Deep Radar Sensor Models for Accurate and Robust Object Tracking

Jasmin Ebert[1], Thomas Gumpp[2], Sebastian Münzner[1], Alex Matskevych[1],
Alexandru P. Condurache[1,3] and Claudius Gläser[2]

***Abstract*— Object tracking is one of the key challenges for perception systems of autonomous vehicles. Recursive Bayesian state estimation can be used to obtain object tracks. Both the measurement association and the object update within such Bayesian filters rely on sensor measurement models. These models offer an approximation of the expected sensor values that can be error-prone due to a mismatch between model and reality. The discrepancy is caused by the limited descriptive capacity of measurement models since sensor measurements are highly object and situation dependent.**

We address this problem in a data-driven approach by using Deep Neural Networks (DNNs) to learn situation dependent sensor measurement models. In detail, the DNN acts as a virtual sensor that uses current sensor measurements to regress necessary corrections of predicted object states. It can be directly plugged into existing tracking frameworks, substituting the previously hand-modeled association and update steps during Bayesian Filtering. We apply the proposed DNN-based measurement models to the problem of vehicle tracking using radar data in an Extended Kalman Filter setup and compare it to a classical closest reflex and an L-shape measurement update model. Extensive evaluation on a real-world dataset shows that our model improves performances significantly compared to state of the art methods.

I. INTRODUCTION

Driverless vehicles require an accurate understanding of their surroundings. Therefore, they need to perceive all relevant properties of the stationary and dynamic environment. Due to their measurement principles, radar sensors provide accurate distance and relative velocity information of every single reflection perceived in the environment. Moreover, radar sensors are robust to different weather conditions. Consequently, many current driver assistance systems rely on radar sensors to describe dynamic traffic participants.

In order to model traffic participants, object tracks consisting of a time-series of object states are widely used. They can be obtained via recursive Bayesian state estimation, e.g. using Kalman Filters (KF) [1]. For each object hypothesis, the object state is first predicted to the current time frame using object motion models. Afterwards, new measurements are associated to the object and used to update the object state. For that matter, measurement association and object update rely on sensor measurement models. One key challenge of the update is to estimate the object state using measured radar reflexes without knowing their origin on the extended object body.

[1]Engineering Cognitive Systems, Automated Driving, Chassis Systems Control, Robert Bosch GmbH, 71229 Leonberg, Germany
[2]Advanced Autonomous Systems, Corporate Research, Robert Bosch GmbH, 71272 Renningen, Germany
[3]Institute for Signal Processing, University of Lübeck, Germany
Firstname.Lastname@de.bosch.com

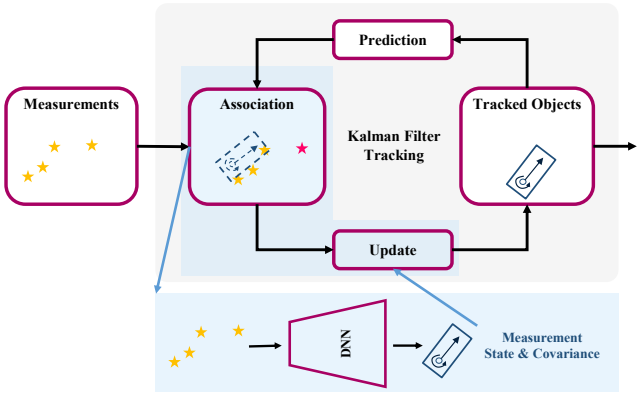

Fig. 1: Association and measurement update using learned sensor measurement models within a classical Kalman Filter setup. Yellow stars: measured radar reflexes; pink star: non-associated radar reflex; blue box: state of the target vehicle.

We address this problem via a data-driven object tracking approach by training a Deep Neural Network (DNN) to learn situation dependent sensor measurement models. The DNN uses measurements to regress correction values of predicted object states as illustrated in Figure 1. Hence, it acts as a virtual sensor transforming distributed measurements into a single meta-measurement. We propose a data-driven solution for both the measurement's mean and covariance. Due to the modular nature of this approach, the model can be directly plugged into existing tracking frameworks, thereby substituting the previously used association and update steps in Kalman Filtering. We apply the learned sensor measurement models to the problem of vehicle tracking using radar data.

