

CARPAL: Confidence-Aware Intent Recognition for Parallel Autonomy

Xin Huang^{1,2}, Stephen G. McGill², Jonathan A. DeCastro²,
Luke Fletcher², John J. Leonard^{1,2}, Brian C. Williams¹, Guy Rosman²

Abstract—Predicting the behavior of road agents is a difficult and crucial task for both advanced driver assistance and autonomous driving systems. Traditional confidence measures for this important task often ignore the way predicted trajectories affect downstream decisions and their utilities.

In this paper we devise a novel neural network regressor to estimate the utility distribution given the predictions. Based on reasonable assumptions on the utility function, we establish a decision criterion that takes into account the role of prediction in decision making. We train our real-time regressor along with a human driver intent predictor and use it in shared autonomy scenarios where decisions depend on the prediction confidence. We test our system on a realistic urban driving dataset, present the advantage of the resulting system in terms of recall and fall-out performance compared to baseline methods, and demonstrate its effectiveness in intervention and warning use cases.

I. INTRODUCTION

In recent years, advanced driver assistance systems (ADAS) have played an important role in improving driving safety. One important aspect of these systems is the prediction of future driver intentions. Many motion prediction approaches have been proposed [1], [2], [3], [4], [5], [6] to generate accurate probabilistic motion predictions for vehicles, but leveraging of the prediction uncertainties in decision making towards safe driving systems with a low false alarm rate still remains an open challenge.

Solving this challenge could benefit from better insight on how prediction could affect downstream decision making systems. In [4], the authors propose an uncertainty-aware predictor that reasons about predictor errors in parallel to generating predictions. The predictor errors, however, are task-agnostic, in that they do not relate to the prediction effect on the downstream task. In this paper we extend this approach to task-specific confidence measures, and demonstrate this idea within parallel autonomy [7], an extension of shared control [8], [9], that monitors safety of a human-driven vehicle and intervenes if necessary to avoid near collisions. Our approach (CARPAL) not only predicts future driver trajectories but also estimates task-specific utilities and their uncertainties through a deep neural network regressor to decide whether it is necessary to take over human control. The regressor enables real-time decision making, by avoiding the need for computing utilities on the fly that can be computationally expensive.

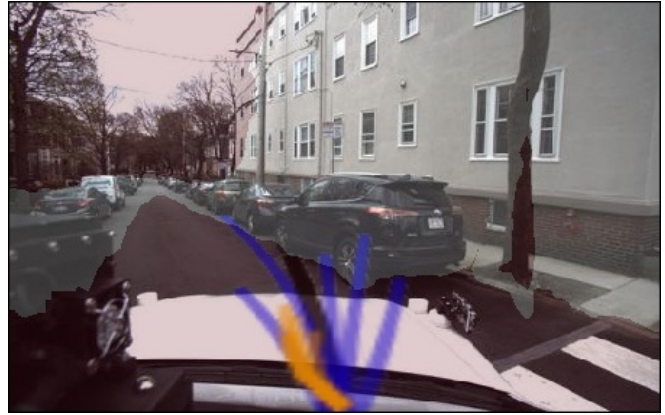


Fig. 1: A motivating scenario from an augmented risky driving dataset, where the future driver trajectory (in black) leads to a near collision with a parked car. While a task-agnostic predictor fails to intervene due to the high uncertainties in its predictions (in blue), our system, estimates task-specific utilities from the predictions and decides to intervene because of a better estimated utility from a backup planner, which generates safe backup trajectories (in orange) based on an obstacle map (shaded area) obtained from the front camera image and a driver intention map.

Inspired by [7], we define the trajectory utility based on two principles: safety and similarity to driver intention. There exist many ways to measure vehicle safety. In our work, we emphasize on keeping a minimum distance from obstacles, in order to avoid potential collisions with them. Intention similarity is also crucial to consider in a parallel autonomy system, as we want to follow driver intentions as close as possible, subject to safety. For intention similarity, we look at how close are the predicted trajectories to what the human driver could have done, in terms of the predicted distribution.

In this paper, we utilize a backup planner, which generates safe and physically feasible motion plans that follow driver intentions. Different from existing shared control literature [7], [10], [11], [12], [13], our work incorporates both uncertainty estimates that are relevant to utility of the driver predictions and the uncertainties from the rest of the parallel autonomy stack to improve decision making, especially when compared to task-agnostic confidence estimates.

A motivating example is illustrated in Figure 1, where our probabilistic motion predictor generates a set of prediction samples over the next three seconds in blue, and the acausal future trajectory taken by the driver is plotted in black, which indicates a near collision to a parked car on the right. Due to weak coupling between the prediction uncertainty

¹Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, USA
xhuang@csail.mit.edu

²Toyota Research Institute, Cambridge, MA 02139, USA

and the scene information, a task-agnostic predictor such as [4] may not intervene based on its confidence level in terms of prediction accuracies. On the other hand, our proposed method *CARPAL* is able to make better task-specific confidence estimates, such as the utilities of its predictions measured in this specific scene. Such estimates afford a better decision by taking over driver control using a safe backup plan (orange trajectory), which is generated by an auxiliary planner based on an obstacle cost map (shaded area) and a driver intention map.

