Robust Aleatoric Modeling for Future Vehicle Localization

Max Hudnell True Price Jan-Michael Frahm
University of North Carolina at Chapel Hill
{mhudnell,jtprice,jmf}@cs.unc.edu

Abstract

The task of 2D object localization prediction, or the estimation of an object's future location and scale in an image, is a developing area of computer vision research. An accurate prediction of an object's future localization has the potential for drastically improving critical decision making systems. In particular, an autonomous driving system's collision prevention system could make better-informed decisions in the presence of accurate localization predictions for nearby objects (i.e. cars, pedestrians, and hazardous obstacles). Improving the accuracy of such localization systems is crucial to passenger / pedestrian safety. This paper presents a novel technique for determining future bounding boxes, representing the size and location of objects – and the predictive uncertainty of both aspects – in a transit setting. We present a simple feed-forward network for robust prediction as a solution of this task, which is able to generate object locality proposals by making use of an object's previous locality information. We evaluate our method against a number of related approaches and demonstrate its benefits for vehicle localization, and different from previous works, we propose to use distribution-based metrics to truly measure the predictive efficiency of the network-regressed uncertainty models.

1. Introduction

The task of 2D object localization has been an area of heavy research in recent years. Specifically, this task involves identifying the location and size of an object in an image, often represented as a 'bounding box'. Proposed methods, such as RCNN [14] and Fast RCNN [13], have achieved high accuracy in 2D object localization tests performed on tasks with a large amounts of training data available. These methods have been extended and improved upon by SSD [22] and YOLO [27], in order to improve efficiency and have become capable of object localization in real-time, while maintaining a high level of accuracy.

A natural extension of the task of 2D object localization is the *prediction* of 2D object localizations (or future object localization). This task is defined by the ability to produce an accurate estimate of an object's localization in a future timestep, given some amount of auxiliary information of the object (at minimum the previous localizations of that object). Given that we now have methods for obtaining previous localizations in real-time, it is now sensible to work towards producing a real-time method for object localization prediction which makes use of those previous localizations.

Accurate localization prediction is a potentially invaluable source of information especially for decision-making systems that rely on complex, real-time optimal control,

Figure 1. Visualization of past localizations of a tracked vehicle (orange), predicted +1 second future localization (red, dashed), and the true +1 second future localization (white), drawn over the +1 second future frame. The area of confidence is highlighted around the predicted box. Our method predicts the red dashed box as the mean $B(t)$ of a robust 4D probability distribution based on the Huber loss with scale parameter $\sigma(t)$, given only the sequence of orange bounding boxes as input. This model well captures the potential amount of divergence for different input scenarios. In both instances above, a car is observed traveling through an intersection, but the lateral view (bottom) has a larger spread of possible bounding boxes due to the car speeding up or slowing down.

such as collision prevention systems present in Advanced Driver Assistance Systems (ADAS) and Autonomous Vehicles (AV). Decisions made by these systems result in the alteration of the acceleration or steering direction of the vehicle, and are consequentially crucial to passenger and pedestrian safety. Accurately and efficiently extrapolating potential object *trajectories* thus decreases reactionary pressure from the control planning algorithms in these systems. A growing number of techniques [32] have investigated overhead or 3D future-state modeling for objects in the vicinity of an AV. However, such models inherently rely on tracked observations made from on-board sensors (*e.g.*, bounding box detections in forward-facing cameras [26]), and relatively few approaches have explored whether accurate future localization in the sensor space can help to improve planning or tracking algorithms, or indeed, whether such prediction is even feasible.

Motivated by this lack of analysis and toward the industry goal [28] of integrating machine-learning prediction algorithms in planning/tracking systems, we propose a data-driven approach for predicting future vehicle localizations, as well as uncertainty in this prediction, from ego-centric views on a moving car (Fig. 1). Uncertainty estimation – namely aleatoric uncertainty, which relates to inherent ambiguities in possible output states [16] – is a particularly important aspect to characterize in vehicle localization prediction, as the relatively short observation times and variability in driver behaviors can lead to highly divergent future states for similar initial scenarios. We thus seek a simple model that can model future localizations with as high of a confidence as possible, while also relating the spread of possible future states in the bounding box domain. Such information can be used, for example, to determine the region of the image in which the actual future bounding box is likely to occur with a certain level of confidence.

The key contributions of this paper are the following:

- We present a feed-forward neural network capable of accurately performing object localization prediction from a first-person vehicular perspective, solely using prior bounding boxes as input.
- We introduce a probabilistic formulation of the Huber loss that allows us to capture uncertainty in the future bounding box location while maintaining robust loss properties during training.
- We propose to evaluate the predictive distribution of a learned regressor, which better reflects the aleatoric accuracy of the predictor than existing metrics that directly evaluate the prediction mean against the ground-truth observation.
- We demonstrate that simple polynomial regression works at least as well as using a recurrent neural network (RNN)-based regressor to generate future localization predictions. In addition to being more efficient, this goes against trends in state of the art methods like [4] that expect an RNN to be more efficient in modeling uncertainty.

