This is the published version of a paper presented at 2019 IEEE Intelligent Vehicles Symposium (IV).

Citation for the original published paper:

Mänttäri, J., Folkesson, J. (2019)
Incorporating Uncertainty in Predicting Vehicle Maneuvers at Intersections With Complex Interactions
In: 2019 IEEE Intelligent Vehicles Symposium (IV) IEEE
https://doi.org/10.1109/IVS.2019.8814159

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-257881
Abstract—Highly automated driving systems are required to make robust decisions in many complex driving environments, such as urban intersections with high traffic. In order to make as informed and safe decisions as possible, it is necessary for the system to be able to predict the future maneuvers and positions of other traffic agents, as well as to provide information about the uncertainty in the prediction to the decision making module. While Bayesian approaches are a natural way of modeling uncertainty, recently deep learning-based methods have emerged to address this need as well. However, balancing the computational and system complexity, while also taking into account agent interactions and uncertainties, remains a difficult task. The work presented in this paper proposes a method of producing predictions of other traffic agents’ trajectories in intersections with a singular Deep Learning module, while incorporating uncertainty and the interactions between traffic participants. The accuracy of the generated predictions is tested on a simulated intersection with a high level of interaction between agents, and different methods of incorporating uncertainty are compared. Preliminary results show that the CVAE-based method produces qualitatively and quantitatively better measurements of uncertainty and manage to more accurately assign probability to the future occupied space of traffic agents.

I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) and the development of autonomous driving have shown great promise in improving the safety of traffic participants, both by minimizing human error and by relying on intelligent systems to handle dangerous driving situations. In order for these systems to be pro-active and make informed decisions, it is necessary for their planners to be able to recognize and predict the future intentions and maneuvers of other agents in traffic [1]. However, prediction systems cannot be constantly correct, and many seemingly similar situations can evolve into highly different scenarios due to the stochastic behavior of human drivers or a road structure that allows several possible maneuvers. For a planner it is then of great importance to know all possible maneuvers and calculate a safe trajectory based on their likelihood. To produce these probabilities, many methods adopt a Bayesian approach such as [2], [3], [4]. These methods are often computationally expensive, especially if high levels of interactions are to be taken into account. On the other hand, Deep Learning approaches can often excel at learning arbitrary interactions, and have several methods of incorporating uncertainty. One such method is learning to replicate the behavior of a probabilistic approach [5], [6], but are limited by the strength of the underlying probabilistic model. Another common approach is to learn a distribution as a latent representation which can then be sampled for possible future trajectories, such as [7]. However, depending on the representation used in such systems, it can be difficult to keep the complexity low when incorporating interactions and road geometry as well.

The method proposed in this paper utilizes a single Deep Learning-based module that is capable of learning complex interactions between agents, by leveraging a generic image representation of the traffic scenario and capturing spatio-temporal characteristics through recurrent convolutional operations, previously shown to be effective in predicting lane change maneuvers in highway scenarios [8]. In order to incorporate uncertainty two approaches are evaluated, Monte Carlo Dropout and using a Conditional Variational Auto Encoder. This allows the system to capture the stochastic behavior of traffic participants as well as the ambiguity in the data without scaling poorly to the amount of agents considered, due to the input sequence length and computations remaining constant for scene prediction and samples being generated in parallel during run-time using mini batches.

A. Related work

When aiming to incorporate uncertainty in a prediction system, a natural direction to start with is to use a Bayesian approach. One such method, by Streubel et al., [2] trains three different Hidden Markov Models (HMMs), corresponding to the three possible maneuvers at an intersection on an input sequence of velocity, acceleration and yaw rate of a vehicle approaching the intersection. For prediction, each model is fed the input sequence in order to examine which model fits the data the best, and the chosen model then corresponds to the predicted maneuver. While this method is intuitive and provides probabilities for different maneuvers, its high accuracy is mostly obtainable in less complex scenarios as it focuses on one vehicle at a time, disregarding the possible interactions between them. An approach by Hubbman et al. [3] which aims to model interactions as well is instead based on Partially Observable Markov