Contributions This paper makes the following contributions. i) We propose a *confidence-aware predictor* that estimates trajectory predictions in addition to regressing *task-specific* utility expectation and uncertainty from its predictions. Our utility regressor is embedded within the predictor and provides estimates for *real-time decision making*. ii) We define a utility function and a planner that conforms to this utility function. We then propose a binary intervention decision function in parallel autonomy, which considers utility estimates to balance safety and following driver intentions. We provide an upper bound of the utility uncertainty of parallel autonomy in terms of the regressed utility estimates. iii) We compare to baseline methods in the context of parallel autonomy, and show that our predictor achieves better recall and fall-out. We demonstrate the effectiveness of our system in augmented realistic risky scenarios that would require interventions or warnings by ADAS.

II. RELATED WORK

Parallel autonomy is a vehicle shared-control framework [9], [7] that allows an intelligent driver assistant to monitor driver actions and intervene before an unsafe event could happen. Many existing methods on parallel autonomy assume a deterministic future driver trajectory or simple dynamics given current driver commands [7], [10], [11], [12], [13], which is insufficient in safety critical driving scenarios such as turning and merging. In [14], although multiple trajectories are predicted, only one trajectory is used to evaluate driver risk. Besides, all aforementioned approaches assume that the downstream backup intervention system has access to perfect environment information and thus always produces safe plans to replace risky human actions. However, due to perception noises from imperfect sensors and various weather and lighting conditions, there also exist uncertainties in the rest of the parallel autonomy stack such as the backup planner. In this work, we propose a robust parallel autonomy method that considers the uncertainties from both driver actions and perception noises leading to imperfect planned trajectories.

A handful of approaches [15], [16], [17] have been proposed to recognize driver intentions through driver’s physical state, Although these cues are good indicators of possible maneuvers that the driver is executing or planning to execute, they cannot be used to compute explicit risk measurements such as distance to obstacles or probability of near collisions.

On the other hand, vehicle motion prediction algorithms provide an efficient way to compute risk measurements

explicitly, e.g., by checking the distance from the predicted vehicle positions to obstacles. While many work focuses on estimating uncertainties from human drivers by generating probabilistic prediction results as Gaussian mixture model [2], [18], [19], or a distribution over a grid-based map [20], it is important to estimate the confidence of the predictors themselves, which is related to the model uncertainty. Along this line, [4] utilizes in-vehicle data from a front camera and CAN bus to produce probabilistic driver trajectory predictions associated with confidence scores estimating the errors of the predictions. Beyond prediction, [21], [22] estimate the confidence level in imitation learning by reconstructing its own input and detecting novelties based on reconstruction errors. In this work, we extend [4] and propose a multi-task model to estimate task-specific utility scores associated with predictions to facilitate real-time decision making in parallel autonomy. The transition from task-agnostic uncertainty in [4] to task-specific measures is inspired by ideas in value of information [23], and recent approaches for task-specific uncertainty approximations [24].

A challenge in parallel autonomy is to determine whether to intervene on behalf of the driver based on the confidence of driver risk evaluation. A low confidence indicates that the predictor may not be able to provide accurate predictions, which can lead to a false alarm if the driver is driving safely but gets intervened. In [25], [26], the authors propose a prediction-aware planning system that infers the confidence of predictions by maintaining a Bayesian belief over model variances, which is updated through tracking agent actions against a set of known goal states. In our work, we measure the confidence level of our prediction by computing the variances of utilities over sampled predicted trajectories, which do not require the goal states to be known a priori. In [27], the authors measure the uncertainties of an end-to-end control task by computing the variances of sampled control outputs, and decide whether to give control back to a backup planner by comparing against a dynamically optimized threshold as opposed to a fix task-agnostic threshold. Using a similar idea in parallel autonomy, we compare the driver utilities with the utilities of alternative driving options such as a backup planner to facilitate task-specific decision making. While there exist many planner options, we demonstrate our framework through an example backup planner that finds optimal paths by avoiding obstacles identified by the instance segmentation results extracted from front camera images, while following driver intentions subject to safety.

III. PROBLEM FORMULATION

Given the observed data from the driver vehicle, including past trajectories, CAN bus inputs, velocity and acceleration states, and front camera images, our goal is to estimate future trajectories and their utilities, as well as utilities of alternative options to facilitate decision making in parallel autonomy. The trajectories are represented by a probabilistic distribution $\tau \sim P(Y|X)$, where Y denotes future trajectories and X denotes observed data. Similar to [4], we assume that the