II. RELATED WORK

Correcting object states using multiple measurements per timeframe is a twofold process. First, new measurements need to be associated to a given object. Second, this set of reflex samples needs to be related to the object's state. Sensor models serve both purposes by providing an approximate mapping between real-world objects and sensor measurements while taking sensor characteristics into account.

A thorough review of existing sensor models is provided in [2], where the authors differentiate between models defining a set of points on a rigid body [3], [4], [5] and models approximating the spatial distribution of measurements [6], [7], [8]. Sensor models of both categories may rely on physics-based modeling [9], [10], [11] or more recent machine-learning based approaches [12], [13].

The general thought of physics-based modeling is to calculate the propagation of electromagnetic waves in complex environments. Set of points and spatial sensor models relying

on these ray-tracing models incorporate prior knowledge in form of chosen reference points or the choice of analytic functions to describe likelihood values across space.

Due to the object- and situation-dependent nature of sensor measurements, their models can only offer approximations. It is likely to assume that hand-designed models are not able to fully exploit the information offered by the set of raw measurements. One way to overcome manual engineering is to use data-driven solutions, namely machine learning models. Berthold et al. [14] conducted first experiments on the modeling capacity of radar data for contour estimation. They accumulated radar measurements to model the spatial distribution of locations given a certain target vehicle with different orientations and visually proved its capabilities. A first machine learning approach of automotive radar simulation was conducted by Wheeler et al. [12]. They constructed stochastic radar models trained on measurements for simulation purposes. These models could be used for association within a standard tracker. However, it is non-trivial to exploit the information of distributed measurement samples to update extended objects. Scheel et al. [13] aimed to relate measurements to the vehicle state using conditional density functions by learning a variational Gaussian mixture model. The model outperformed a hand-designed approach within a finite set statistics based multi-object tracker.

For the subsequent measurement update step, a distributed set of associated reflex samples needs to be related to the object's state. Classical approaches dealing with spread measurements either cluster the measurements out of the distributed samples to update the object or correct for each measurement separately [2], [15]. Alternatively, L-shapes or boxes can be fitted to the distribution [16], [17]. The latter is related to object detection for which a vast amount of machine learning approaches exist [18], [19]. Moreover, such detectors can provide uncertainty estimates, which can be beneficial in tracking applications [20], [21], [22].

To the best of our knowledge, there is currently no learned sensor measurement model that fulfills both measurement association and update for spread reflexes. The proposed approach can be directly incorporated into existing KF trackers, substituting the available hand-designed models. The size of the models is comparatively small since it only needs to learn a subtask within the KF procedures.

In the following paragraph we discuss how to integrate our DNN approach into a Kalman Filter framework. Please note that object lifecycle management is not within the focus of this paper. It rather remains within the scope of the employed overall tracking framework.

III. TRACKING WITH LEARNED SENSOR MEASUREMENT MODELS

A. Kalman Filter based Tracking

Kalman filtering can be divided into two steps [23]: First, the object state x_{k-1} at time step $k-1$ is propagated in time using process functions, also called motion models f, control input u_{k-1} and the Gaussian process noise w_{k-1} with normal probability distribution $p(w_{k-1}) \sim \mathcal{N}(0, Q_{k-1})$ and process noise covariance Q_{k-1}

$$x_k^- = f(x_{k-1}, u_{k-1}, w_{k-1}). \tag{1}$$

For nonlinear motion models f, the extended KF linearizes around the current mean and covariance [1]. The covariance matrix P_k^- of the predicted state x_k^- is estimated using the Jacobian A_k of f with respect to x and the Jacobian W_k of f with respect to w

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T. \tag{2}$$

Secondly, the predicted state x_k^- and its covariance P_k^- are corrected using sensor measurements z_k and their noise v_k, which is normally distributed according to $p(v_k) \sim \mathcal{N}(0, R_k)$ with measurement noise covariance R_k. Measurements first need to be associated to the predicted objects and then used to correct their states. The association step is traditionally handled outside the KF. Next, the actual measurement update of the KF update step uses sensor measurement models h, since object states need to be mapped to the measurement in order to determine the offset. The complete update step with the Jacobian H_k of h with respect to x and the Jacobian V_k of h with respect to v is given by