1Robotics, Perception and Learning Lab, KTH, Stockholm, Sweden (e-mail: manttari@kth.se, johnf@kth.se)
Decision Processes (POMDPs). Each vehicle is modeled as 
a POMDP with their intended route being a hidden variable. 
Although the hidden variable is a joint representation of 
the intended route of all vehicles, the value for this is 
inferred using a naïve Bayesian classifier which tends to 
easily over-fit to training data. Apart from this, the effects 
of interactions between vehicles on the predicted future 
positions are modeled with an offset on acceleration based 
on Time-To-Collision (TTC) and is not jointly taken into 
account with the possible intended maneuvers. A Bayesian 
approach that fully encompasses interactions is that of Kuhnt 
et al. [4]. The authors used the Object-Oriented Probabilistic 
Relational Modeling Language (OOPRML) to construct a 
directed probabilistic graphical model (PGM) over the traffic 
scenario. The PGM models the state (position, velocity, 
acceleration) of each vehicle, that can be present on any route 
with a certain behavior, such as following behind a vehicle 
or merging in front of one. These behaviors are dependant on 
interactions with other vehicles, which are defined as their 
routes for example crossing, merging, or diverging. Using 
the available information for the value of graph nodes, they 
can then infer the values of intended routes or behaviors 
of vehicles. This method takes interactions into account 
extensively, and shows clear results on how the interactions 
increase prediction performance. However, similarly to [3], 
these Bayesian approaches are computationally expensive, 
a cost that also increases with the number of vehicles and 
routes present. It is worth noting that Kuhnt et al. mitigated 
this by implementing a heuristic prior to keep the cost from 
growing exponentially.

Deep learning methods have increased in popularity not 
only in autonomous driving areas, but have also been ex-
tended to fulfill applications where uncertainty is required by 
either learning a distribution as a representation or learning to 
replicate the behavior of a probabilistic method. An example 
of the latter is the Dynamic Occupancy Grid (DynOG) based 
method by Hoermann et al. [6]. The system first transforms 
all of the registered vehicle and pedestrian tracks from the 
sensors into an occupancy grid, which is then fed to a particle 
filter to produce dynamic occupancy grids which incorporate 
velocity information as well as the Dempster-Shafer masses 
of the cell being occupied or free. They then train a CNN 
on the multi-channel input to predict the resulting DynOG 
different time-steps into the future. The great strength of 
this method is, that like our proposed method, it learns 
agent interactions directly from the data without a need 
for explicitly modeling them, and can consider an arbitrary 
amount of traffic participants. These predictions are however 
generated using only one frame as input to the CNN instead 
of a sequence, suggesting that the explicitly included velocity 
is enough to capture the spatio-temporal aspect of the data, 
which is not always the case when either explicit or implicit 
modeling of acceleration could give key information in a 
scene. Furthermore, although the computational time does 
not necessarily increase with the number of agents in the 
scene, there is always an overhead included in the pre-
processing required by executing the particle filter.

An additional example of a deep-learning approach which 
learns to replicate probabilistic behavior is that by Nadarajan 
et al. [5]. In this work the authors first produce both an 
Augmented Occupancy Grid (AOG) and a Predicted Oc-
cupancy Grid (POG). The AOG includes the occupancy, 
position, velocity, heading and acceleration of any vehicle 
occupying a given cell. The POG gives a probabilistic 
rendition of the future cell occupancy by assigning proba-
bilities to different maneuvers with a rule-based method and 
deviations in these maneuvers with a triangular probability 
density function based on the input features. They then train 
Stacked Denoising Autoencoders (SDAs) to learn a lower-
level representation of the AOGs which are then used as 
input to Random Forest (RF) classifiers. Each RF is given 
the entire AOG as input and is tasked with learning to predict a 
single cell in the corresponding POG, meaning for a grid size 
of 80 by 80 there are 6400 RF classifiers that must be trained. 
Although this method does take interactions into account, the 
probability incorporated in it is still rule-based using expert 
knowledge, meaning there is a strong bias on the human 
prior set in the system. Furthermore, even though the authors 
decreased computational time by having the RFs operate on a 
lower dimensional representation, the computational load of 
training 6400 RF classifiers on a large dataset is substantial.