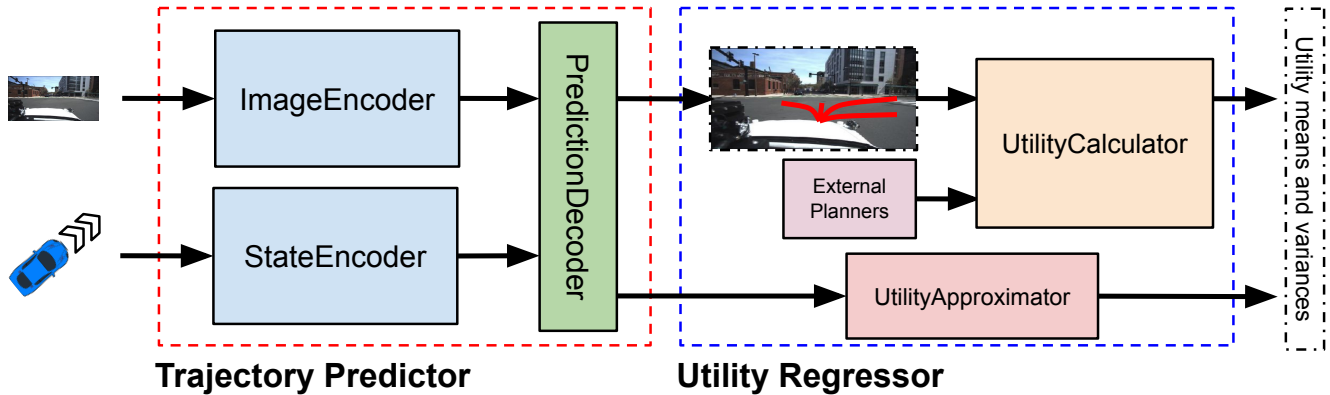


Fig. 2: Architecture diagram of our confidence-aware predictor. The deep neural network-based trajectory predictor generates probabilistic vehicle trajectories by taking camera images and vehicle states as input. A utility calculator provides supervisory utility cues by computing utilities for both predicted samples and planned trajectories from a backup planner. These utility cues are used as target values for a utility approximator, which estimates the means and variances of prediction and planned utilities to enable real-time decision making in test time.

trajectories can be projected onto a second-order polynomial basis, and represent them as a set of projection coefficients.

There exist many ways to define the utility function to cover different requirements for parallel autonomy. The utility in this work is defined based on the two principles introduced in Section I that capture safety and similarity to driver intentions. It is composed of two terms as follows:

$$u(\tau) = \text{Sigmoid}(d_{\mathcal{O}}^2(\tau)) + \alpha u_I(\tau) \quad (1)$$

$$d_{\mathcal{O}}^2(\tau) = \frac{1}{T} \sum_{t=1}^T \min_{o \in \mathcal{O}} \|\tau_t - o\|^2 \quad (2)$$

$$u_I(\tau) = \frac{1}{T} \sum_{t=1}^T \log P(\tau_t | I) \quad (3)$$

where the first safety term $d_{\mathcal{O}}^2$ represents the average minimum Euclidean distance to a set of obstacles \mathcal{O} , and the second intention similarity term u_I represents the likelihood of matching human intentions I , which are approximated by a spatial density map generated from sampled predicted human trajectories, and α is the weighting coefficient to scale the intention similarity term to balance between safety and following driver intentions. When computing the overall utility u , we apply a sigmoid function to the safety term, because the distance to obstacles matters more when it is small (e.g., in the linear region of the sigmoid function), which indicates a near collision.

Given the trajectory prediction distribution, we can sample a set of future trajectory prediction samples \mathcal{T}_I , and define the mean and variance of their utilities as follows:

$$\mu_I = \mathbb{E}_{\mathcal{T}_I} [u(\tau_I)] \quad (4)$$

$$\sigma_I^2 = \text{var}_{\mathcal{T}_I} [u(\tau_I)] \quad (5)$$

The mean value indicates the expected utility of the predicted samples, and the variance measures the uncertainties associated with the utilities. Similarly, we can define the mean and variance from a set of backup planner trajectories \mathcal{T}_P that are

designed to guarantee driver safety in risky situations given upstream perception information:

$$\mu_P = \mathbb{E}_{\mathcal{T}_P} [u(\tau_P)] \quad (6)$$

$$\sigma_P^2 = \text{var}_{\mathcal{T}_P} [u(\tau_P)] \quad (7)$$

Given a set of predicted samples and planned samples, we can define the decision function for parallel autonomy as a user-specified function F depending on their utility means and variances:

$$D(\mathcal{T}_I, \mathcal{T}_P) = F(\mu_I, \mu_P, \sigma_I^2, \sigma_P^2) \quad (8)$$

The decision function is a binary function that returns either a positive decision, which is to take over control using the backup autonomy trajectory from a backup planner, or a negative decision, which is to keep the human driver operating the vehicle.

The output of our system consists of the probabilistic trajectory predictions, as well as the estimated utility statistics for the predicted trajectories and the backup planned trajectories.

IV. METHOD

In this section, we describe our confidence-aware prediction system depicted in Figure 2, where we first introduce a trajectory predictor based on deep neural networks that outputs distribution over future vehicle positions given input data, and then show how we train a utility approximator model to regress the utility statistics by computing the target supervisory values from training data. Finally, we present how the regressed values can be used in a task-specific parallel autonomy system to improve driver safety. The training steps of the system are illustrated in Algorithm 1.

A. Variational Trajectory Predictor

The trajectory predictor consists of two parts. In the first part, we encode inputs from different sensors using a series of deep neural networks. For instance, the camera image

is processed with a pre-trained VGG network [28] with the last classification layer removed to extract road features and drivable areas. The vehicle states, including steering wheel angle, gas pedal signal, past trajectory, and linear and angular velocities, are fed into individual fully connected child networks to produce encoded states. The encoded states from each child encoder network are concatenated together into an embedding layer, which is fed into a trajectory decoder that consists of a series of fully connected networks to generate Gaussian parameters over trajectory coefficients. More details on the network architecture can be found in [4].

B. Utility Regressor

The utility regressor is divided into three parts, where we first compute acausal supervisory utility values by simulating a perception stack given the front camera image, second build a utility-based backup planner given the perception information, and third regress the acausal values through a deep neural network.

1) *Utility Computation*: We start by obtaining the acausal utility values defined in Eq. (1), which will be used as supervisory cues to train a regression deep neural network.