The rest of the paper is organized as follows: Section 2 reviews related work for both general and vehicle-specific localization prediction. Section 3 details our neural-network-based approach and describes the integration of uncertainty estimation into the Huber loss. Experiments and results are presented in Section 4, and we summarize our work in Section 5.

2. Related Work

Recent computer vision applications for future object localization have been partially inspired by predicting the motion of humans. One influential work is that of Alahi *et al.* [1], who trained a recurrent neural network (RNN) to generate probabilistic future trajectories for independent pedestrians, which is meant as a data-driven alternative to hand-tuned crowd navigation models. Among other non-vehiclar works, perhaps most relevant to our own is that of Yagi *et al.* [30], who construct a convolution-deconvolution architecture for the task of future person localization. They utilize three-input streams for encoding location-scale information, the motion of the camera wearer (ego-motion), and human pose information. Outputs of these input streams are concatenated and fed to a deconvolutional network to generate future 2D joint locations. However, this model does not consider aleatoric uncertainty.

Vehicle localization/motion prediction is a quickly growing area of research, with several contemporary approaches to our own and many other recent efforts. The majority of related works have focused on overhead modeling scenarios, where positions and trajectories are expressed in the coordinates of the 2D road surface [32]. Relatively few works have investigated 2D future vehicle localization for ego-motion video scenarios, although a number of vehicle tracking approaches use simple linear or quadratic regression to predict a rough localization of current bounding boxes given previous observations.

Among many recent and relevant overhead modeling scenarios [23, 26, 8, 7, 29], Altché and de La Fortelle [2] use a long short-term memory (LSTM) RNN to predict the overhead trajectory of a target vehicle given sequentially provided observations of the vehicle and its surrounding neighbors' positions and velocities. Kim *et al.* [17] predict the relative overhead positions of surrounding vehicles using a generic LSTM-RNN, with separate RNN instances being applied to each tracked vehicle. Independent RNNs are trained to predict positions at 0.5s, 1.0s, and 2.0s in the future. Li *et al.* [21] train a two-layer hidden Markov model to classify driving situations and then predict overhead trajectories for all vehicles in the scene using scenario-specific state behaviors. Driver actions are

simultaneously evolved according to their current scenario states and a learned Gaussian mixture model for state transitions. While we do not model state transitions, our model could serve to inform such a predictive formulation. Lee *et al.* [20] propose to use a conditional variational autoencoder to capture the possible future states of a given input scenario in an overhead representation. As part of their pipeline, they train an RNN to create samples from the non-parametric underlying distribution. Our approach contrasts theirs by directly regressing a distribution of localization transformations, rather than using a generative model that must be sampled from in order to predict possible paths. Their method also uses a Euclidean-distance-based loss objective, unlike our proposed robust loss objective, which is a confidence-weighted version of the Huber loss [15].

Detection-based 2D vehicle tracking pipelines typically also make use of simple predictive models that allow them to branch and bound correspondences in the current set of putative detections. For example, Choi [6] uses linear and quadratic models to prune bounding box hypotheses, and Duelholm *et al.* [9] use a simple linear model predict bounding boxes for objects that have lost tracking. We explore the potential of using these simpler continuous models instead of heavyweight, discrete RNN architectures when modeling the future state progression.

Possibly the most related approach to our work is the egocentric future bounding box localization method of Bhattacharyya *et al.* [4]. There, the authors use an RNN to jointly predict future vehicle odometry and future bounding boxes for pedestrians. They adopt a confidence-weighted Gaussian loss to model the future bounding boxes in a manner that captures both aleatoric and epistimological uncertainty. We demonstrate that this type of loss is not robust for future vehicle localization. Finally, we also note a number of contemporary works have recently appeared online covering similar topics in overhead and egocentric future localization [11, 10, 31, 24]. To our knowledge, however, no other works have explored improving the modeling of aleatoric uncertainty by adopting robust (particularly, non-Gaussian) underlying distributions.

3. Methods

The general goal of future object localization is to regress a model of not-yet-reached state(s) given some number of previous observations up to the current moment in time, t_0. In our case, previous observations consist of n bounding boxes for a single object (*i.e.*, a vehicle) obtained by a 2D object tracking system over the last ns seconds, where s is the frame-rate of the tracker. Given these bounding boxes $\{B_{-n+1}, B_{-n+2}, \ldots, B_0\}$ as input, we train a neural network to regress a function $B(t)$ that yields a predicted bounding box for the object at any future time $t > t_0$. Since uncertainty in object localization generally grows as a function of time, the network is also trained to output a second function, $\sigma(t)$, that models the uncertainty region for the localization $B(t)$; smaller $\sigma(t)$ values indicate higher confidence. Together, these regressed functions model the distribution of possible object states that may be observed at time t.