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1}, \tag{3}$$

$$x_k = x_k^- + K_k(z_k - h(x_k^-, v_k)), \tag{4}$$

$$P_k = (I - K_k H_k) P_k^-. \tag{5}$$

The calculated matrix K_k (3) is the so-called Kalman gain, that minimizes the a posteriori error covariance P_k (5) and gives a measure to what extent the predicted state x_k^- is corrected using the measurements z_k (4).

B. Learned Sensor Measurement Models

In case of spread measurements along an extended target object each measurement has to be related to the object state, usually by determining associated reference points on the object. Alternatively, associated measurements can be transformed into a single meta-measurement that represents the object state. This single meta-measurement is subsequently fed into the update step. In order to avoid hand-crafted solutions we suggest to learn sensor measurement models that have the capacity to fulfill both the association and the measurement update step and implicitly learn all object and situation dependencies directly from data as illustrated in Figure 1. The models receive current sensor measurements normalized according to the state prediction as an input and yield state corrections of the predicted objects as an output.

These learned sensor models hence can be described as virtual sensors, that map measurements into another abstract dimension. Thereby, both the association of distributed measurements as well as the transformation into a meta-measurement is implicitly solved by the trained neural network model m

$$z_k^{DNN} = m(z_k). \tag{6}$$

The model can be simply plugged into any existing KF tracker's measurement update step

$$x_k = x_k^- + K_k(z_k^{DNN} - h^{meta}(x_k^-, v_k)). \tag{7}$$

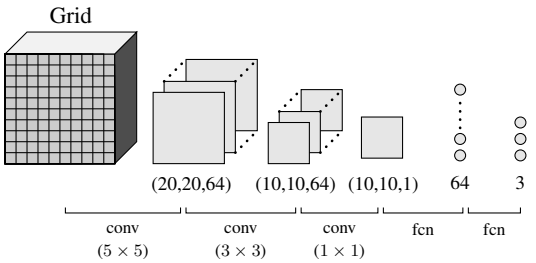

Fig. 2: Grid Net: Convolutional neural network based on an input grid. Reflexes are accumulated into a spatial grid that includes reflex attributes in its depth. A total number of three refined reflex attributes are used. The grid input is subsequently processed by convolutional (conv) and fully connected (fcn) layers.

Consequently, the sensor measurement function h^{meta}, that maps the object state to the meta-measurement space, can be a constant or even an identity mapping due to the outputs of the virtual sensor m.

To derive measurement covariance matrices R of the virtual neural network sensor we divide the sensor field of view (FoV) into a polar grid and estimate R for each grid cell individually. Hence, we apply measurement noise matrices, which are azimuth and distance dependent. In detail, noise matrices are estimated for each grid cell by calculating the deviations of the network model's prediction z_k^{DNN} to the ground truth gt_k

$$R_k^{DNN} = \mathrm{Cov}[z_k^{DNN} - gt_k]. \tag{8}$$

Similar to the measurement model the trained covariance matrices can be directly included into the existing tracker resulting in the modified measurement update equations

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k^{DNN} V_k^T)^{-1} \tag{9}$$

$$x_k = x_k^- + K_k (z_k^{DNN} - h^{meta}(x_k^-, v_k)) \tag{10}$$

$$P_k = (I - K_k H_k) P_k^- \tag{11}$$

The general setup is usable for all association and update tasks and is directly applicable to linear, extended and unscented KFs.

C. Network Architectures

Different network architectures are used to learn the sensor measurement models, namely a convolutional neural network based on a grid input (Grid Net) as depicted in Figure 2 and a point processing network architecture based on adaptive lists (Point T-Net) shown in Figure 3.