To extract the obstacle states, we simulate a perception stack by extracting semantic information from the front camera image, using a pre-trained semantic segmentation model, such as *DeepLabv3* [29]. The model provides pixel-wise labels on different objects in the scene, including roads, sky, and obstacle objects such as vehicles and road curbs. We transform the obstacle locations from image space to ground space to obtain an obstacle cost map using a homography transformation [30]. In parallel, we obtain a driver intention map by computing a Gaussian kernel density [31] spatial map extracted from a set of sampled predicted trajectories. Although the acausal future trajectory represents a more accurate driver intention, it ignores the multi-modality of human actions. As an alternative, our intention map represents the marginalized estimate of the intended trajectory trace based on the law of total probability.

After obtaining the obstacle locations and the driver intention spatial density map, we can compute the acausal utility for a given trajectory using Eq. (1).

2) *Backup Trajectory Planner*: In order to guarantee driver safety in cases where the predicted driver behavior is risky, it is necessary to plan some backup trajectories in parallel that can help avoid possible near collisions. Furthermore, in the context of parallel autonomy, the backup trajectories should follow driver intentions as close as possible when safety is guaranteed.

In this work, we propose a utility-based planner that maximizes our specified utility function by utilizing the obstacle cost map and intention density map obtained in Section IV-B.1. Additionally, to capture the inherent planner uncertainty due to the noise from upstream perception tasks, we add random obstacles or remove existing obstacles to and from the obstacle cost map, respectively, to simulate noises from perception and semantic segmentation systems. Besides, we shift the goal position, which can be obtained

Algorithm 1: Training procedure of CARPAL.

input : $X = (c: \text{front camera image}, s: \text{vehicle states}),$
 $\tau_a: \text{acausal future trajectory},$
 $\mathcal{M}: \text{current prediction model}$

output: \mathcal{M}' : updated prediction model

- 1 Obtain the predicted trajectory distribution and the estimated utility statistics using the current model \mathcal{M}

$$\hat{Y}, \hat{\mu}_I, \hat{\sigma}_I^2, \hat{\mu}_P, \hat{\sigma}_P^2 = \mathcal{M}(X)$$

- 2 Generate n trajectory samples $\mathcal{T}_I \sim \hat{Y}$
- 3 Compute a set of m obstacle cost maps \mathbf{M}_o given camera image c with m independent noises
- 4 Compute driver intention cost map $M_i = KDE(\{\tau_i\}_1^n)$ given predicted trajectory samples
- 5 Compute the final cost maps $\mathbf{M} = \mathbf{M}_o + \gamma M_i$, where γ is a scaling factor
- 6 Generate m backup planned trajectories \mathcal{T}_P using hybrid A* given \mathbf{M}
- 7 Compute utility statistics $\mu_I, \sigma_I^2, \mu_P, \sigma_P^2$ given $\mathcal{T}_I, \mathcal{T}_P$
- 8 Compute nll loss

$$\mathcal{L}_{nll} = -\log(P(\tau_a|\hat{Y}))$$

- 9 Compute regression losses

$$\begin{aligned} \mathcal{L}_{\mu_I} &= (\mu_I - \hat{\mu}_I)^2 & \mathcal{L}_{\sigma_I^2} &= (\sigma_I^2 - \hat{\sigma}_I^2)^2 \\ \mathcal{L}_{\mu_P} &= (\mu_P - \hat{\mu}_P)^2 & \mathcal{L}_{\sigma_P^2} &= (\sigma_P^2 - \hat{\sigma}_P^2)^2 \end{aligned}$$

- 10 Update model \mathcal{M} to \mathcal{M}' given losses $\mathcal{L}_{nll}, \mathcal{L}_{\mu_I}, \mathcal{L}_{\sigma_I^2}, \mathcal{L}_{\mu_P}, \mathcal{L}_{\sigma_P^2}$ using an optimizer
-

from training data, occasionally by a random noise vector as the planner does not always have the perfect driver's destination information.

As a result, we combine the obstacle cost map and the driver intention spatial map weighted by a scaling factor, to generate a set of physically feasible shortest path to the goal state using a hybrid A* search algorithm [32] given vehicle specs, such as the starting velocity and turning radius. We choose hybrid A* because of its simplicity, which can be replaced by any other planners the reason about cost (or utility) functions in a spatial map. The output includes a set of planned backup trajectories given noise samples in perception and goal locations.

3) *Utility Regressor*: In order to enable real-time decision making in test time, our prediction system embeds a utility approximator to regress the utility statistics. This is achieved by adding a few additional linear layers after the embedding layer to generate corresponding scalar values.

To obtain target values when computing the regression losses, we first sample n trajectory predictions, given the distribution over predicted future vehicle trajectories \hat{Y} , and compute their utilities means and variances. Additionally, we compute the utility statistics for m backup planner trajectories from Section IV-B.2.

C. Model Losses

We train our confidence-aware prediction model with five loss functions: a negative log-likelihood loss to measure the accuracy of predictions compared to the acausal future trajectory, and four L2 reconstruction losses between the regressed utility values and the target values to capture regression accuracy.

D. Confidence-Aware Decision Maker

While there exist many options for decision making depending on safety requirements and driver preferences, we propose a simple task-specific binary decision function F as detailed in Algorithm 2 to encourage safe driving while avoiding false alarms.

To guarantee driver safety, F would first compare the expected utility between the driver predictions (μ_I) and a backup planner (μ_P). In cases where μ_I is worse compared to μ_P (e.g., the predicted human trajectories are getting too close to obstacles), F needs to decide whether to intervene or not based on the confidence levels of the utility estimates. If both utility uncertainties are small such that they are below certain thresholds η_I and η_P for human predictions and backup planner trajectories, respectively, F would output a positive decision (1) to intervene because of a high confidence level. (For the sake of simplicity, we assume that $\eta_I = \eta_P$ and unify the notations as η .) Otherwise, F would warn instead to avoid false alarms. We use this extra logic to prevent interventions where there is a considerable chance of the driver utility outperforming the planner utility, and express people’s willingness to accept human errors but not automated system ones. We will show examples that merit such warning logic in Section V-E, yet it will not play a role in our quantitative analysis for our binary decision making system in Section V-D. We defer reasoning about a ternary decision function as future work.