3.1. Data Representation and Architecture

Our network takes as input n prior bounding boxes $\{B_i\}$, with $B_i = [x_i, y_i, w_i, h_i]$ denoting the (x, y) center, width, and height of the box. The network architecture consists of a simple 4-layer feed-forward neural network. Each hidden layer is a fully connected layer consisting of 64 nodes, with ReLu [25] activations after each layer. The final layer is a linear regressor that outputs parameters $\left(\theta_{\mathbf{B}} = \left\{\theta_B^x, \theta_B^y, \theta_B^w, \theta_B^h\right\}, \theta_\sigma = \left\{\theta_\sigma^x, \theta_\sigma^y, \theta_\sigma^w, \theta_\sigma^h\right\}\right)$ for the bounding box predictor $B(t; \theta_{\mathbf{B}})$ and uncertainty model $\sigma(t; \theta_\sigma)$. In our implementation, each of the four bounding box dimensions are modeled by a separate prediction function that relies on its own set of parameters, *i.e.*,

$$B(t; \theta_{\mathbf{B}}) = \left[B_x(t; \theta_B^x), B_y(t; \theta_B^y), B_w(t; \theta_B^w), B_h(t; \theta_B^h)\right], \tag{1}$$

and similarly for $\sigma(t; \theta_\sigma)$:

$$\sigma(t; \theta_\sigma) = \left[\sigma_x(t; \theta_\sigma^x), \sigma_y(t; \theta_\sigma^y), \sigma_w(t; \theta_\sigma^w), \sigma_h(t; \theta_\sigma^h)\right]. \tag{2}$$

3.2. Relative Transformations as Output

Existing approaches for future object localization [31, 30] have sought to output a transformation of bounding boxes in the image space, *i.e.*, they return pixel coordinate offsets for the box center and a pixel change in width and height. Instead of regressing to the pixel displacement from the most recent box to the predicted box, we regress to a scale-invariant transformation [14]. This transformation consists of a width-space translation of the center coordinate, and a log-space translation of the width and height. Our main motivation for using such normalized transformations is that they allow us to assume a log-linear distribution of the width and height parameters (see Fig. 4).

A ground-truth scale-invariant transformation, $\hat{T}(t)$, from anchor box B_0 (the most recent known bounding box) to the ground-truth prediction box, $\hat{B}(t)$, is generated for training as follows:

$$\hat{T}(t) = [\hat{T}_x(t), \hat{T}_y(t), \hat{T}_w(t), \hat{T}_h(t)] \tag{3}$$

$$\hat{T}_x(t) = \frac{\hat{x}(t) - x_0}{w_0} \qquad \hat{T}_y(t) = \frac{\hat{y}(t) - y_0}{h_0}$$

$$\hat{T}_w(t) = \log\left(\frac{\hat{w}(t)}{w_0}\right) \qquad \hat{T}_h(t) = \log\left(\frac{\hat{h}(t)}{h_0}\right)$$

These serve as our target values during training. To generate a predicted box, $B(t)$, from an anchor box using the network-regressed transformation $T(t)$, we reverse the transformation, and apply it to anchor box B_0:

$$T(t) = [T_x(t), T_y(t), T_w(t), T_h(t)] \tag{4}$$

$$x(t) = w_0 T_x(t) + x_0 \qquad y(t) = h_0 T_y(t) + y_0$$
$$w(t) = w_0 \exp\big(T_w(t)\big) \qquad h(t) = h_0 \exp\big(T_h(t)\big)$$

Thus, our training fits $T(t)$ to $\hat{T}(t)$, rather than $B(t)$ to $\hat{B}(t)$, and the predictor as a function of network output can instead be understood as $B(t; \theta_\mathbf{B}) = B(B_0, T(t; \theta_\mathbf{B}))$.

3.3. Predictive Function Regression

In addition to performing absolute transformation regression, previous works in future object localization have modeled $B(t)$ in various forms at fixed timepoints in the future, including as the sequential application of a recurrent neural network (RNN) and a separate regression of the independent transformation at each future timepoint. For the task of 2D vehicle localization, however, it is potentially useful to allow predictions at arbitrary future timepoints, for example to provide higher-latency tracking algorithms with future predictions that temporally align to the frame of deployment. Moreover, an RNN must be sequentially applied to reach a desired discrete timestep. We also argue that, at least for our use-case, the RNN prediction approach is 'overkill', in that its ability to model highly variable motion patterns (*e.g.*, the path of a human navigating a crowd) is not necessary for the relatively smoother trajectories of automotive vehicles. We accordingly propose to model the motion trajectory as an ordinary polynomial, and we demonstrate that this approach fits the expected transformation distribution with at least as much efficiency as an RNN approach, while being simpler to compute and not requiring iterative application.

