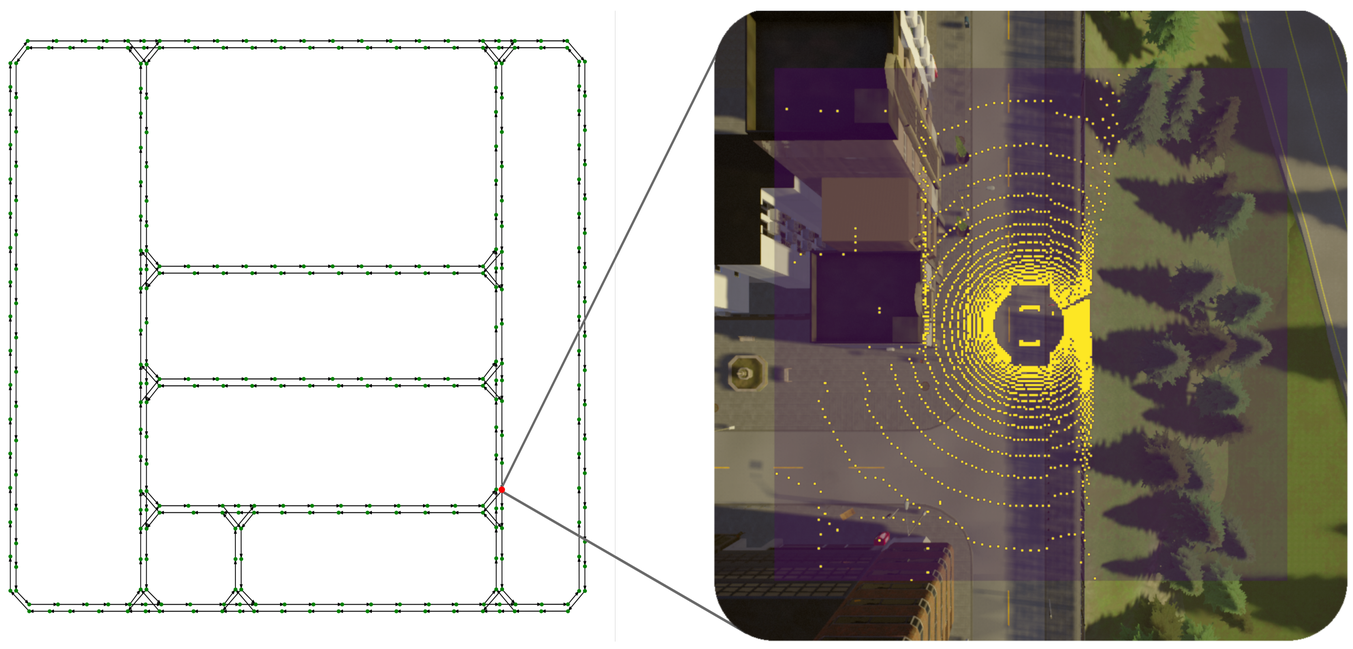
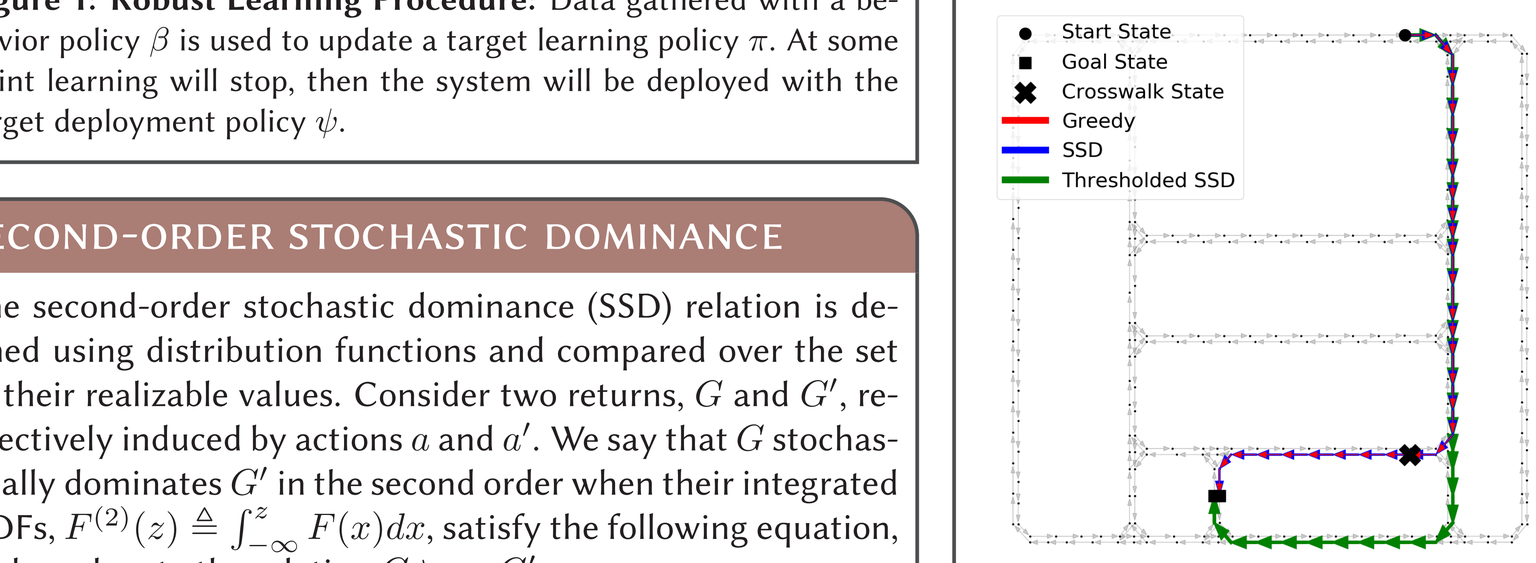
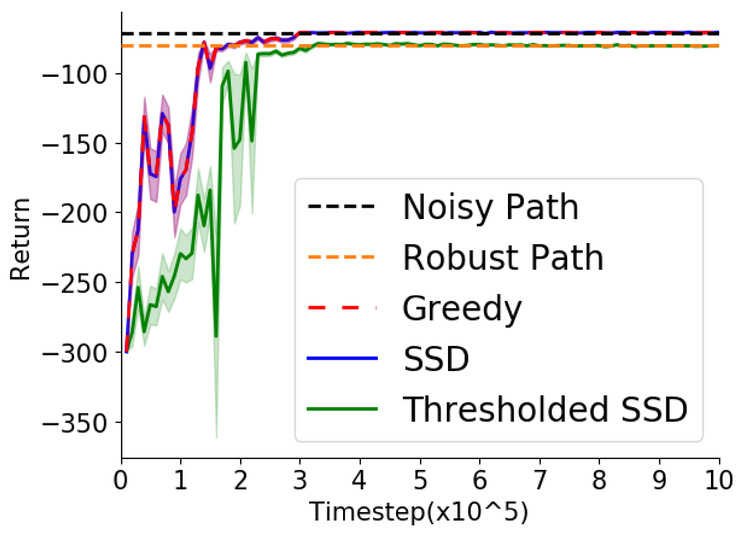