In contrast, if the expected driver utility is acceptable, F would output a negative decision as 0 to not intervene.

Instead of comparing the expected driver utility μ_I to a fixed threshold to detect near collisions, we find it effective to compare it with the expected planner utility μ_P , thanks to the design of our utility function in Eq. (1) that includes a safety term and an intention similarity term. In cases where μ_P is higher than μ_I , the planner trajectories need to be farther away from the obstacles than the predicted trajectories to achieve a higher value in the safety term $d_O^2(\cdot)$, as the intention similarity term u_I would always favor the predicted driver trajectories. Furthermore, since the obstacle distance is transformed by a sigmoid function in the utility computation, the gap in $d_O^2(\cdot)$ would only make a difference if the distance from the driver trajectories to the obstacles is small enough. Otherwise, this distance would be in the saturation area of the sigmoid function, and thus lead to a similar utility compared to the utility of planner trajectories with a larger distance.

E. Bounding Utility Uncertainty in Parallel Autonomy

Here we show that under certain conditions, the utility uncertainty of the parallel autonomy system is bounded by

Algorithm 2: An example decision function for CARPAL, which depends on task specific utility estimates from the predictor and a backup planner.

```

input :  $\mu_I, \sigma_I^2, \mu_P, \sigma_P^2, \eta_I, \eta_P$ 
output:  $d$ : a binary decision bit, where 0 means a
           negative decision (e.g., not intervene), and 1
           means a positive decision (e.g., intervene)

1 if ( $\mu_I < \mu_P$ ) then
2   | if ( $\sigma_I^2 < \eta_I$  and  $\sigma_P^2 < \eta_P$ ) then
3   |   | return 1
4   | else
5   |   | warn
6   |   | return 0
7   | end
8 else
9   | return 0
10 end

```

the utility uncertainty estimated by our regressor. We denote the overall parallel autonomy utility entropy to be $h(U)$. For the intervention-only case, the overall utility U is a choice between the utility from driver action I and the utility from the planner action P .

Lemma 1: For the purpose of deciding on intervention, and assuming the utility regressors estimate utility uncertainty with a margin of Δ_u , the utility uncertainty of the parallel autonomy system is approximated by the regressor.

Proof: Using chain rule for differential entropy:

$$h(U) = h(U|I, P) + h(I, P) - h(I, P|U) \quad (9)$$

Since U can be completely determined by I and P , we have $h(U|I, P) = 0$. In addition, since U is a function of I and P , we have $h(I, P|U)$ to be non-negative. Therefore,

$$h(U) = h(I, P) - h(I, P|U) \leq h(I, P) \quad (10)$$

By properties of joint entropy, we get:

$$h(U) \leq h(I, P) \leq h(I) + h(P) \quad (11)$$

Plugging in the differential entropies for I and P based on Gaussian approximations:

$$h(I) = \ln(\sqrt{2\pi\sigma_I^2}) + \frac{1}{2} + \Delta_u \quad (12)$$

$$h(P) = \ln(\sqrt{2\pi\sigma_P^2}) + \frac{1}{2} + \Delta_u, \quad (13)$$

we obtain the final result. ■

V. RESULTS

In this section, we introduce the details of our method and two baseline methods, followed by a description of an augmented test set including risky scenarios that is used for verifying the intervention performance. We then demonstrate the advantage of our method over the baselines by measuring recall and fall-out rates in a binary classification setting, and show a number of intervention and warning use cases to highlight the effectiveness of our system.

A. Model Details

Our model utilizes a network similar to [4]. We add a utility approximator that consists of three linear layers with (64, 16, 4) neurons, where each linear layer is followed by a batch norm, ReLU, and dropout layers. We used PyTorch and trained our model on an AWS server with Tesla V100 GPUs. The prediction horizon is selected to be 3 seconds. The weighting coefficient α in Eq. (1) is selected to be 0.1, and the scaling factor γ for intention spatial map is selected to be 0.01. Both the number of predicted samples n and planning samples m are selected to be 10. The model is trained and validated on a naturalistic driving dataset collected in multiple urban trips under different weather and lighting conditions, which includes in-vehicle sensor data and images from a front facing camera.

B. Baselines

We introduce two baselines, where the first one assumes a deterministic constant velocity model that has been used in existing shared control literature [12] to estimate driver risk, and the second one is based on [4] that makes task-agnostic decisions according to the regressed accuracy of predicted vehicle trajectories.

1) *Velocity-Based Predictor*: The velocity-based predictor (VBP) assumes a constant velocity model to predict a deterministic future trajectory of the driver vehicle given the current velocity. To determine near collisions, VBP checks if the average minimum distance (see Eq. (2)) between its prediction and any obstacles identified in the front camera image is smaller than a minimum safe distance threshold d_S . The system intervenes if a near collision is detected, and does nothing otherwise. While we could have used other dynamics models such as constant acceleration [7], we observe that a constant velocity model leads to better prediction accuracy over longer horizons and choose it as a representative from the existing parallel autonomy literature.