Recall that our network output consists of separate parameters (θ_B^d, θ_c^d) for each bounding box dimension $d \in \{x, y, w, h\}$. For our bounding box predictor, we choose to model $\theta_B^d = (\theta_B^{d(1)}, \theta_B^{d(2)}, \ldots, \theta_B^{d(p)})$ as the coefficients of a p^{th}-degree zero-intercept polynomial. Here, p is a hyperparameter of our algorithm. The associated transformation for dimension d is thus

$$T_d(t; \theta_B^d) = \sum_{i=1}^{p} \theta_B^{d(i)} t^i. \tag{5}$$

For our confidence regression, we expect the uncertainty in our future bounding box location to grow (perhaps slowly) as t increases. Thus, we model the uncertainty $\sigma_d(t; \theta_\sigma^d)$ of dimension d as

$$\sigma_d(t; \theta_\sigma^d) = |\theta_\sigma^{d(1)} t| + |\theta_\sigma^{d(0)}| + \epsilon, \tag{6}$$

where $\theta_\sigma^d = (\theta_\sigma^{d(0)}, \theta_\sigma^{d(1)})$, and ϵ is a small positive constant that helps avoid poor conditioning during training. In all our experiments, we use $\epsilon = 0.001$, and all our networks output $p + 2$ coefficients for each bounding box dimension, or $4p + 8$ total outputs.

3.4. Training Objective with Confidence

Our training objective minimizes a localization loss which measures error from the model's predicted transformation, $T(t)$, to the target transformation $\hat{T}(t)$ at multiple future timepoints $\{t_k\}$ with ground-truth localizations. We adopt the Huber loss [15] (sometimes called the smooth$_{L1}$ loss) due to its ability to robustly train against abnormal or outlier ground-truth bounding boxes [13]. This loss is applied separately over each bounding box dimension d.

Different from similar work in object localization prediction [20, 4, 31], we thus seek to *robustly* characterize in $\sigma_d(t)$ the potential confidence in our prediction. This can be directly integrated into the Huber loss. Consider the typical definition of the Huber loss between a predicted value x and target value $\hat{x}$:

$$H(\hat{x}, x) = \begin{cases} \frac{1}{2}(\hat{x} - x)^2 & \text{if } |\hat{x} - x| < \tau \\ \tau|\hat{x} - x| - \frac{1}{2}\tau^2 & \text{otherwise,} \end{cases} \tag{7}$$

where τ is a threshold at which the function switches from an L2-loss to an L1-loss. Taking a maximum-likelihood approach, we can interpret a solution to this loss as minimizing the negative log-likelihood of a modified, heavy-tailed version of the Gaussian distribution with mean $\mu = x$ and fixed scale parameter σ:

$$p(\hat{x}|x, \sigma) = \begin{cases} \frac{1}{c} \exp\left(-\frac{(\hat{x}-x)^2}{2\sigma^2}\right) & \text{if } |\hat{x} - x| < \tau \\ \frac{1}{c} \exp\left(-\frac{\tau}{\sigma^2}|\hat{x} - x| + \frac{\tau^2}{2\sigma^2}\right) & \text{otherwise,} \end{cases} \tag{8}$$

where $c = \sigma\sqrt{2\pi}\,\text{erf}(\frac{\tau}{\sigma\sqrt{2}}) + \frac{2\sigma^2}{\tau}\exp(-\frac{\tau^2}{2\sigma^2})$ is a normalizing constant that makes the area under the distribution curve equal to one, and erf$(\cdot)$ is the Gauss error function. As explained below, we use $\tau = 1.345\sigma$.

Here, we consider σ to be unknown *a priori* and thus regress it as $\sigma_d(t)$. Our prediction for transformation dimension d becomes our distribution mean, *i.e.*, $\mu = T_d(t)$. Taking the negative log-likelihood of Eq. (8), we arrive at the confidence-weighted Huber training objective for our bounding box regression:

$$\min_{\theta_B^d, \theta_\sigma^d} \sum_d H_d(\hat{T}_d(t), T_d(t; \theta_\sigma^d), \sigma_d(t; \theta_\sigma^d)), \tag{9}$$

$$H_d(\hat{T}, T, \sigma) = \log c + \begin{cases} \frac{(\hat{T}-T)^2}{2\sigma^2} & \text{if } |\hat{T} - T| < \tau \\ \frac{\tau}{\sigma^2}|\hat{T} - T| - \frac{\tau^2}{2\sigma^2} & \text{otherwise,} \end{cases}$$

where c is the normalizing constant defined above.

Value for τ. Ideally, the hyperparameter τ should scale with the certainty in the prediction. To provide some intuition for this property, consider the functional design of the Huber loss. The L1 tails of the loss provide a robust function for significantly abnormal/erroneous predictions; this dampens gradient steps toward outlier predictions during training. The L1 gradient around zero, however, is generally too large for the most accurate training cases, and thus to prevent overfitting and instability, the L2 loss is substituted when the error is small, since it has a comparably flatter gradient for errors in $[-1, 1]$. Thus, if the variance in the prediction-*vs*-ground-truth error is expected to be small (for instance, in our case, particularly when t is close to t_0), we should seek to have higher robustness to high-error ground-truth observations, since they are by definition outliers according to the variance model. On the other hand, larger uncertainty should lead to larger τ, since the spread of "relatively accurate" ground-truth observations is larger and we are therefore less confident that the associated gradient direction will lead to a general overall improvement.

