A Proof of Theorem 1

Proof. We will first establish the bound on the FPR. Note that

$$c^\star_{t+\tau} \geq \hat{c}^{M-n}_{t+\tau} \wedge c^\star_{t+\tau} < c^{\mathrm{a}}_{t+\tau}(p) \implies \hat{c}^{M-n}_{t+\tau} < c^{\mathrm{a}}_{t+\tau}(p) \ . \tag{5}$$

Hence,

$$\mathop{\mathbb{P}}_{S\sim\phi^M}[c^\star_{t+\tau} \geq \hat{c}^{M-n}_{t+\tau} \wedge c^\star_{t+\tau} < c^{\mathrm{a}}_{t+\tau}(p)] \leq \mathop{\mathbb{P}}_{S\sim\phi^M}[\hat{c}^{M-n}_{t+\tau} < c^{\mathrm{a}}_{t+\tau}(p)] \ . \tag{6}$$

Since $\hat{c}$'s are ordered by their value in S, the upper bound in (6) is essentially the probability of sampling $S \sim \phi_{t+\tau}(\hat{c}|o_t)^M$ such that at most n elements of S lie above $c^{\mathrm{a}}_{t+\tau}(p)$, i.e., lie in the p-quantile tail of ϕ. Whether a sample of ϕ lies in the top p-quantile tail or the remainder of the $1-p$ portion of the distribution is Bernoulli distributed. Therefore,

$$\mathop{\mathbb{P}}_{S\sim\phi^M}[\hat{c}^{M-n}_{t+\tau} < c^{\mathrm{a}}_{t+\tau}(p)] = \sum_{i=0}^{n} \binom{M}{i} p^i (1-p)^{M-i} \ . \tag{7}$$

Using (7) in (6) followed by (2) gives the upper bound on FPR.

Analogously, we can bound the FNR. We have the following:

$$c^\star_{t+\tau} < \hat{c}^{M-n}_{t+\tau} \wedge c^\star_{t+\tau} \geq c^{\mathrm{a}}_{t+\tau}(p) \implies \hat{c}^{M-n}_{t+\tau} > c^{\mathrm{a}}_{t+\tau}(p) \implies \hat{c}^{M-n}_{t+\tau} \geq c^{\mathrm{a}}_{t+\tau}(p) \ . \tag{8}$$

Hence,

$$\mathop{\mathbb{P}}_{S\sim\phi^M}[c^\star_{t+\tau} < \hat{c}^{M-n}_{t+\tau} \wedge c^\star_{t+\tau} \geq c^{\mathrm{a}}_{t+\tau}(p)] \leq \mathop{\mathbb{P}}_{S\sim\phi^M}[\hat{c}^{M-n}_{t+\tau} \geq c^{\mathrm{a}}_{t+\tau}(p)] \ . \tag{9}$$

As before, we use the fact that drawing samples from ϕ such that they lie in the p or the $1-p$ portions of the distribution is Bernoulli distributed to write

$$\mathop{\mathbb{P}}_{S\sim\phi^M}[\hat{c}^{M-n}_{t+\tau} \geq c^{\mathrm{a}}_{t+\tau}(p)] = \sum_{i=n+1}^{M} \binom{M}{i} p^i (1-p)^{M-i} \ . \tag{10}$$

Using (10) in (9) followed by (3) gives the upper bound on FNR, completing the proof. □

B Additional Experimental Details

B.1 Comparison methods

We compare our detector against four other approaches: (i) likelihood detection, (ii) uniform and partial degradation tests (UDT and PDT) [4], (iii) a time-to-collision (TTC) based detector, and (iv) detection based on Hamilton-Jacobi (HJ) reachability analysis [1].

Likelihood detection. We label the likelihood of the estimated state $x_{t+\tau}$ as the value in the PDF of the distribution over the predicted non-ego agent states $\psi(\hat{x}^{\mathrm{ne}}_{t+1:t+T}|x_{0:t})$ provided by the prediction module. As such, we directly represent the likelihood of the estimated non-ego positions given the predicted positions. For anomaly detection, we only check agents within 10 m of the ego since this is the region in which trajectory prediction is most accurate, and the non-ego agents are most relevant.

UDT and PDT. These degradation tests use the predicted costs for each agent as a set of reference signals. We use the same cost function as our method. UDT compares a weighted mean of the reference signals to the online signal in order to determine when reward degradation is significant and is optimal when the reference signals come from a multivariate normal distribution. PDT detects deterioration using a subset of the time steps of the reference signal and tends to perform better in practice when the normality assumption does not hold, see [4] for more information on these tests. After predicted costs are computed, we use the false alarm rate control method provided in [4] to choose a threshold for detection based on the desired p-value and the reference (i.e. predicted) costs. If the achieved cost is above the threshold, it is labeled as an anomaly.