2) *Accuracy-Based Predictor*: The accuracy-based predictor (ABP) generates decisions by estimating the accuracy of the predicted trajectories, as done in [4]. Here accuracy is defined by the displacement error between the predicted trajectories and the acausal driver trajectory. We compare the estimated prediction error with a threshold η_{ABP} , and if the predictor is accurate enough, we decide whether to intervene using the same criteria in VBP based on the predicted trajectories. ABP is task-agnostic as it ignores the detailed task information and uses only the accuracy to decide on intervention choices. Reasoning about the downstream task would be useful for improving safety performance, as we will show in both quantitative and qualitative results.

C. Augmented Test Set

Beyond training and validation, we use a set of test examples to verify the intervention performance of our system. As driving in this dataset was conducted by a safety driver, there do not exist risky scenarios, making it challenging to validate whether our system correctly intervenes or not. Therefore, we augment 10% of the test set using one of the two ways

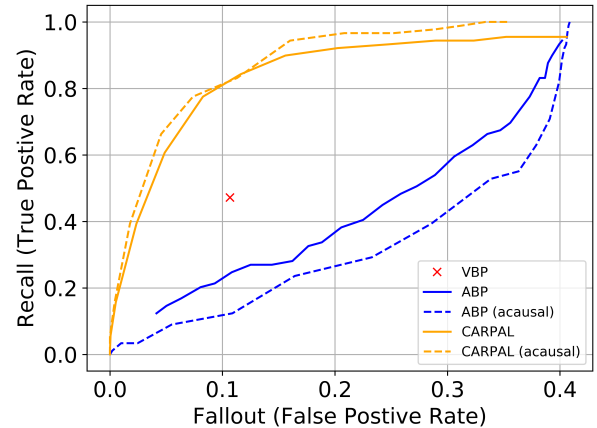


Fig. 3: ROC curves for our method (CARPAL) and two baselines, where our method achieves high recall while maintaining low fall-out over different threshold values, as compared to the baselines. Solid lines indicate results using regressed utilities, while dashed lines indicate results using acausal utilities.

to create potential “risky” scenarios. First, we scale up the past and future trajectory of the driver by 20%, causing the future trajectory to occasionally hit obstacles. This simulates cases where the driver is inattentive to the road and driving recklessly. Second, we add random obstacles to the front camera image, to simulate risky events when a pedestrian or a cyclist jumps in front of our car. We note these approaches work well because of the inputs of the predictor and planner we have chosen, but in general, creating realistic risky events remains a difficult problem.

D. Quantitative Results

When evaluating our binary intervention decision system in parallel autonomy, we define *positive events* (e.g., an intervention should happen) when the acausal driver trajectory reaches distance to the obstacles smaller than d_S , and the acausal planner utility is better than the acausal driver utility. While most existing literature [7], [11], [13] defines risky scenarios based on the near collisions, our work also considers the reliability of the backup planner, which can sometimes fail due to perception errors, before an intervention should happen. This allows us to ensure that the intervention is effective, as we expect the backup planner to improve the utility in risky scenarios.

We set the minimum safe distance d_S to be 1.6 meters, by considering the size of a normal size car with width of approximately 1.8 meters and length of approximately 4.6 meters, and show in Figure 3 two receiver operating characteristic (ROC) curves (in solid lines) by ranging possible threshold values, such as η for our method and η_{ABP} for ABP, and a fixed point for VBP since it does not depend on any thresholds.

In this work, we define the two metrics: recall and fall-out, in the following way, to measure the accuracy of our system and the false alarm rate, respectively. The first metric, recall (or sensitivity), measures the percentage of positive events

that are actually recognized as positive and intervened by the system. The second metric, fall-out, measures the percentage of negative cases that are intervened by the system. Our goal is to maximize recall while minimizing fall-out.

It is observed that both our method and ABP, shown in solid lines, trade off recall and fall-out by varying their threshold values. Compared to ABP, our method demonstrates better performance, because in many positive cases where an intervention is needed, ABP becomes conservative and fails to intervene due to low estimated accuracies in its predictions. On the other hand, our method is able to utilize the task-specific utility estimations, especially by recognizing that the planner utility is higher than the prediction utility, to correctly identify such positive cases. Additionally, we see that a deterministic predictor such as VBP would outperform ABP due to ABP's conservatism. However, VBP suffers from its prediction accuracy and has higher fallout compared to our method at the same recall level. This is because VBP, relying on the current velocity, does not have enough information to recognize that drivers may slow down when approaching obstacles in the near future, and thus creates more false alarms.

We also plot (in dashed lines) the ROC curves using the acausal utility values instead of the regressed ones, to verify the regression performance. We observe that the regressed curves are close to the acausal curves, and even though there exist inaccuracies in the regression, our method still outperforms the other two baselines.

E. Qualitative Results

We present a handful of examples to demonstrate three possible use cases of our predictor in the context of parallel autonomy. In the safe scenarios, we want to avoid overtaking the driver to maintain the usability of our system. In other scenarios where predicted trajectories have low utilities, we should gauge the estimated utility uncertainties to decide whether to intervene or warn, in order to maintain a high recall rate while avoiding false alarms.

1) *Safe Scenarios*: In most daily driving scenarios, the driver behaves safely and intervention is not needed. We show two common safe cases in the left column of Figure 4, where the drivers operate the vehicle safely on an open road or at an open intersection. Our system estimates high utilities on the driver predictions with low uncertainties, and thus decides to do nothing, regardless of the planner utilities.