In summary, τ is a scaled version of uncertainty computed as: $\tau_d(t) = M\sigma_d(t)$, where hyperparameter $M = 1.345$ was suggested by Huber to offer a good trade-off point for balancing the efficiency of the Gaussian with the robust L1 tails [19]. While the properties of the Huber scale parameter (σ) are generally well known [19], to our knowledge the trained regression of its value has not been explored for robust aleatoric modeling.

4. Experiments

We outline two main points of comparison for the task of future vehicle localization: (1) We evaluate whether our proposed confidence-weighted Huber loss (Eq. 9) is better suited for heteroscedastic modeling versus alternative probability-based loss functions. To this end, we also compute results for versions of our network trained with L1 and L2 confidence-weighted loss objectives, which respectively correspond to Laplace and Gaussian distribution models (*c.f.* Eq. (8)). We demonstrate that our robust aleatoric objective better learns the distribution space of possible future bounding box transformations. (2) We compare the results of our direct polynomial regression against the regression of an initial state for a co-trained RNN that predicts localizations, which has been proposed in similar approaches [4].

In addition to exploring the space of training configurations, we also propose alternative metrics to the displacement error and intersection-over-union statistics reported in existing works on future localization. Specifically, we analyze the accuracy of the predicted *distribution* of possible transformations compared to the ground-truth distribution of the testing data, instead of measuring whether the predicted future state evolved as predicted. This analysis gives a more holistic understanding of whether the variability of possible future states is truly understood by the network.

In all experiments, we evaluate our model using $p = 6$, which was experimentally chosen for p from 2 to 7 because it gave the lowest training loss across the L1, L2, and Huber models. Each network is trained using a batch size of 128 and a learning rate of $5e$-4 with Adam optimization [18].

4.1. Dataset and Implementation

To evaluate our methods, we train and test on samples generated from the KITTI "Raw" dataset [12]. This dataset consists of 38 videos with object tracklet information for various types of driving environments including: city, residential, and road settings. We consider the vehicle object tracklet labels 'Car', 'Van', and 'Truck' during evaluation.

We adapt the supplied tracking information for use with the vehicle localization prediction task. First, we isolate continuous two-second periods (20 frames) of tracking information for a given object; this makes up one sample. The first second of tracking information (defined by ten bounding boxes, split a tenth of a second apart) are established as *past* observations, and used as the input $\{B_{-n+1}, B_{-n+2}, \ldots, B_0\}$ for a given prediction task.

The object localizations associated with the last ten frames are reserved as the ground truth bounding boxes for the sample. We construct transformations from the anchor B_0 to each of the target bounding boxes $\{B_1, B_2, \ldots, B_{10}\}$ via the process detailed above. This set of ten transformations serve as the regression targets for the sample.

Our RNN implementation is a stand-alone network that takes as input the previous bounding box transformation and outputs a new transformation. This architecture is sequentially applied to output transformations at 0.1s intervals. The network consists of a 64-element gated recurrent unit (GRU) layer [5], followed by a 64-element hidden layer processing the GRU's hidden state, followed by a final linear layer. We modify our proposed neural network to output an initial hidden state for the RNN, instead of polynomial coefficients. The first future transformation is estimated directly from this initial hidden state, bypassing the RNN.

4.2. IoU and Displacement Error Analysis

We begin by reporting two widely used measurements for analyzing future bounding box locations: displacement error (DE) and intersection-over-union (IoU). Displacement error evaluates location prediction error and is calculated by taking the Euclidean distance between the centers of the predicted $B(t)$ and ground-truth $\hat{B}(t)$ bounding boxes. We report DE for +0.5s and +1.0s in the future, and we also report average displacement error (ADE) for future timepoints ranging from +0.1s to +1.0s at intervals of 0.1s. IoU evaluates location and scale error for our network's predictions and is computed as the overlap ratio for the predicted and ground-truth bounding boxes versus their joint area.

		All					Hard				
		DE		ADE	IoU		DE		ADE	IoU	
Loss	Func.	+0.5s	+1.0s		+0.5s	+1.0s	+0.5s	+1.0s		+0.5s	+1.0s
–	constant	32.06	72.15	36.98	0.498	0.339	44.23	102.40	51.62	0.326	0.128
–	linear	14.61	39.51	17.95	0.663	0.464	23.14	63.27	28.56	0.492	0.219
L1	$p = 6$	12.81	29.05	14.78	0.697	0.564	17.06	38.06	19.56	0.607	0.475
L2	$p = 6$	15.13	37.43	18.12	0.671	0.521	20.90	51.97	25.02	0.563	0.394
Huber	$p = 6$	12.58	29.18	14.72	0.708	0.584	16.18	36.76	18.76	0.622	0.488
Huber	RNN	14.01	31.37	16.12	0.686	0.570	19.62	44.63	22.71	0.577	0.442

Table 1. Displacement error and IoU scores for future prediction using different confidence-weighted objectives and functional regressions, averaged over all testing samples. The middle columns consider all testing samples, and the right columns consider only "hard" samples.