The Grid Net's input is a square spatial grid of 10 m width and length with a cell size of 0.25 m by 0.25 m centered at the predicted object position. The depth of the matrix contains a processed subset of reflex attributes, namely radial velocity and radar cross section of the measured reflex. As soon as multiple measurements fall into the same grid cell, only the attributes of the strongest reflex are used. The number of reflexes measured in each cell of the spatial grid is encoded as their sum. The network consists of three convolutional and one fully connected layer, which have been optimized using Bayesian hyperparameter optimization with Hyperband [24]. In total, the network holds 45,316 trainable parameters. The count of neurons in the output layer determines the number

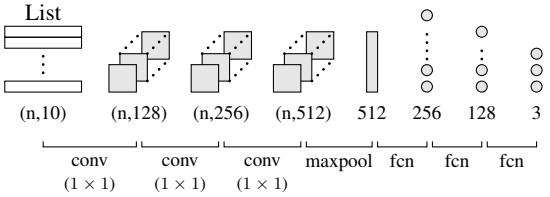

Fig. 3: Point T-Net: Point processing network based on adaptive list input. The input list varies in length and depends on the number of reflexes n per timeframe. Each element of the list is a vector that contains all ten measured raw reflex attributes. The adaptive list input is subsequently processed by convolutional (conv), maxpooling (maxpool) and fully connected (fcn) layers.

of state attributes the network regresses. In principle, a bigger number of attributes could be learned, which would increase the number of output nodes. In our case, the x and y position of the target vehicle's center and the orientation of the target car is learned.

The proposed Point T-Net architecture is inspired by the T-Net which was published as a transformation network within the PointNet introduced by Qi et al. [25]. Its early layers learn independent features per radar reflex, which are subsequently fused to a global feature vector describing the entire scene. This global feature vector is used in subsequent fully connected layers to regress transformation parameters. Instead of learning transformation parameters, our network is trained to directly predict state corrections. Like in the Grid Net architecture, the target vehicle's center point in x and y direction and its orientation is learned. In total, the network holds 330,755 trainable parameters.

IV. Experimental Setup

The introduced measurement updates with data-driven models need to be trained using real-world radar data. This section describes the dataset, training setup, baseline methods and metrics, which were used to train and evaluate learned sensor measurement models.

A. Dataset

We recorded a dataset utilizing one recording vehicle and one target vehicle equipped with Differential Global Positioning System (DGPS) sensors that give precise information of the position, orientation and velocity of the cars. The measured position is accurate within a 2 cm error interval and the orientation is accurate within 0.005 degrees. The DGPS system provides the target object state at all times and is used as ground truth for training the network. Moreover, the recording vehicle was equipped with a front-facing mid-range radar to perceive its environment. In total, we recorded more than 4,000 sequences of 5 seconds of highly dynamic scenes on a test track. An additional test sequence of 50 minutes covering driving on rural and urban roads is used to further evaluate our approach in a real-world setting.

B. Training Data Generation and Setup

The objective of the learned sensor measurement model is to correct predicted object states given incoming measurements. It thereby implicitly also solves the association

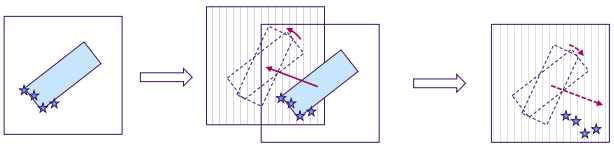

Fig. 4: Artificial training data generation process. Ground truth: blue box; radar measurements: blue stars; artificial offset: dashed line box; offset shift/correction: purple arrows.

of reflexes to the tracked vehicle. Labels for the individual radar reflexes are not available. To generate training data, we add artificial offsets to the object state ground truth in order to simulate possible erroneous KF state predictions. For each training step, the offsets are sampled from a uniform distribution in the range of [-2, 2] meters for both the x and y dimensions and from a uniform distribution in the range of [-22.5, 22.5] degrees for the orientation attribute. The offset generation process is illustrated in Figure 4. Due to the sampling nature, we create an infinite dataset of possible KF predictions given a fixed size of measurements.

Measurement noise covariance matrices are likewise calculated detached of the tracker. This is achieved by afflicting object states with artificial random offsets, just like in the data generation case. A total number of 2500 offsets are introduced for each measurement frame. The ground truth of the validation set is sorted into a polar grid with a cell size of 10 meters and 10 degrees. If a polar grid cell within the sensor's FoV contains less than 10 reflexes, the covariance is not calculated for this cell and is set to a default value.