Figure 2: Observation sample from proposed environment. The learning system observations are composed of a 2D binary occupancy grid at ground level, with cell state corresponding to the presence of a LIDAR return within a specific block of space.

Experimental Results



(a) Executed paths in Town 1. The path taken during the final evaluation is shown for each deployment policy. Greedy and SSD methods take the noisy shortest path in all trials. Thresholded SSD takes the second-shortest path to avoid unwanted stochasticity in nearly all trials, which is visualized here. The crosswalk state serves as a source of random travel delay, with stochastic rewards reflective of the delays induced by pedestrians crossing the roadway.



(b) Learning curves of QR-DQN with different target deployment policies. We report the mean and standard deviation resulting from 30 trials. As indicated by the returns, both the greedy and exact SSD policies take a path with high stochasticity in all trials. The thresholded-SSD policy sacrifices optimality for robustness, and during the final evaluation, the second shortest path (Robust Path in the plot) is taken in 22 trials, the third shortest path is taken in seven trials, and the shortest path (Noisy Path in the plot) is taken in one trial. Both the second shortest path and the third shortest path are noiseless, and their lengths differ by only one step.

Figure 3: Learning Experiments comparing the performance of different target deployment polices (greedy policy, exact SSD policy, and thresholded SSD policy), measured with the approximate discounted return ($\gamma = 0.99$). Performance measurements are taken during an independent evaluation, at one-hundred equidistant time-steps over a total of one-million steps of learning. This is intended to model the counterfactual scenario of what could happen if learning were paused and the deployment policy was executed.