DE, ADE, and IoU results are shown in Table 1, along with two reference models, one of which predicts no bounding box motion ("constant"), and another which uses simple linear extrapolation of the transformation from $t = -0.1s$ to $t = 0s$ ("linear"). We report the average statistic over all test samples, as well as the average over only "hard" test cases (54% of the test samples). The latter ignores "easy" test cases, where the $t = +1.0s$ bounding box can be predicted with IoU greater than 0.5 using simple linear extrapolation. From the table, we observe that the transformation means predicted by the L2 loss are, on average, less accurate than those predicted by the L1 or Huber losses. However, we argue that these statistics only tell part of the story in terms of how the different predictions compare.

Distribution of IoU scores and inadequacy of exact evaluation metrics. DE and IoU are scores of future prediction: They measure how often the predicted bounding box distribution mean $B(t)$ happened to match well with the actual future bounding box. Neither measure takes into account the estimated uncertainty, $\sigma(t)$, and effectively, these metrics model the future as deterministic. This can readily be observed if we visualize the distribution of IoU scores, rather than simply assessing their mean. As shown in Fig. 2, the L1 loss achieved a higher rate of "exactly correct" predictions compared to the Huber loss, and the Huber loss had a slightly higher rate of "mostly right" (IoU around 0.75) predictions. We can conclude that the L1 loss was more effective at exactly regressing certain cases. However, this says very little about the underlying aleatoric encodings of the networks: *e.g.*, both losses exhibit a similar number of complete failure cases (IoU = 0) that may still fit within a comfortable confidence interval given $\sigma(t)$.

To glean a broader picture of network uncertainty, Bhattacharyya *et al.* [4] propose to evaluate the relationship between estimated uncertainty and the distance of the predicted mean from the ground-truth future observation. They note that the predicted uncertainty provides an upper bound on the error of the predictive mean and conclude that the model is thus useful in its prediction. While this is correct, the correlation of the uncertainty and error only demonstrates that the uncertainty model is behaving as intended:

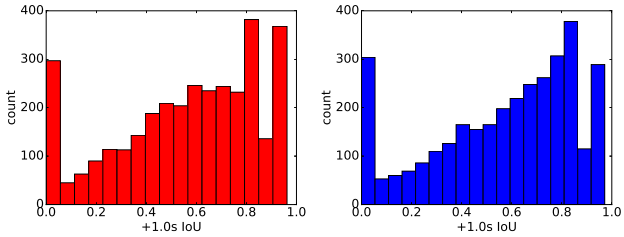


Figure 2. IoU distributions for our network trained with an L1 loss and $p = 6$ (left) and a Huber loss with $p = 6$ (right) at $t = +1.0s$. The peaks at IoU = 1 are partly due to stationary test instances.

The model is *accurate* in that it has a sense of when it might be wrong, but this observation says nothing about the *efficiency* of the model, that is, how correct it is about what the possible future states of a scenario may be. For instance, the uncertainty for the lower image in Fig. 1 might be larger than that for the top image, but if the uncertainty captures highly unlikely bounding box transformations that, say, allow the car to drastically change its size or vertical position in the image, then the underlying model is inefficient. We therefore argue that network models must be compared on the basis of how well their predictions match the probability space of actual future scenarios for a given input, rather than based on the *a posteriori* evaluation of their regressed mean against the known future occurrence.

4.3. Test Set Distribution Matching

When assessing the "understanding" of the future that our networks have learned, a proper statistic should relate how accurately and efficiently the distribution $(B(t), \sigma(t))$ describes the probabilistic space of all possible future states for a given scenario. If the underlying generative models were well understood, this could be assessed for each test case separately by repeatedly simulating future states for the given input and performing a statistical test on this sampling versus the regressed distribution. In lieu of a viable per-instance sampling approach, the next-best solution is to evaluate how well the space of regressed distributions matches the distribution of test samples. A regressor that is accurate will closely match the ground-truth values.

More specifically, we analyze the distribution of all 4D bounding box transformations $\hat{T}(+1.0s)$ in our test set. We bin the transformation space into voxels of size $\{0.1\}^4$ units in each dimension, corresponding to a 10% shift in x and y relative to the anchor box size and a 10% log-scale change in width and height. Ground-truth test-set transformations are then aggregated in this space using quadrilinear interpolation. The first images of Figs. 3 and 4 show the log-marginals of this aggregation for (x, y) and (w, h), respectively. More likely distributions appear brighter, and black regions correspond to voxels containing no transformations.