TTC detection. For TTC calculations, we assume vehicles are a 1m radius circle and pedestrians are a 0.2m radius circle. We propagate the agents forward in time to determine if there would be a collision at some point in the future if the agents maintained their velocity along the current heading.

HJ reachability. We follow the method for HJ reachability analysis from [1]. We assume the ego and other vehicles follow a four-state bicycle model with the following dynamics

$$[\dot{x}, \dot{y}, \dot{\theta}, \dot{v}] = [v\cos(\theta), v\sin(\theta), \tan(d)/L, a] \tag{11}$$

Where d is the control steering angle, a is the control acceleration, and L is the length of the vehicle. We assume that the vehicles' acceleration lies within $[-2, 1]$ ms^{-2} and the steering angle lies within $[-10, 10]$ degrees. Pedestrians are modeled as agents which can move in any direction with acceleration in $[-0.5, 0.5]$ ms^{-2}. We assume a 1m radius circle for vehicles and a 0.2m radius circle for pedestrians and set the initial value function such that any collision should result in a negative value. We compute the value function offline to improve operation speed. Online, we check the value for each agent with the ego using a lookup table. If the lowest value is below a threshold, we label the scene as anomalous.

B.2 Planning

We follow a similar procedure as in [51] to make a motion plan for the ego. First, we discretize the control space so that we can generate a tree of motion primitive trajectories by integrating along all possible combinations of discrete control actions at each time step over the planning horizon. In particular, we use $[-2, 0, 1]$ ms^{-2} for the acceleration control and $[-0.3, -0.1, 0, 0.1, 0.3]$ rad s^{-1} for the radial velocity control as the discretization. To allow for time to run cost computations, we set the first control action of each primitive to be the most recent control action as in [51]. As such, there are $15^{(T-1)}$ possible motion primitives where T is the planning horizon. We compute the predicted cost for each motion primitive using predictions of all other agents in the scene and select the primitive which has the lowest predicted cost over the planning horizon. We parallelize the predicted cost computations to improve run-time. A cost computation for a single primitive takes about 1 ± 0.5ms and for a planning horizon of 2 seconds (4 control inputs), there are 3375 motion primitives (15^3) which need cost computations. On the 18 core (36 thread) computer we use for simulations, with parallelization, the best motion primitive can be reliably computed in less than 0.25s, which allows for real-time planning.

B.3 Cost functions

Let $x_e, v_e, a_e, j_e, \theta_e, \dot{\theta}_e, \ddot{\theta}_a$ be the position, velocity, acceleration, jerk, heading, rotational velocity, and rotational acceleration of the ego, respectively. Similarly, $x_a, v_a, a_a, j_a, \theta_a, \dot{\theta}_a, \ddot{\theta}_a$ represent the analogous states for the non-ego agents. Note that when we compute predictions of the cost, we use the predicted non-ego agent states instead of the achieved agent states in the cost computation; otherwise, the cost computations are identical for the predicted and the observed costs. We add subscript $\parallel$ or $\perp$ to denote decomposition parallel to or perpendicular to the *ego's heading* (e.g. ego's longitudinal velocity $v_{e,\parallel}$ and agent's lateral acceleration with respect to the ego's heading $a_{a,\perp}$). We define x_{goal} as the position of the end of the reference trajectory (i.e., the goal), and x_r, θ_r are points and headings on the reference trajectory. Let ϵ's represent scaling factors and let ttc represent the time-to-collision function; ttc outputs the time until a collision between the ego and another agent (assuming constant velocity at their current heading direction) and modeling vehicles as a 1 m radius circle and pedestrians as a 0.2 m radius circle. We now express the cost functions we use as follows:

$$c_{\text{ttc}} = 1 - \max_{\text{non-ego agent}} \min\left(\frac{\text{ttc}(x_e, x_a, v_e, v_a)}{\epsilon_{\text{ttc}}}, 1\right) , \tag{12}$$

$$c_{\text{d2a}} = \max_{\text{non-ego agent}} \left[e^{-0.5\epsilon_{\text{rbf}}((x_{a,\parallel}-x_{e,\parallel})^2(v_{a,\parallel}-v_{e,\parallel})^2+(x_{a,\perp}-x_{e,\perp})^2(v_{a,\perp}-v_{e,\perp})^2)}\right] , \tag{13}$$

$$c_{\text{d2g}} = \|x_{\text{goal}} - x_e\| / \|x_{\text{goal}} - x_{0,e}\| \text{ where } x_{0,e} = x_e \text{ at } t = 0 , \tag{14}$$

