Appendix

6.1 Proofs

Proposition 1: Under Assumption 1, Algorithm 1 is $\epsilon + 1/(1+|\mathcal{A}|)$-safe (with respect to $\hat{Y}, \hat{Z}$).

Proof. Given a sequence of data points $(Y_1, Z_1), \cdots, (Y_T, Z_T)$, denote the subsequence of "unsafe" data as $(Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M})$ where Z_{c_t} is the t-th unsafe example (i.e. $f(Z_{c_t}) < f_0$), so $M = |\mathcal{A}|$. Suppose that $\hat{Z}$ is also unsafe, i.e. $f(\hat{Z}) < f_0$. Let $\wr \cdot \int$ denote an unordered bag (i.e. it is a set that can have repeated elements). We can bound the safety by

$$\begin{aligned}
\Pr[w(\hat{Y}) = 0] &= \Pr[q > 1 - \epsilon] \\
&= \mathbb{E}\Big[\Pr\big[q > 1 - \epsilon \mid \wr (Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M}), (\hat{Y}, \hat{Z}) \int\big]\Big] && \text{Tower} \\
&= \mathbb{E}\Big[\Pr\big[|\{t \mid g(Y_{c_t}) < g(\hat{Y})\}| + U > (1-\epsilon)(M+1) \\
&\qquad - 1 \mid \wr (Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M}), (\hat{Y}, \hat{Z}) \int\big]\Big] && \text{Definition}
\end{aligned}$$

By the assumption of exchangeability we are equally likely to observe any permutation of $\wr (Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M}), (\hat{Y}, \hat{Z}) \int$. Intuitively, it is equally likely for $g(\hat{Y})$ to be the largest, 2nd largest, etc, among $g(Y_{c_1}), \cdots, g(Y_{c_M}), g(\hat{Y})$. Formally, the random variable $|\{t \mid g(Y_{c_t}) < g(\hat{Y})\}| + U$ takes on all values $\{0, 1, \cdots, M\}$ with equal probability. Therefore,

$$\begin{aligned}
&\Pr\Big[|\{t \mid g(Y_{c_t}) < g(\hat{Y})\}| + U > (1-\epsilon)(M+1) \\
&\qquad - 1 \mid \wr (Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M}), (\hat{Y}, \hat{Z}) \int\Big] \\
&= 1 - \Pr\Big[|\{t \mid g(Y_{c_t}) < g(\hat{Y})\}| + U \leq (1-\epsilon)(M+1) \\
&\qquad - 1 \mid \wr (Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M}), (\hat{Y}, \hat{Z}) \int\Big] \\
&\leq 1 - \frac{\lceil (1-\epsilon)(M+1) - 1 \rceil}{M+1} \\
&= \frac{\lfloor M + 1 - (1-\epsilon)(M+1) + 1 \rfloor}{M+1} \\
&\leq \frac{1 + \epsilon M + \epsilon}{M+1} \\
&= \epsilon + \frac{1}{M+1}
\end{aligned}$$

We can combine this with the original result to get

$$\begin{aligned}\Pr[w(\hat{Y})=0] &= \mathbb{E}\Big[\Pr\big[|\{t \mid g(Y_{c_t}) < g(\hat{Y})\}| + U > (1-\epsilon)(M+1) \\ &\qquad - 1 \mid \wr(Y_{c_1}, Z_{c_1}), \cdots, (Y_{c_M}, Z_{c_M}), (\hat{Y}, \hat{Z})\wr\big]\Big] \\ &\leq \mathbb{E}\left[\epsilon + \frac{1}{M+1}\right] \\ &= \epsilon + \frac{1}{M+1}\end{aligned}$$

6.2 Lower bound on the false positive rate

Consider a function w that maps a dataset $\mathcal{D} = (g(X_1), Y_1), \cdots, (g(X_T), Y_T)$ of unsafe examples, and a new data point $g(\hat{X})$, to $\{0, 1\}$. We argue that any w that gives a distribution-free false negative rate guarantee should depend only on the ordering between $g(X_1), \cdots, g(X_T), g(\hat{X})$, and not on their specific values. In other words, w should take the form defined by

$$w(\mathcal{D}, \hat{Z}) = \begin{cases} \phi\left(\#\{t, g(\hat{X}) < g(X_t)\}\right) & \text{with probability } \gamma \\ 1 & \text{with probability } 1-\gamma \end{cases} \tag{6}$$

for some deterministic function ϕ and real number γ. We know that when the data is exchangeable, $\#\{t, g(\hat{X}) < g(X_t)\}$ is uniformly distributed on $\{0, 1, \cdots, T\}$.

Case 1 Suppose ϕ takes the value 0 for at least one possible input; then the false negative rate is given by

$$\text{FNR} \geq \gamma/(1+T) \tag{7}$$

and the false positive rate is given by

$$\text{FPR} \geq 1-\gamma \tag{8}$$

so combined we have

$$\text{FPR} \geq 1-\gamma \geq 1-(1+T)\text{FNR} \geq 1-(1+T)\epsilon \tag{9}$$

Case 2 Suppose ϕ takes the value 0 for none of the inputs; then the false negative rate is given by

$$\text{FNR} = 0, \text{FPR} = 1 \tag{10}$$

so we would still (trivially) have $\text{FPR} \geq 1-(1+T)\epsilon$.