Next, we compare the aggregated predicted distribution of each analyzed network against this 4D test-set distribution. To obtain an aggregated predicted distribution for a given network, we first calculate the predicted distribution $p^{(k)}(\hat{T}^{(k)}|T^{(k)}(+1.0s), \sigma^{(k)}(+1.0s)) = \prod_d p_d^{(k)}(\hat{T}_d^{(k)}|T_d^{(k)}(+1.0s), \sigma_d^{(k)}(+1.0s))$ for each test instance k, where $p_d^{(k)}(\cdot)$ follows Eq. 8 in the case of the Huber loss, a Gaussian distribution for the L2 loss, and a Laplace distribution for the L1 loss. We then compute the probability of the transformation at each voxel center in the binned 4D space and normalize the integral of the space to sum to one. We calculate the average probability over all test cases to arrive at the final distribution for the 4D space. Figs. 3 and 4 show aggregations for different confidence-weighted loss functions, and also using an RNN.

In Table 2, we compare the predicted distribution $\mathcal{T}$ to the ground-truth distribution $\hat{\mathcal{T}}$ using the squared Hellinger distance $\mathcal{H}^2(\mathcal{T}, \hat{\mathcal{T}})$ [3], which summarizes the overall distance between distributions. The metric is computed as $\frac{1}{2}||\sqrt{\mathcal{T}} - \sqrt{\hat{\mathcal{T}}}||_2^2$, where $\sqrt{T}$ denotes the element-wise square root of the discrete probability volume; this metric summarizes the overall distance between distributions.

As can be seen in the table, the L2 loss still displays the worst performance – being a less robust metric, it evidently failed to adequately capture the edges of the distribution, which can be qualitatively observed in Figs. 3 and 4. The L1 metric is more robust, but it has higher error than our proposed confidence-weighted Huber loss. Fig. 3 also qualitatively suggests that the L1 loss yields slightly lower predictive confidences near the distribution tails. Interestingly, the RNN model, which has a much larger set of learned parameters and should theoretically be able to characterize a much wider set of future motions, does not outperform our polynomial models. On some level, this may be due to the nature of the dataset, which has a relatively short time horizon and captures objects with highly dynamic but smooth trajectories. We conclude that the complexity of RNNs is ultimately not necessary for modeling near-future vehicle localization, and due to their need to be iteratively applied to compute future states and their computational overhead, we advocate for the simpler polynomial regression proposed here.

Config.	L1	L2	Huber	Huber (RNN)
$\mathcal{H}^2(\mathcal{T}, \hat{\mathcal{T}})$	0.568	0.607	0.562	0.562

Table 2. Squared Hellinger distance between the ground-truth and predicted test-set transformation distributions, using different confidence-weighted objectives and functional regressions.

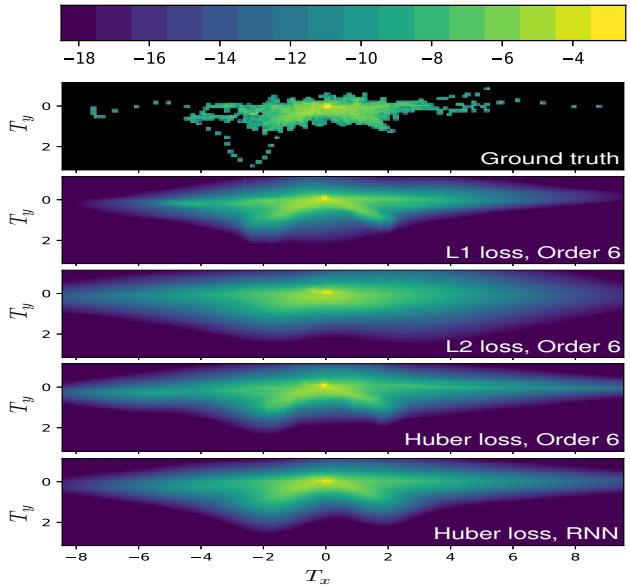
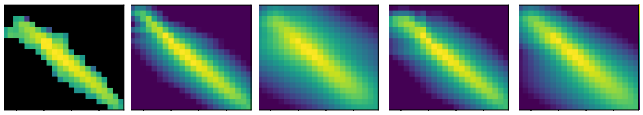


Figure 3. Marginal log-probability distributions for the space of $(T_x(+1.0s), T_y(+1.0s))$ transformations. Top: Ground-truth distribution. Other rows: Distributions for different confidence-weighted losses and predictive function parameterizations.



GT L1 ($p = 6$) L2 ($p = 6$) Huber ($p = 6$) Huber (RNN)

Figure 4. Marginal log-probability distributions for the space of $(T_w(+1.0s), T_h(+1.0s))$ transformations. T_w (x-axis) ranges from -0.9 to 1.4, and T_h (y-axis) ranges from -0.6 to 1.2.

5. Conclusion

In this paper, we introduced a robust neural network framework for predicting future vehicle localizations while accounting for inherent aleatoric uncertainty. We demonstrated that networks trained using a confidence-weighted Huber loss have better efficiency for modeling real-world future scenarios versus confidence-weighted L1 and L2 losses, and we argued for a distribution-based approach to compute this difference. Our results also showed that using RNNs to regress future vehicle states, which has been a recent trend, is at minimum no more performant than using a simpler polynomial model. In the future, our proposed approach could perhaps be extended to better model the inter-relationship between the dimensions of bounding box transformations, rather than considering each as having a sepa-

rate underlying probability distribution. We are also excited about the future integration of predictive machine-learning approaches like ours into planning and tracking systems, which is an open goal for the AVs [28] that could further the field towards full vehicle autonomy.

