A Sampling an Episode (line-6 of Algorithm 1)

Once a CPOMDP $\mathcal{P}_{\phi,\epsilon_i}$ to use is selected, CAT uses a POMCP-like method [40] to perform episode sampling — a sequence of $\langle s, a, o, r \rangle$, where $s \in S$, $a \in A$, $o \in O$, and $r = R(s, a)$. Each node of $\mathcal{T}$ is represented as a set of state particles, and to sample an episode, CAT starts by sampling a particle from the root node b_0 of $\mathcal{T}$. Suppose the sampled particle is $s \in S$, then CAT selects an action $a \in A$ to use based on UCB1 [3] strategy: $a = \underset{a' \in A}{argmax} \left(Q_i(b_0, a') + c' \cdot \sqrt{\frac{log(N_i(b_0))}{N_i(b_0, a')}} \right)$, where $Q_i(b_0, a')$ is the estimated Q-value for performing a' at belief b_0 under the POMDP problem $\mathcal{P}_{\phi,\epsilon_i}$. This strategy has been shown to enable convergence to the optimal policy [23, 40]. Once an action is selected, CAT samples a next state $s' \in S$ based on $T(s, a, s')$, samples an observation $o \in O$ based on $Z(s', a, o)$, and a reward $r = R(s, a)$ is then incurred. If the pair (a, o) has been used to expand b_0, s' is added to the set of particles representing node b, which is the child of b_0 via (a, o). In this case, if s' is not a terminating state, the sampling process repeats starting from b. Otherwise, backup is performed to revise the value estimate. If the pair (a, o) has not been used to expand b_0, a new node b is added as a child of b_0 in $\mathcal{T}$ via an edge labelled (a, o). A default (roll-out) strategy is then performed to provide an initial value estimate for b, and backup is performed to revise the value estimate of the nodes visited by the sampling process.

B Autonomous Driving Software used in Section 5.2

This section provides a short description of the software systems being used to evaluate the effectiveness of the proposed safety assessment mechanism.

TCP [49] is a camera-only model. By observing that waypoints are stronger at collision avoidance compared to directly predicting controls, it proposes a situation-dependent network with two branches which generates the waypoints and control signal respectively. During run time, the two outputs are generated with a weighted average that varies based on whether the vehicle is turning.

NEAT [8] proposes neural attention fields which enables reasoning for end-to-end imitation learning. It uses imitation-learning with attention and implicit functions to iteratively compress high dimensional 2D image features into a compact bird-eye-view representation for driving. The attention mechanism has been demonstrated to be a powerful module, however the utilization of a relatively dense representation drastically increases model complexity.

AIM [33] takes the birds-eye-view of the target location as an input, similar to NEAT, which is then sent to a ResNet 34 encoder pre-trained on ImageNet. It outputs waypoints through four GRU decoders followed by PID controllers. Adding auxiliary tasks during training such as using a deconvolutional decoder to predict the 2D depth and semantic segmentation is shown to increase driving performance.

TF++ [20] is an improved variant of Transfuser [9] by modifying its architecture, output representation and training strategy. It uses a transformer decoder for pooling features to mitigate out of distribution errors that may arise when steering directly towards a target point. It also considers the prediction uncertainties into the final output by using a confidence weighted average of the predicted target speed as input to the controller as an attempt to reduce collisions.