The neural network weights are learned using a mean squared error loss function. Since the regressed attributes of the virtual measurement are in different value ranges, we weight the individual loss terms in order to achieve equal optimization of each attribute. We train our network parameters using Adam optimization [26].

C. Data Imbalance Handling

There are different kinds of scenes within our measurements. These scenes are by nature not equally distributed. In real-world driving most of the data recorded by the sensor consists of rear views of target vehicles. Due to this imbalance, learned systems are biased to give preference to improve these scenes, if no countermeasures are taken. Since we do not want biased predictions in our dataset but intend to generalize to all kind of situations, we define a scene signature in order to compensate dataset imbalances and to split our measurements into a train, validation and test set. In detail, this is done by calculating a signature consisting of the mean and variance of some abstract descriptors for each sequence. In general, all kinds of abstract descriptions are conceivable. In our case, the object state attributes orientation and azimuth angle are used. Next, a rareness score is calculated based on the signature for the entire dataset, as shown in Figure 5. The rareness score is defined as the reciprocal number of occurrences of a signature. Our objective is to sample a balanced test and validation set. Sequential importance resampling [27] is used to sample a validation and test set without duplicating data points.

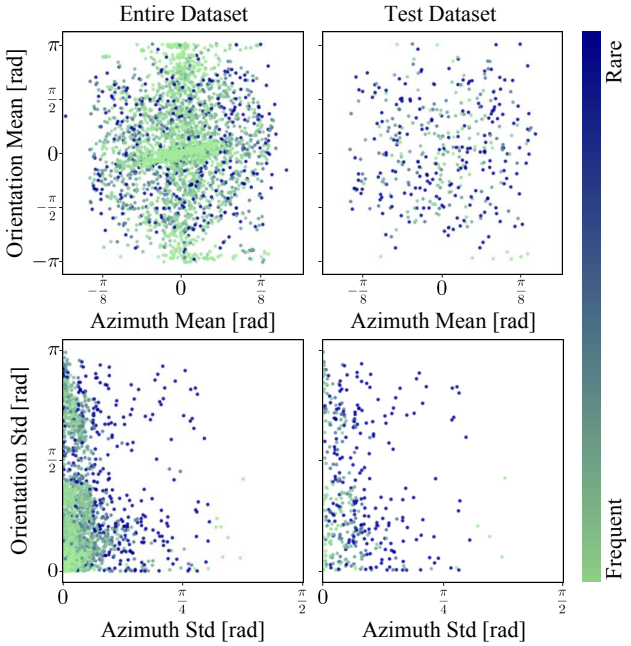

Fig. 5: Signature mean and standard deviation of the entire and test dataset with respect to the scene labels target vehicle's orientation and azimuth angle. Rareness of the sequences is highlighted by the given colormap. Underrepresented signatures of the entire dataset are overrepresented in the test dataset in order to prove generalization to all scenes.

The final statistics of the dataset are 3345 sequences in the training set and 418 sequences in the validation and test set.

D. Classical Reference Models

As a first baseline algorithm, we apply a standard closest reflex measurement model to our dataset. The closest reflex measurement model uses one single reflex having the smallest distance to the predicted state for the measurement update of the vehicle's position. As a second reference, an L-shape sensor model is employed. First, relevant reflexes are determined using a gating mechanism within a radius of 5 meters. Whenever there are more than 5 reflexes measured in the gating corridor, the L-shape sensor model is used to update the object state. If there are less reflexes, the classical closest reflex model is applied. For the KF prediction step, a constant speed heading motion model is used for all approaches.

E. Evaluation Metrics

We use the performance metrics for single-target object tracking proposed by Čehovin et al. [28] to measure the tracker's accuracy and robustness, since labels are given by the double DGPS system only. For assessing the accuracy of the tracker, the intersection over union (IoU) on re-initialized tracks is computed. Re-initialization is the process of setting the predicted object state to the ground truth as soon as the IoU falls below a threshold of 10^{-4}. The IoU is calculated using the DGPS ground truth and the KF state after updating it with our deep sensor model.