Acknowledgements This work was partially supported by NSF grant No. CNS-1405847.

References

[1] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese. Social LSTM: Human trajectory prediction in crowded spaces. In *CVPR*, pages 961–971, 2016. 2

[2] F. Altché and A. de La Fortelle. An lstm network for highway trajectory prediction. In *International Conference on Intelligent Transportation Systems (ITSC)*, pages 353–359. IEEE, 2017. 2

[3] R. Beran. Minimum hellinger distance estimates for parametric models. *Annals of Statistics*, 5(3):445–463, 1977. 7

[4] A. Bhattacharyya, M. Fritz, and B. Schiele. Long-term on-board prediction of people in traffic scenes under uncertainty. In *CVPR*, pages 4194–4202, 2018. 2, 3, 4, 5, 6

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014. 5

[6] W. Choi. Near-online multi-target tracking with aggregated local flow descriptor. In *ICCV*, pages 3029–3037, 2015. 3

[7] N. Deo, A. Rangesh, and M. M. Trivedi. How would surround vehicles move? a unified framework for maneuver classification and motion prediction. *Transactions on Intelligent Vehicles*, 3(2):129–140, 2018. 2

[8] N. Deo and M. M. Trivedi. Convolutional social pooling for vehicle trajectory prediction. *CVPR Workshops*, 2018. 2

[9] J. V. Dueholm, M. S. Kristoffersen, R. K. Satzoda, T. B. Moeslund, and M. M. Trivedi. Trajectories and maneuvers of surrounding vehicles with panoramic camera arrays. *Transactions on Intelligent Vehicles*, 1(2):203–214, 2016. 3

[10] D. Feng, L. Rosenbaum, and K. Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *ITSC*, pages 3266–3273. IEEE, 2018. 3

[11] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer. Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection. *arXiv preprint arXiv:1809.05590*, 2018. 3

[12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets Robotics: The KITTI Dataset. Technical report. 5

[13] R. Girshick. Fast R-CNN. *ICCV*, 2015 Inter:1440–1448, 2015. 1, 4

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 1, 3

[15] P. J. Huber. *Robust Statistics*. John Wiley & Sons, 1981. 3, 4

[16] A. Kendall and Y. Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NIPS*, pages 5574–5584, 2017. 2

[17] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi. Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network. In *Intelligent Transportation Systems (ITSC)*, pages 399–404. IEEE, 2017. 2

[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[19] S. Lambert-Lacroix, L. Zwald, et al. Robust regression through the hubers criterion and adaptive lasso penalty. *Electronic Journal of Statistics*, 5:1015–1053, 2011. 5

[20] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *CVPR*, pages 336–345, 2017. 3, 4

[21] J. Li, H. Ma, W. Zhan, and M. Tomizuka. Generic probabilistic interactive situation recognition and prediction: From virtual to real. In *ITSC*, 2018. 2

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1

[23] W. Luo, B. Yang, and R. Urtasun. Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In *CVPR*, pages 3569–3577, 2018. 2

[24] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha. TrafficPredict: Trajectory prediction for heterogeneous traffic-agents. In *AAAI*, 2019. 3

[25] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 3

[26] E. A. Pool, J. F. Kooij, and D. M. Gavrila. Using road topology to improve cyclist path prediction. In *Intelligent Vehicles Symposium (IV)*, pages 289–296. IEEE, 2017. 2

[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1

[28] W. Schwarting, J. Alonso-Mora, and D. Rus. Planning and decision-making for autonomous vehicles. *Annual Review of Control, Robotics, and Autonomous Systems*, 2018. 2, 8

[29] M. Suraj, H. Grimmett, L. Platinskỳ, and P. Ondrúška. Predicting trajectories of vehicles using large-scale motion priors. In *Intelligent Vehicles Symposium (IV)*, pages 1639–1644. IEEE, 2018. 2

[30] T. Yagi, K. Mangalam, R. Yonetani, and Y. Sato. Future person localization in first-person videos. In *CVPR*, 2018. 2, 3

[31] Y. Yao, M. Xu, C. Choi, D. J. Crandall, E. M. Atkins, and B. Dariush. Egocentric vision-based future vehicle localization for intelligent driving assistance systems. *arXiv preprint arXiv:1809.07408*, 2018. 3, 4

[32] W. Zhan, A. La de Fortelle, Y.-T. Chen, C.-Y. Chan, and M. Tomizuka. Probabilistic prediction from planning perspective: Problem formulation, representation simplification and evaluation metric. In *Intelligent Vehicles Symposium (IV)*, pages 1150–1156. IEEE, 2018. 2