2) *Intervention Scenarios*: In cases where the driver actions are risky, our system is able to recognize them and intervene after estimating a low driver utility and a high planner utility with high confidence. In addition to the motivating example in Figure 1, we illustrate two extra examples shown in the middle column of Figure 4, in which both driver actions, indicated by the acausal future trajectories in black, lead to a collision to an augmented obstacle. The obstacle in the top example simulates a situation where a pedestrian jumps suddenly from the gap between parked cars, and the obstacle in the bottom example simulates a vehicle or a cyclist on the opposite lane trying to pass the car blocking

their road. In both cases, our system decides to intervene after, with high estimated confidence levels, estimating a low utility for the predicted driver trajectories in blue and a high utility for the planner trajectories in yellow that can be used to take over driver control and avoid collisions.

The illustrated examples also demonstrate the advantage of our system over a task-agnostic system such as ABP, which decides whether to intervene based on the estimates over prediction accuracies and tends to be conservative by not intervening in these cases since the estimated accuracies are low. Instead of relying on a fixed threshold against prediction accuracies, our system approximates the utilities in the downstream backup planning task and finds them to be better than the average predicted driver utilities with high confidence levels, leading to better decisions.

3) *Warning Scenarios*: In addition to intervening the drivers when a clear near collision is detected, we demonstrate a few examples where an intervention is unneeded, but a warning can be helpful to remind the driver of potential risks, similar to what is done in modern cars such as vibrating the steering wheel and flashing a light on side mirrors. In the examples shown in the right column of Figure 4, our system detects that the driver utilities are lower than the planner utilities. However, due to a novel input with construction cones in the top example or the existence of moving pedestrians, the estimated driver utilities come with high uncertainties. Therefore, our system decides to warn the driver instead of taking over control. This would allow our system to successfully prevent a false alarm, as indicated by the safe acausal trajectory, while notifying the driver potential risks.

VI. CONCLUSIONS

In this paper, we propose a multi-task trajectory predictor that estimates in real time the utilities associated with its predictions for downstream decision making tasks, and show how our utility estimates benefit a parallel autonomy system. Compared to existing task-agnostic methods, our system achieves a higher true positive rate while maintaining a lower false alarm rate in an augmented test set considering common risky driving intervention use cases. Future work includes considering and comparing different utility definitions, decision functions, and risky events data.

REFERENCES

- [1] 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*, 2017, pp. 336–345.
- [2] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *CVPR Workshops*, 2018, pp. 1468–1476.
- [3] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *ICRA*. IEEE, 2019, pp. 2090–2096.
- [4] X. Huang, S. McGill, B. C. Williams, L. Fletcher, and G. Rosman, "Uncertainty-aware driver trajectory prediction at urban intersections," in *ICRA*, 2019, pp. 9718–9724.
- [5] X. Huang, S. McGill, J. A. DeCastro, B. C. Williams, L. Fletcher, J. J. Leonard, and G. Rosman, "Diversity-aware vehicle motion prediction via latent semantic sampling," *arXiv:1911.12736*, 2019.