To measure the robustness of the tracker, the failure rate on re-initialized tracks is determined. The failure rate is defined

Model	Model w/o tracking	Model w/ tracking	
	Accuracy [IoU]	Accuracy [IoU]	Robustness [Failure Rate]
Closest Reflex	-	0.374	$2.63 \cdot 10^{-2}$
L-shape	0.241	0.456	$9.94 \cdot 10^{-3}$
Grid Net	0.544	0.580	$\mathbf{1.15} \cdot 10^{-3}$
Point T-Net	**0.718**	**0.728**	$1.36 \cdot 10^{-3}$

TABLE I: Performance of sensor models on test track scenes.

as the average number of track losses within a sequence. A track loss is encountered as soon as the IoU of the DGPS ground truth and the updated state vector is less than 10^{-4}.

In order to judge the plain performance of the DNN without any temporal filtering, average overlap is additionally calculated on randomly shifted object states as depicted in Figure 4. These artificial offsets mimic potential prediction errors that may arise from erroneous motion models. We define this metric as accuracy of the model w/o tracking. Since the closest reflex model is not able to yield orientation corrections, it is not evaluated detached from the tracker.

V. RESULTS & DISCUSSION

Evaluation results in Table I show that both the Point T-Net and the Grid Net outperform the classical closest reflex model and the L-shape model by a large margin. As expected, amongst the classical sensor models the L-shape model reaches a much better performance than the closest reflex model. Although the L-shape model is only slightly worse in accuracy than the Grid Net, it is by far not as robust as any of the data-driven approaches in a tracking setting. While the robustness of both data-driven approaches are comparable, the Point T-Net yields significantly better results than the Grid Net with respect to accuracy. There are multiple reasons: First, the Point T-Net is not affected by discretization errors compared to the grid network, which uses a fixed 2D grid to accumulate the reflexes. Second, the architecture design of the Point T-Net, which first extracts features out of each reflex before fusing them into a global vector, is expected to perform better than the Grid Net's design, which first fuses the input and then builds a combined feature. Finally, the proposed Point T-Net has a higher number of trainable parameters and, consequently, a higher modeling capacity.

The sensor models' performance within the KF tracker reflect the trends given by the performance of sensor models detached from the tracker. As expected, the overall tracking accuracy is slightly better when benefiting from temporal filtering compared to the single shot evaluation. Furthermore, the usage of learned measurement noise matrices allows the tracker to properly weight regressions in individual frames.

For measurement noise covariance investigation, Figure 6 shows the resulting norm of every measurement covariance matrix per grid cell. The matrix norm is a measure of the overall measurement uncertainty. It indicates that uncertainty values are lower in the close-range center of the radar's FoV, although there are some outliers at the borders due

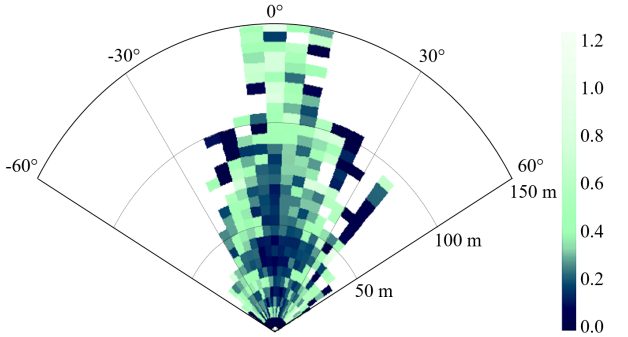

Fig. 6: Data driven covariance estimation. The norm of each grid cell is visualized in its polar representation. Results show a tendency to smaller uncertainties in the center of the FoV.