One of the arguably most comprehensive Deep Learning 
based approaches is that of the DESIRE [7] framework 
presented by Lee et al. The framework consists of a multi-
module system where different modules are tasked with 
incorporating uncertainty, interactions, and scene infrastruc-
ture. Initially, previous and future trajectories are sent as 
a sequence of positions into a CVAE which is sampled 
to produce likely predictions. These predictions are then 
combined with an environment feature representation ac-
quired by a CNN, as well as the features of the surrounding 
vehicles according to a polar grid around the vehicle of 
interest. This combination is then used as input to a further 
decoder in a refinement module, which according to a cost 
function learns an optimal displacement to iteratively shift 
the trajectories from the CVAE in order to incorporate this 
information and produce more robust results. The result is 
the top K most likely prediction candidates for the future 
trajectories. Although the feature pooling in the polar grid 
may perhaps not distinguish multiple vehicles following each 
other, the system does take into account relevant interactions, 
with any amount of vehicles present. The CVAE module is 
also capable of learning the stochastic behavior of traffic 
agents and the CNN feature extraction allows for the scene 
structure to be taken into account. The largest drawback 
of this system is the cost for operating on absolute positions 
while incorporating all of this information. Although there 
is no loss in precision for the predictions, the multi-module 
architecture is computationally demanding and requires not 
only multiple samples from the CVAE, but also several 
iterations of the refinement module to correctly take all of 
the available information into account. The method also does 
not produce a probability of future maneuvers but rather a 
ranked list of the top K most likely future trajectories.
In the proposed method, the goal is to produce a prediction system that incorporates uncertainty and takes all available interactions between an arbitrary number of traffic agents into account, while keeping the complexity low. To allow the system to learn the effect of arbitrary interactions on vehicles, the generic scene representation previously proposed in [8] is adopted, explained in detail in section II. This enables the method to learn interactions directly from data without the need to specify explicit models for vehicle dynamics or interactions. To incorporate uncertainty in the predictions, two extensions of the prediction network are compared. One method augments the previous system to use Monte Carlo Dropout (MC Dropout) [9] while the other aims to learn a distribution of possible future trajectories as a latent representation using an Conditional Variational Auto Encoder (CVAE). This way, we can approximate distributions for predicted maneuvers by sampling our method without the need for computationally expensive purely Bayesian approaches. As deep learning models require large datasets and we want to focus on intersections with a high level of interactions and stochastic behavior, a simulator was constructed based on an extended Intelligent Driver Model (IDM) [10] for data acquisition. Our main contributions are:

- Extension of the previously defined prediction framework in [8] to incorporate uncertainty.
- Comparison of Monte Carlo Dropout and Conditional Variational Auto Encoders as means of learning stochastic behaviour in traffic agents.

The remainder of the paper is structured as follows: Section II presents an overview of the prediction system used, as well as how vehicle tracking data is converted to be used in the network. Experiments to evaluate the proposed method are described in Section III, with their results shown in Section IV. Finally, Section V presents the conclusions and future work.

II. MANEUVER PREDICTION WITH DEEP NEURAL NETWORK

The proposed method utilizes a deep learning approach to predict the trajectories of vehicles in a traffic scenario by converting vehicle tracking data to a generic image representation. The following subsection explains the representation used and how to convert vehicle track data to it, while subsection B explains how the generic representation is used to generate predictions.

A. Generic Visual Representation

We begin the conversion process by accumulating all the vehicle tracks at a rate of 5 Hz in a 40m x 40m square area with the intersection origin as the center, which operates as the sensor range of the system. These vehicles are then rendered according to their assumed size and positions relative to the intersection onto a 128x128 single-channel image representing the sensor range, as seen in fig. 4. While it is possible to increase the size of this image to increase the precision of the predictions, this resolution was chosen as it provides a satisfactory trade-off between computation time and the prediction accuracy. If the prediction of a single specific agent is of interest, additional images depicting only that agent are also generated in order to simplify the analysis later. The key advantages of representing the traffic scenario in this way are the following:

- There is no need to restrict the vehicle interaction types taken into consideration.
- The number of traffic agents considered does not affect the computational complexity.
- Arbitrary traffic agent classes can be specifically modeled by extending the representation.