$$c_{\text{d2r}} = \frac{1}{4}\|x_r^* - x_e\|^4 + \frac{1}{2}(\theta_r^* - \theta_e)^2 \text{ where } x_r^*, \theta_r^* = \underset{(x_r,\theta_r)\in\text{reference trajectory}}{\text{argmin}} \|x_r - x_e\| \tag{15}$$

$$c_{\text{velocity}} = \max(\|v_e - \epsilon_{v,r}\| - \epsilon_r, 0)^2 / \epsilon_{\text{limit}}^2 , \tag{16}$$

$$c_{\text{comfort}} = \frac{1}{6}\sum \max\left(\left[\frac{|a_{e,\parallel}|}{\epsilon_{a,\parallel}} - 1, \frac{|a_{e,\perp}|}{\epsilon_{a,\perp}} - 1, \frac{|j_{e,\parallel}|}{\epsilon_{j,\parallel}} - 1, \frac{\|j_e\|}{\epsilon_j} - 1, \frac{|\dot{\theta}_e|}{\epsilon_{\dot{\theta}}} - 1, \frac{|\ddot{\theta}_e|}{\epsilon_{\ddot{\theta}}} - 1\right], \bar{0}\right) , \tag{17}$$

$$c_{\text{reverse}} = 1[v_{e,\parallel} < 0] \ . \tag{18}$$

In our experiments, we use $\epsilon_{\text{ttc}} = 3$. $\epsilon_{\text{rbf}} = 0.5$ when the agent is a vehicle, and $\epsilon_{\text{rbf}} = 1$ when the agent is a pedestrian. The reference velocity $\epsilon_{v,r} = 0.8\|x_{\text{goal}} - x_{0,e}\|/\text{time of reference trajectory}$, and $\epsilon_r = \max(0.1\epsilon_{v,r}, 1)$, $\epsilon_{\text{limit}} = \max(30 - \epsilon_{v,r}, 10)$. The comfort tuning parameters are based on the built-in comfort metric from nuPlan [6], $(\epsilon_{a,\parallel}, \epsilon_{a,\perp}, \epsilon_{j,\parallel}, \epsilon_j, \epsilon_{\dot{\theta}}, \epsilon_{\ddot{\theta}}) = (2.4, 4.89, 4.13, 8.37, 0.95, 1.93)$ and the $\max$ in (17) refers to element-wise maximum. Along with the tuning parameters, the weights are set as $[w_1, \ldots, w_7] = [1, 10, 1, 1, 1, 0.5, 10]$.

B.4 Hand labeling scenarios

See Fig. 5 for examples of positive and negative hand labels. Fig. 5(a) is an example of a negative label since agents are not close to the ego and predictions are relatively accurate. Fig. 5(b) is an example of a positive label. The circled agents are predicted to slow down or stop before reaching the ego. Instead, these agents get very close to the ego. Lastly, Fig. 5(c) is an example of a negative label with task-irrelevant prediction errors. Non-ego agents which are circled are predicted to turn into the ego's lane, but they do not.

(a) (b) (c)

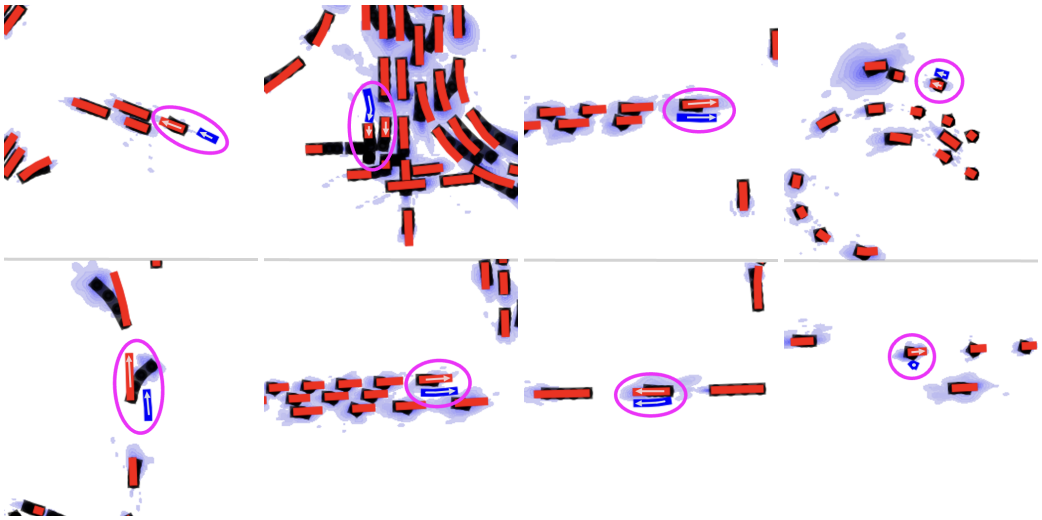
Figure 5: Illustration of hand labeling through selected scenarios from nuPlan. The ego vehicle trajectory is in blue, non-ego vehicle trajectories are in red, arrows indicate direction of travel, Trajectron++ predictions are the purple contour, and the black line is the most likely trajectory given the predictions. Important agents are circled. (a) Example of a negative label. (b) Example of a positive label. (c) Example of a negative label with task-irrelevant prediction errors.