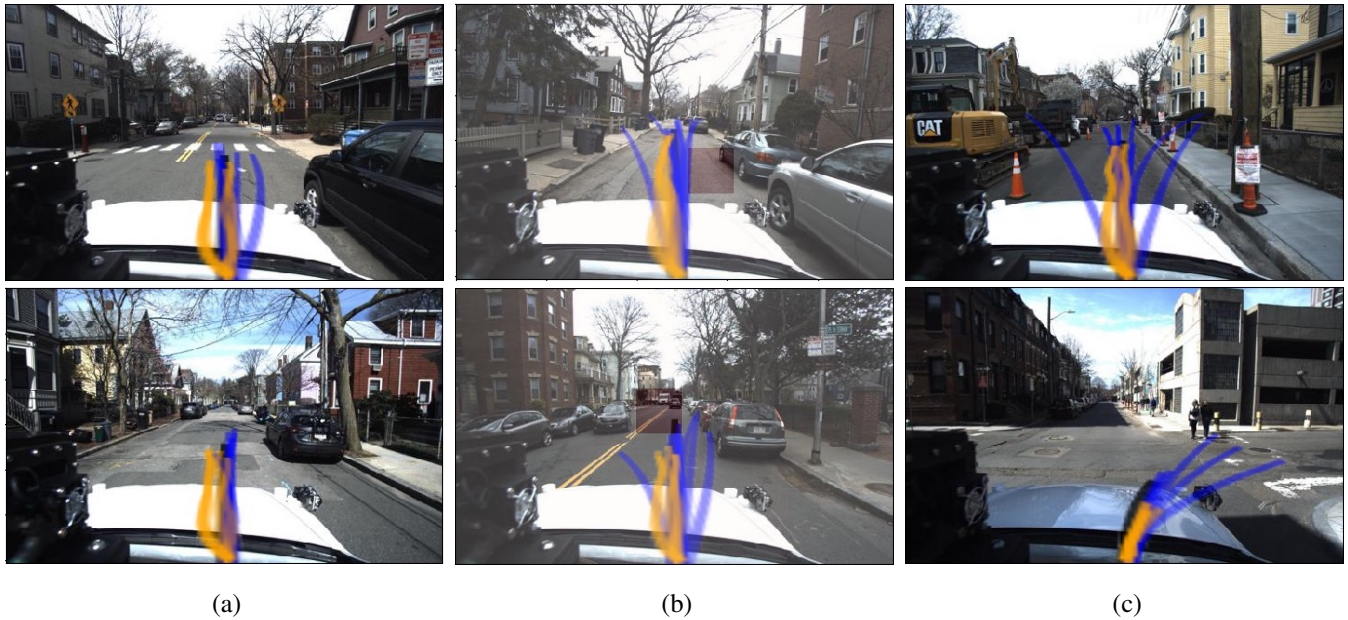


Fig. 4: Use cases of our predictor in various driving scenarios. Blue and orange indicate predicted and planner samples, respectively. Black indicates the actual trajectory. Shaded squares in the middle column indicate augmented obstacles. (a) Safe scenarios, where estimated utilities are high with small uncertainties. Our system does not intervene. (b) Intervention scenarios, where the actual driver trajectories are risky. Our system intervenes after estimating the driver utilities to be low and the planned utilities to be high, both with high confidence levels. In contrast, ABP acts conservatively without intervention because prediction uncertainties are large. (c) Warning scenarios, where estimated utilities are low with high uncertainties, even though the planner has better utilities. Our system warns instead of intervention to avoid false alarms.

- [6] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv:1910.05449*, 2019.
- [7] W. Schwarting, J. Alonso-Mora, L. Pauli, S. Karaman, and D. Rus, "Parallel autonomy in automated vehicles: Safe motion generation with minimal intervention," in *ICRA*. IEEE, 2017, pp. 1928–1935.
- [8] D. A. Abbink, M. Mulder, and E. R. Boer, "Haptic shared control: smoothly shifting control authority?" *Cognition, Technology & Work*, vol. 14, no. 1, pp. 19–28, 2012.
- [9] L. Saleh, P. Chevrel, F. Claveau, J.-F. Lafay, and F. Mars, "Shared steering control between a driver and an automation: Stability in the presence of driver behavior uncertainty," *T-ITS*, vol. 14, no. 2, pp. 974–983, 2013.
- [10] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," *arXiv:1802.01744*, 2018.
- [11] S. J. Anderson, S. B. Karumanchi, and K. Iagnemma, "Constraint-based planning and control for safe, semi-autonomous operation of vehicles," in *IV*. IEEE, 2012, pp. 383–388.
- [12] J. Alonso-Mora, P. Gohl, S. Watson, R. Siegwart, and P. Beardsley, "Shared control of autonomous vehicles based on velocity space optimization," in *ICRA*. IEEE, 2014, pp. 1639–1645.
- [13] S. M. Erlien, "Shared vehicle control using safe driving envelopes for obstacle avoidance and stability," Ph.D. dissertation, Stanford University, 2015.
- [14] V. A. Shia, Y. Gao, R. Vasudevan, K. D. Campbell, T. Lin, F. Borrelli, and R. Bajcsy, "Semiautonomous vehicular control using driver modeling," *T-ITS*, vol. 15, no. 6, pp. 2696–2709, 2014.
- [15] E. Murphy-Chutorian, A. Doshi, and M. M. Trivedi, "Head pose estimation for driver assistance systems: A robust algorithm and experimental evaluation," in *ITSC*. IEEE, 2007, pp. 709–714.
- [16] B. Morris, A. Doshi, and M. Trivedi, "Lane change intent prediction for driver assistance: On-road design and evaluation," in *IV*. IEEE, 2011, pp. 895–901.
- [17] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *T-IV*, vol. 1, no. 1, pp. 90–104, 2016.
- [18] J. Wiest, M. Höffken, U. Kreßel, and K. Dietmayer, "Probabilistic trajectory prediction with Gaussian mixture models," in *IVS*. IEEE, 2012, pp. 141–146.
- [19] B. Ivanovic and M. Pavone, "The Trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *ICCV*, Seoul, South Korea, 10 2019.
- [20] 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 *ITSC*. IEEE, 2017, pp. 399–404.
- [21] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *RSS*, 2017.
- [22] A. Amini, W. Schwarting, G. Rosman, B. Araki, S. Karaman, and D. Rus, "Variational autoencoder for end-to-end control of autonomous driving with novelty detection and training de-biasing," in *IROS*. IEEE, 2018, pp. 568–575.
- [23] R. A. Howard, "Information value theory," *IEEE Transactions on systems science and cybernetics*, vol. 2, no. 1, pp. 22–26, 1966.
- [24] G. Rosman, C. Choi, M. Dogar, J. W. F. III, and D. Rus, "Task-specific sensor planning for robotic assembly tasks," in *ICRA*. IEEE, 2018, pp. 2932–2939.
- [25] J. F. Fisac, A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, S. Wang, C. J. Tomlin, and A. D. Dragan, "Probabilistically safe robot planning with confidence-based human predictions," *arXiv preprint arXiv:1806.00109*, 2018.
- [26] D. Fridovich-Keil, A. Bajcsy, J. F. Fisac, S. L. Herbert, S. Wang, A. D. Dragan, and C. J. Tomlin, "Confidence-aware motion prediction for real-time collision avoidance1," *IJRR*, 2019.
- [27] K. Lee, K. Saigol, and E. A. Theodorou, "Safe end-to-end imitation learning for model predictive control," *arXiv:1803.10231*, 2018.
- [28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv:1409.1556*, 2014.
- [29] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.
- [30] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [31] E. Parzen, "On estimation of a probability density function and mode," *The annals of math. stat.*, vol. 33, no. 3, pp. 1065–1076, 1962.
- [32] J. Peterleit, T. Emter, C. W. Frey, T. Kopfstedt, and A. Beutel, "Application of hybrid A* to an autonomous mobile robot for path planning in unstructured outdoor environments," in *ROBOTIK 2012: 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.