So far we have shown that if w were to take the specific form of Eq. (6), then the false positive rate must be lower bounded by $1-(1+T)\epsilon$. In other words, when $\epsilon = o(1/T)$, the false positive rate tends to 1 when T is large.

6.3 Additional Experimental Details: Driver Alert System

Safety score: We define the safety score by the Mahalanobis distance between the ego-vehicle and the agent, where the first eigenvector is aligned with the ego-vehicle's velocity vector, and the second eigenvector is orthogonal to the ego-vehicle; the magnitude of the first eigenvector is the magnitude of the velocity, and the magnitude of the second eigenvector is approximately half of a car width (we use 1m). Intuitively, this means that agents that are along the ego-vehicle's velocity vector appear closer than agents in the perpendicular direction. This metric is similar to time to collision (TTC), but it is continuous whereas TTC is not — TTC is infinite unless two vehicles are exactly on a collision course.

Dataset details: The nuScenes dataset includes 952 scenes collected across Boston and Singapore, divided into a 697/105/150 train/val/test split (the same split used for the original Trajectron++). Each scene is 20 seconds long. The Kaggle Lyft Motion Prediction dataset is a subset of the full Lyft Level 5 dataset (chosen over the full dataset for computational reasons). It includes approximately 16k scenes, divided into an 70%/15%/15% train/val/test split. Each scene is 25 seconds long. Both datasets include labeled ego-vehicle trajectories as well as labeled detections and trajectories for other agents in the scene. Note that for both of these datasets, because the training split was used to train the Trajectron++ model, we used the validation split as the input training data for Algorithm 1.

Additional experimental results: We demonstrate empirically on the nuScenes dataset that the sum of ϵ and the false positive rate must be high when there are few (e.g. $< 1/T$) samples, which is consistent with what our theory from Section 3.2 would predict. Figure 4 plots the epsilon bound as well as the false negative and false positive rates vs. the number of unsafe samples in the validation dataset; we see that when ϵ decreases as $1/T$, the false positive rate is relatively flat and low.

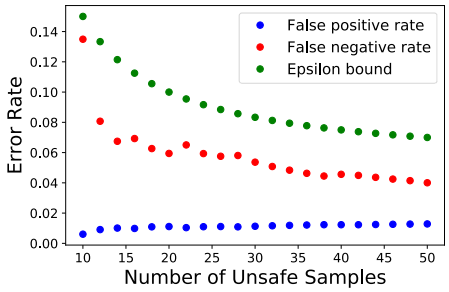

Fig. 4: Epsilon bound, false negative rate, and false positive rate on the nuScenes dataset while varying the number of unsafe samples. Consistent with our theory from Section 3.2, the sum of ϵ and the false positive rate is high when there are few samples.

We also demonstrate empirically on the Kaggle Lyft dataset that the variance on the false negative rate over different train/test splits is low. Table 1 displays

the variance on the false negative rate calculated over the 100 trials at each ϵ value. All of the variances are well below 0.003, suggesting that the test sequence false negative rates are clustered around ϵ (rather than having some sequences that fail on zero examples and others with catastrophic failures). As further evidence, in Figure 5, we provide a representative box plot of the false negative rates over the 100 trials with $\epsilon = 0.04$. The variances are indeed clustered around 0.04.

ϵ	0.02	0.04	0.06	0.08	0.10
Variance	0.00096	0.0019	0.0014	0.0023	0.0024

Table 1: Variance on the test sequence false negative rates at different ϵ.

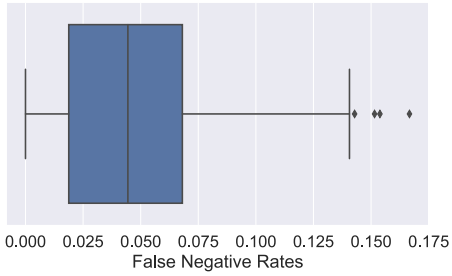

Fig. 5: Box plot of the 100 false negative rates calculated over randomized train/test splits with $\epsilon = 0.04$.

6.4 Additional Experimental Details: Robotic Grasping Experiments

Model and dataset details: The Grasp Quality Convolutional Neural Network (GQ-CNN) from [18] is a model that classifies whether a candidate robotic grasp will be successful. The inputs to a GQ-CNN are a point cloud representation of an object, $\mathbf{y}$, and a candidate grasp, $\mathbf{u}$. A GQ-CNN outputs the predicted probability, $Q_\theta(\mathbf{y}, \mathbf{u})$, that the candidate grasp will be able to successfully pick and transport the object. We use this predicted probability as the safety score, $g = Q_\theta(\mathbf{y}, \mathbf{u})$. We consider a candidate grasp "unsafe" if it will not be able to successfully pick the object (i.e. the true label is $Z = 0$). Note that this is exactly the ROC curve threshold tuning setup. We use the DexNet dataset of synthetic objects grasped with a parallel jaw gripper [18], which includes approximately 500k pick attempts not used in training the GQ-CNN model. These are divided into a 50%/50% train/test split. Each example is labeled a success if the robot successfully picks and places the object, and a failure otherwise.