B.5 Additional experimental results: Representative success and failure cases

See Fig. 6 for representative examples of true negatives, true positives, false negatives, and false positives. The images provide qualitative information about where the failure predictor succeeds and fails. The true negatives in Fig. 6(a) show standard driving scenarios and the ability of our detector to ignore task-irrelevant prediction errors. The true positives in Fig. 6(b) are scenarios where other agents are predicted to maintain their speed but speed up or slow down more quickly. The false negatives in Fig. 6(c) are cases where predictions have are incorrect but have larger variance. The images are labeled as positive by the hand-labeling, but the higher variance is enough to prevent a p-quantile anomaly detection from occurring. The false positives in Fig. 6(d) are cases which look like normal driving scenarios, but anomalies are detected due to the very close agents and small variance in predictions.

B.6 Additional experimental results: Adaptive re-planning rate

In order to assess if our detector can feasibly be used for an adaptive re-planning rate, we set up another experiment. Initially, the ego is given a long horizon plan (10 seconds, 20 time steps) from the reference trajectory. We run our quantile anomaly detector with quantile $p = 0.25$ and re-plan once an anomaly is detected. The result is a detector which determines when the predictions



(a) True negatives (TN) (b) True positives (TP) (c) False negatives (FN) (d) False positives (FP)

Figure 6: Illustration of representative (a) true negatives (TP), (b) true positives (TP), (c) false negatives (FN), and (d) false positives (FP) from the nuPlan dataset. The ego vehicle trajectory is in blue, non-ego vehicle trajectories are in red, arrows indicate direction of travel, Trajectron++ predictions are the purple contour, and the black line is the most likely trajectory given the predictions. Important agents are circled.

are incorrect enough such that re-planning may be required. The dataset is made of 651 scenarios which come from 4 classes of scenario types: (i) ego at pick-up/drop-off (PUDO), (ii) ego following vehicle, (iii) ego stopping at traffic light, and (iv) nearby dense vehicle traffic. See Fig. 7 for a cumulative distribution function (CDF) of detections for each scenario type. We also present the mean detection time for each of the scene types in Table 4. We see that the ego following vehicle scene type has the lowest time between re-plans while the ego stopping at traffic light and ego at pick up/drop off both scene types have, on average, longer time between re-planning. The nearby dense traffic scenarios tend to have many more vehicles and the ego following vehicle has many more lane-changes and a large diversity of agent speeds. This results in a larger number of task-relevant prediction failures and we therefore see faster re-planning rate for these scenarios. In contrast, we observe that the ego at PUDO and the ego stopping at traffic light scenarios have many fewer non-ego agents around as compared with the other scenario types and therefore have fewer task-relevant prediction failures. The long average time duration between re-plans in Table 4 suggest the viability of our approach as an effective way to adapt the re-planning rate.

Table 4: Average time between re-plans

Scene type:	Ego at PUDO	Following vehicle	Stopping at light	Nearby traffic
Mean re-plan time	5.18 s	3.55 s	5.23 s	4.49 s

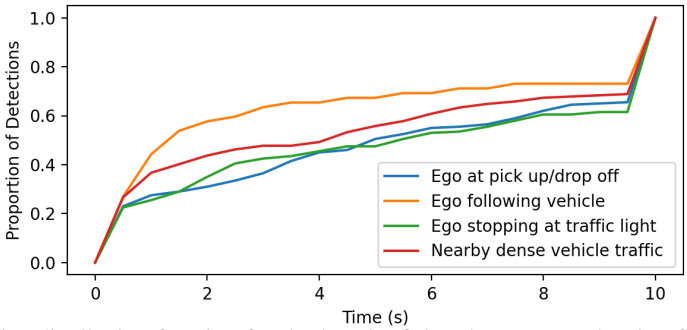


Figure 7: Cumulative distribution function for the length of time between re-planning for various types of scenarios. For ego following vehicle scenes (orange line), there is more frequent re-planning. For the ego stopping at traffic light (green line) and ego at pick up/drop off (blue line) scene types, re-planning is less frequent.