Prediction models that only consider specific vehicle relationships often need strict specifications for the maximum number of agents considered, how to segment the space around a vehicle, or how to pool the features of agents residing in these segments. A benefit of employing recurrent and convolutional operations on an image as per the proposed method, is that visual representations offer by default relevant information such as possible configurations or the amount of vehicles to examine. Also, added traffic participants do not have a negative impact on the method’s scaling since the complexity of the representation that serves as the network’s input is not affected by the number of traffic agents. It should be noted that, although the current representation operates only on passenger vehicles, cyclists, pedestrians or other traffic agents could be integrated as distinct shapes and colors, namely icons, thus extending the method’s learned interactions. Representing traffic agents with icons, as opposed to occupancy grids, allows us to extend the method’s applicability by simply denoting different agent classes with different icons and additionally, eliminates the problem of reliably determining which cell an agent occupies at every given time-step. Although this representation retains enough information to predict maneuvers, certain aspects should be acknowledged:

- The accuracy of the generated paths is dependent on the resolution used in the images.
- Explicit modeling of the dynamics of the vehicles is lost.
- Performance is dependent on training data.

By converting the vehicle tracks to a discrete pixel representation, the proposed method is unable to produce predicted trajectories as continuous variables. This discretized representation is sufficient for predicting maneuvers and providing coarse future trajectories, but it could prove insufficient if higher precision predictions are required. Furthermore, even though spatio-temporal information is retained in the image sequences, there is no explicit modeling of variables that greatly impact short-term predictions, such as yaw-rate, acceleration or forces affecting the vehicle. The ability of the method to predict various interactions is also highly dependant on the training data, making continuous learning and extensive data gathering a requirement for deploying this method on real-world platforms.
B. Prediction Generation

An overview of the prediction system for the CVAE method can be seen in fig. 2. A CVAE was chosen for its ability to learn one to many mappings. As the data contains many complex scenarios with many interactions, the resulting outcome of very similar input frames can differ, requiring the system to be able to encode this mapping present in the data. CVAEs are able to learn typical outcomes of a scene conditioned on the input by encoding both the scene input and the target prediction during training, resulting in the encoded latent representation $z = \mu, \sigma$ being dependant on both the input and target prediction. The decoder then operates not only on the sample from $p(z|x, y)$ but also on the concatenated latent representation of the input $x$. During training the network produces the reconstructed prediction by encoding a sequence of $k = 10$ input images, denoted $x$, corresponding to the data received the 2 seconds leading up to the current time, $t$, and the ground truth of the position of the agent of interest 6 seconds into the future, denoted $y$. Normally, the system produces a prediction that includes all vehicles present in the scenario as one output, however, it can be of interest to examine the predicted behavior of just one agent for analysis. To this end, the network is also taught to produce only the position of one agent if it receives a highlighted indication of the agent in question (seen as the lower input to the upper encoder branch) in addition to the entire scene representation as input. After training, at inference time the ground truth of the prediction is not available and instead the prior (a unit Gaussian) is sampled as the latent representation and concatenated with the conditioned input to produce likely outcomes of the input scenario.