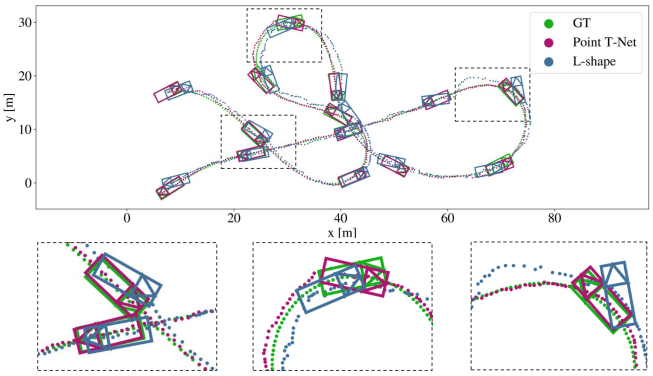

Fig. 7: Exemplary comparison of object tracks obtained by application of the Point T-Net and L-shape model overlaid with the ground truth (GT).

to limited data statistics. High uncertainty values are mainly represented at high distances and azimuth angles.

For a qualitative impression of the tracking performance of the Point T-Net compared to the L-Shape model, we use a bird eye view of object tracks as in Figure 7. The learned sensor measurement model outperforms the L-shape model in sharp curves and for larger distances. Furthermore, our data-driven approach is favorable for longitudinal distance estimation.

Since data-driven approaches are highly dependent on the training data, there is a risk that the models are not robust to noisy measurements in dynamic real-world scenes. In particular, our training data covers recordings of single objects on a test track, which is significantly different to the usual multi-object settings on public streets. To investigate this, we applied our previously trained models from the test track scenario to a real-world dataset of 50 minutes driving on rural and urban roads.

The evaluation results of the different sensor models are summarized in Table II. The closest reflex measurement model shows similar results in both the real-world and the highly dynamic test track scenes. The L-shape model also demonstrates similar performance in accuracy, but its robustness dropped by a factor of four. This could be caused by the more challenging association problem. Due to the increased complexity within the data, the tracking accuracy of our sensor models dropped. As expected, the robustness of the learned sensor measurement models decreases significantly when compared to the performance on the test

Model	Accuracy [IoU]	Robustness [Failure Rate]
Closest Reflex	0.397	$2.29 \cdot 10^{-2}$
L-shape	0.411	$4.05 \cdot 10^{-2}$
Grid Net	0.553	$1.63 \cdot 10^{-2}$
Point T-Net	**0.620**	$\mathbf{1.19} \cdot 10^{-2}$

TABLE II: Real-world performance of sensor models.

track scenes. But still in this real-world setting, our proposed sensor models outperform the classical closest reflex and the L-shape model in both accuracy and robustness. This demonstrates that our test track trained models have the potential to generalize to real-world scenarios.

Although our implementation is not intended for real-time usage, a first run time analysis on a Nvidia GTX 1080 showed that the novel measurement step took less than 5 ms for both architectures. The computational complexity of the Grid Net is $8.7 \cdot 10^6$ FLOPs. Point T-Net requires $1.7 \cdot 10^8$ FLOPs. Our future work will focus on finding suitable trade-offs between network performance and computational complexity.

VI. CONCLUSION

In this paper, we proposed to augment radar based object tracking with machine learned sensor measurement models. Deep neural networks were used to implicitly learn measurement associations and measurement corrections of predicted object states. Moreover, we presented a data-driven technique for covariance estimation, given the learned measurement models. Due to the modular nature of our approach, the trained model can be directly plugged into existing tracking frameworks, thereby substituting the previously used association and update steps in Kalman Filtering.

When evaluating our approach, we showed significant improvement compared to a conventional tracking system using a closest reflex and an L-shape model both in terms of accuracy and robustness.

As a next step, we want to extend the approach to several object classes, multiple objects and different vehicle extensions in dynamic real-world scenes. In addition, we plan to learn situation dependent measurement covariances utilizing neural network output uncertainties.

REFERENCES

[1] H. Sorenson, *Kalman Filtering: Theory and Application*. New York, IEEE Press, 1985.

[2] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *Journal of Advances in Information Fusion*, vol. 12, no. 2, pp. 139–174, 2017.

[3] M. Buhren and B. Yang, "Simulation of automotive radar target lists using a novel approach of object representation," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 314–319.

[4] J. Gunnarsson, L. Svensson, L. Danielsson, and F. Bengtsson, "Tracking vehicles using radar detections," in *2007 IEEE Intelligent Vehicles Symposium*, 2007, pp. 296–302.