Two separate encoder structures were evaluated for the CVAE method. One which uses 3D convolutions to capture the spatio-temporal aspect of the data, and another which instead has 2D convolutions and LSTM layers for the fully connected feature layers. The other evaluated method of incorporating uncertainty was based on MC Dropout method, its system overview shown in fig. 3. This method is based upon the previously proposed framework in [8], which produces a prediction for the next frame of the traffic sequence by learning Dynamic Filters [11] that operate on the same input sequence of 10 frames as the CVAE method. After having produced one prediction, the oldest frame in the input sequence is discarded and the prediction is used as the newest frame in the input sequence. This is repeated up to 30 times to produce a predicted sequence spanning a horizon of 6 seconds. The method is extended to include uncertainty by adding dropout to the fully connected layers as proposed in [9]. As the authors describe, this method is a variant of Bayesian approximation using Monte Carlo integration and adding dropout to an existing network. It does this by showing that minimizing the loss for a network including dropout and weight decay is akin to minimizing the Kullback-Leibler (KL) divergence between the intractable predictive posterior of a Deep Gaussian Process and a Variational Inference (VI) approximated posterior where matrix columns in the kernel function are randomly set to 0 according to a Bernoulli distribution.

C. Training

The CVAE network was trained with a loss function using both the pixel-wise cross-entropy loss between the constructed prediction frame $\hat{y}$ and ground truth $y$, as well as the KL divergence between the approximated posterior latent representation and its prior, a unit Gaussian (see eq. 1) similar to appendix B in [12].

$$L = \text{crossentropy}(\hat{y}, y) + D_{KL}(q(z|x, y)||p(z))$$ (1)

Training was done using the ADAM algorithm for gradient descent with a batch size of 64 and an initial learning rate of $0.25 \times 10^{-3}$ on a GeForce GTX 1080TI. The MC Dropout method was trained with dropout rate 0.4 and a batch size of 16 using ADAM with an initial learning rate of $1 \times 10^{-4}$. In order to evaluate the predictive capabilities of the proposed system, the intersection between Turingegatan and Oxbackladen located in Sodertalje, Sweden (see fig. 1) was reconstructed as a 3-way intersection (omitting traffic from the east) with focus on vehicle interactions using a simulator based on the IDM model [10], where vehicles could either

![System architecture of the MC Dropout prediction method, derived from previous work in [8] by adding dropout in the fully connected layers and sampling the resulting predictions multiple times.](image-url)
continue straight through the intersection or turn left. The details of the simulator and dataset are provided in section III-A, while the experiments themselves are presented in section III-B.

A. Simulator and Dataset

Although datasets exist for intersections, many are either too small or involve intersections that are too structured to have any stochastic behavior, by for example only allowing left turns in the left lane. To obtain a sufficient amount of training data with focus on complex interactions, a simulator was constructed. While traffic simulators such as SUMO [13] do exist, they did not provide support for extracting all the required features or modifying the desired vehicle behavior. For this purpose a custom simulator was built using the IDM model as the backbone with several modifications to reflect the characteristics of real world data. The simulator provided randomized values for the parameters of the IDM model used for each vehicle and extended the model behaviour by adding traffic rules for yielding for oncoming traffic, modifying velocity profiles to account for road curvature, and finally adding a randomized lateral sway to the lane following. Vehicles with these characteristics were then created at stochastic intervals and given different driving corridors in order to create three separate flows of traffic in the intersection; one going from North to South, one from West to North, and finally a stream originating from the South where cars would randomly turn left or continue straight. Vehicles from the west would turn left and always had the right of way. Vehicles from the North would have to yield for these vehicles and would then always proceed South. The last direction of traffic coming from the south would have to yield for all other vehicles, and could either continue straight through the crossing or turn left. The prediction of this last flow of traffic was chosen as the target of our evaluation due to the complex interactions that could occur. For example, a vehicle might be slowing down at an intersection in order to decrease their speed for a left turn, but could also be slowing down to yield for another vehicle before continuing straight. The focus of the experiments is to assess the proposed system’s ability to correctly take these interactions into account as well as produce reasonable uncertainty in ambiguous situations. A total of 3 hours of traffic was simulated, providing 108000 frames of data where 70%, 20% and 10% were used for training, testing and validation, respectively. The produced dataset is available at https:// strands.pdc.kth.se/public/Intersection_Sim_Data/, along with specifications for the used network architectures.

B. Experiments

As mentioned previously, the simulated intersection was purposely designed to incorporate a high level of vehicle interactions on the northbound traffic agents. To assess the ability of the proposed system to incorporate knowledge of these interactions, as well as provide accurate predictions of traffic agents of interest, an experiment was constructed to focus on the identification of maneuvers and the ability of the different methods to accurately assign a probability to them. Each network was asked to produce the probability of occupation along the different driving corridors, left and right, by sampling 100 produced predictions and approximating a probability from inspecting how many predictions resulted in each maneuver. In order to examine how accurate and useful these probabilities were, the F1 classification score (2) was calculated with a varying probability threshold required for the prediction to be regarded as correct. For example, at threshold 0.5 more than half of the sampled predictions were required to be the same as the ground truth.

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\]

As many cases in the data are complex and ambiguous with very similar input sequences resulting in different maneuvers, ideally the best method would retain a high score between thresholds 0.5 to 0.75 and drop as the threshold increases. A high score for only low confidence thresholds does not mean the method gives informative predictions, as any planner would have to be over-conservative to regard them. If a method shows a high score at the middle thresholds it indicates that it accurately identifies the maneuver as more probable, however retaining this score at the higher threshold values would instead suggest that the method is over confident, placing its probability mass in a too small space to reflect the uncertainty present in the scenarios. To ensure meaningful test scenarios, only input sequences where the agent of interest had not yet crossed the yield line were considered, so that the system would never receive obvious information about the agents intent.
Fig. 5: An example prediction produced by the CVAE system for a scenario where the agent of interest (highlighted in red) makes a left turn, shown at different prediction start times. The images from left to right are the latest image in the input sequence, the ground truth position 6 seconds later, the prediction for the position 6 seconds later, and the probability of turning left or going straight. At t=0.5s, the agent has only recently approached the intersection, and the prediction is highly uncertain due to lack of information. However, later as the vehicle slows down the prediction becomes much more certain of the agent intending to turn left, as there are no cars it would have to yield for when going straight.

IV. RESULTS

Qualitative results from the CVAE method can be seen in fig. 5 and 6, showcasing example predictions from the framework on two different scenarios with different starting times for the prediction (a video of the full sequence can be viewed at https://youtu.be/JuiNuXZn3I). In the first scenario depicted in fig. 5 we can see a relatively simple case of the vehicle of interest, shown in red, turning left. At starting time t=0.5s the vehicle has only recently approached the intersection and as there is no key information available at the time, such as its velocity with regards to its position in the intersection, the resulting prediction is very uncertain and assigns almost equal probability for both maneuvers with a higher uncertainty on the resulting position for the straight maneuver. At t=1.9s however the vehicle has begun to slow down and there is much more information for the predictor to leverage. As we can see from the input image, there are no other vehicles interacting with the agent of interest if it were to go straight through the intersection, meaning there is no reason for it to decrease its speed if that were its intended maneuver. However, if it were to turn left it would have to first make sure to yield for the vehicle approaching from the north, as well as adjust its velocity profile for the curvature in the left turn. This is reflected in the resulting prediction as well, where we can see that the probability of the vehicle going straight is drastically reduced and the system is almost certain of the vehicle executing a left turn.