[5] L. Hammarstrand, L. Svensson, F. Sandblom, and J. Sorstedt, "Extended object tracking using a radar resolution model," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 48, pp. 2371–2386, 2012.

[6] K. Gilholm, S. J. Godsill, S. Maskell, and D. Salmond, "Poisson models for extended target and group tracking," in *SPIE Optics + Photonics*, 2005.

[7] K. Gilholm and D. Salmond, "Spatial distribution model for tracking extended objects," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 152, no. 5, pp. 364–371, 2005.

[8] P. Broßeit, B. Duraisamy, and J. Dickmann, "The volcanormal density for radar-based extended target tracking," in *IEEE Intelligent Transportation Systems*, 2017, pp. 1–6.

[9] C. Knill, A. Scheel, and K. Dietmayer, "A direct scattering model for tracking vehicles with high-resolution radars," in *IEEE Intelligent Vehicles Symposium*, 2016, pp. 298–303.

[10] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche, "An abstracted radar measurement model for extended object tracking," in *IEEE Intelligent Transportation Systems*, 2018, pp. 3866–3872.

[11] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H. Wuensche, "A radar measurement model for extended object tracking in dynamic scenarios," in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 770–776.

[12] T. A. Wheeler, M. Holder, H. Winner, and M. J. Kochenderfer, "Deep stochastic radar models," in *IEEE Intelligent Vehicles Symposium*, 2017, pp. 47–53.

[13] A. Scheel and K. Dietmayer, "Tracking multiple vehicles using a variational radar model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3721–3736, 2018.

[14] P. Berthold, M. Michaelis, T. Luettel, D. Meissner, and H.-J. Wuensche, "Radar reflection characteristics of vehicles for contour and feature estimation," in *Sensor Data Fusion: Trends, Solutions, Applications*, 2017, pp. 1–6.

[15] S. Bordonaro, P. Willett, Y. Bar-Shalom, M. Baum, and T. Luginbuhl, "Extracting speed, heading and turn-rate measurements from extended objects using the EM algorithm," *IEEE Aerospace Conference Proceedings*, 2015.

[16] J. Schlichenmaier, N. Selvaraj, M. Stolz, and C. Waldschmidt, "Template matching for radar-based orientation and position estimation in automotive scenarios," in *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, 03 2017, pp. 95–98.

[17] F. Roos, D. Kellner, J. Dickmann, and C. Waldschmidt, "Reliable orientation estimation of vehicles in high-resolution radar images," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, pp. 1–8, 2016.

[18] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, 2020 (to appear).

[19] M. F. Meyer and G. Kuschk, "Deep learning based 3d object detection for automotive radar and camera," *2019 16th European Radar Conference (EuRAD)*, pp. 133–136, 2019.

[20] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection," in *IEEE Intelligent Vehicles Symposium*, 2019, pp. 1280–1287.

[21] G. P. Meyer, A. Laddha, E. Kee, C. Vallespi-Gonzalez, and C. K. Wellington, "Lasernet: An efficient probabilistic 3d object detector for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 677–12 686.

[22] D. Feng, L. Rosenbaum, C. Glaeser, F. Timm, and K. Dietmayer, "Can we trust you? on calibration of a probabilistic object detector for autonomous driving," *arXiv preprint arXiv:1909.12358*, 2019.

[23] G. Welch and G. Bishop, "An introduction to the kalman filter," *Proc. Siggraph Course*, vol. 8, 01 2006.

[24] S. Falkner, A. Klein, and F. Hutter, "BOHB: Robust and efficient hyperparameter optimization at scale," in *International Conference on Machine Learning*, 2018, pp. 1437–1446.

[25] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.

[27] J. S. Liu, R. Chen, and T. Logvinenko, "A theoretical framework for sequential importance sampling with resampling," in *Sequential Monte Carlo Methods in Practice*, 2001, pp. 225–246.

[28] L. Cehovin, A. Leonardis, and M. Kristan, "Visual object tracking performance measures revisited," *IEEE Transactions on Image Processing*, vol. 25, no. 3, pp. 1261–1274, 2016.