To show how the prediction system handles complex scenarios with high levels of interaction, another scenario is shown in fig. 6. In this case, there are several vehicles that affect the behavior of the agent of interest. Firstly, there is a leader vehicle in front of the agent, which affects its velocity as it must keep a safe distance from it. Secondly, a vehicle approaching from the North forces both the agent of interest and the leading vehicle to yield for it if they were to make a left turn. Lastly, the vehicle approaching the intersection from the west has the right of way, which would require the agent of interest to slow down and yield for it as well even if it planned on proceeding straight through the intersection. With all of these interactions between the vehicles, it is very difficult to discern anything definitive from the behavior of the agent, even after it has arrived quite close to the yield line. This uncertainty is reflected on the resulting prediction, with the probability of both maneuvers being roughly equal at t=0.2s. Notice also the predicted position of the agent is highly spread out as it is also uncertain which maneuver the leading vehicle will execute, and when. Roughly 1 second later the probabilities remain similar while the future positions have been updated. From the predicted positions, it is possible to see more clearly that agent may also wait for the southbound vehicle to pass before turning left. At t=2.1s it has become very clear the the leading vehicle will turn left and as the agent of interest seems to be speeding up and closing the gap with the leader, we can observe a larger increase in the probability of a straight maneuver. This is still not definitive information however, and while most of the probability is given to the ground truth maneuver there is still some given to both aggressively following the leading
Fig. 6: A further prediction example produced by the CVAE system for a difficult and ambiguous scenario where the agent of interest (highlighted in red) proceeds straight through the intersection, shown at different prediction start times. The images from left to right are the latest image in the input sequence, the ground truth position 6 seconds later, the prediction for the position 6 seconds later, and the probability of turning left or going straight. The scenario includes the agent of interest following a vehicle in front and also vehicles coming both from the west and north which have right of way. The resulting prediction is very uncertain in the beginning, showing similar probability for going straight and turning left. This continues for one second with some fluctuation, with the positions being updated but the maneuver probabilities staying roughly the same. Finally, at $t=2.1s$ when the vehicle starts speeding up after the leading vehicle has turned the prediction quickly changes to more accurately reflecting the ground truth of proceeding straight through the intersection, while still placing some probability to the vehicle turning left immediately with the preceding vehicle or waiting for a suitable gap.

Quantitative results comparing the different methods of modeling uncertainty can be seen in fig. 7. To examine how well each method encompassed the uncertainty in the data, the F1 classification score for the predicted maneuver was calculated with varying degrees of certainty required for the prediction to be regarded as correct. This means that for example at 0.5 on the x axis, more than half of the 100 samples of predictions from the method had to be the same as the ground truth. From the graph we can see that the CVAE method which uses 3D convolutions in the encoder performs the best, with MC dropout being a close second. It is interesting to note here that while MC dropout performs slightly worse, its F1 score does not drop as significantly between required confidence 0.75 and 0.95. This suggests that while its predictions are not as accurate, it places its probability mass in a smaller area, making its prediction more confident, though not necessarily more correct. This behavior may however change if different values of weight decay and dropout rate were to be used in the method. Although the CVAE method using LSTMs performs the best at the lowest required confidence, it seems as if it is unable to find an acceptable encoding of the aleatoric uncertainty in the data, resulting in largely varying predictions that do not have high enough confidence to be useful. This could potentially be remedied by using convolutional LSTMs.

V. CONCLUSIONS AND FUTURE WORK

In this work we present a novel approach for predicting the future positions and maneuvers of traffic agents in intersec-
The proposed method considers complex interactions while also taking into account the stochastic behavior of agents and ambiguous effects of interactions on maneuvers. We compared two separate ways of incorporating uncertainty into the system, Monte-Carlo Dropout and Conditional Variational Auto Encoders (CVAE) and found that the latter better encapsulates the stochastic nature of the complex traffic scene that was considered. Although encoding the spatio-temporal aspect of the data with an LSTM in the CVAE method showed to perform poorly in comparison to using 3D convolutions, it could be possible to improve the performance by using convolutional LSTM layers instead. At the time of writing, the proposed framework has only been tested on having the center of the intersection as the reference point for the generic input representation. While this is still feasible for prediction solutions based on Vehicle-to-Infrastructure (V2I) communication, planned future work includes examining the method’s ability to generalize when having the ego vehicle as the reference point as in previous work, as well as extending the dataset to include several types of intersections. When data acquisition is possible, it would also be of interest to test the method on a smaller real-world dataset after training on both simulated and real-world data. Another extension of interest would be to regard the CVAE-based prediction system could also be used to identify unusual situations showing high risk by training the network on typical safe driving behavior and then examining how far from the prior new and dangerous situations are mapped. Further future work includes extending the framework to also provide explanations of resulting predictions, leveraging methods such as GradCAM [14] or meaningful perturbation of input [15].

**ACKNOWLEDGMENT**

We are grateful to Vinnova project 2016-02547: IQPilot - City Transport with autonomous heavy vehicles for funding the work.

**REFERENCES**