Multimodal Motion Prediction Based on Adaptive and Swarm Sampling Loss Functions for Reactive Mobile Robots

Ze Zhang, Emmanuel Dean, Yiannis Karayiannidis, Knut Åkesson

***Abstract*— Making accurate predictions about the dynamic environment is crucial for the trajectory planning of mobile robots. Predictions are by nature uncertain, and for motion prediction multiple futures are possible for the same historic behavior. In this work, the objective is to predict possible future positions of the target object for the collision avoidance purpose for mobile robots by considering different uncertainty by combining a sampling-based idea with data-driven methods. More specifically, we propose a major improvement on a loss function for multiple hypotheses and test it with convolutional neural networks on motion prediction problems. We implement post-processing heuristics that produce multiple Gaussian distribution estimations, and show that the result is suitable for trajectory planning for mobile robots. The method is also evaluated with the Stanford Drone Dataset.**

I. INTRODUCTION

Motion prediction plays an important role in many technologies such as autonomous driving [1], [2], surveillance and security systems [3], interactive industrial and service robots [4], [5]. For mobile robots navigating in an environment populated by heterogeneous agents such as humans and manually operated vehicles, the trajectory planning problem is facilitated by the use of predictions about future positions of the mobile agents. The predictions enable the generation of smooth trajectories that do not cause the robot to make unnecessary sudden changes typically encountered when only the current positions are available [6].

Motion prediction is challenging due to uncertainty about the long-term goal and the short-term path taken by the agent. This brings two sorts of uncertainty: *aleatory* and *epistemic* [7]. The aleatoric uncertainty is the indissoluble intrinsic randomness and can be estimated given enough data. It can be parameterized statistically as variances or ranges. The epistemic uncertainty is the uncertainty presumably caused by the lack of knowledge and might be more significant when the randomness in motion is small. This situation commonly exists in factories and warehouses where accessible areas and rules of action are regulated. In motion prediction, epistemic uncertainty appears with multiple potential destinations or alternative paths. Taking each choice as a mode, multimodality is the main representation of epistemic uncertainty.

In this work, we focus on multimodal motion prediction assuming that the target object is detected and tracked by a vision system, and propose a deep learning method producing multiple hypotheses of the future position of the target object by improving a multiple hypothesis meta-loss. Hypotheses are treated as samples from unknown ground truth distributions and processed by an unsupervised clustering algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8], to distinguish different modes and rebuild the distribution, see Fig. 1. Such a Clustering and Gaussian Fitting (CGF) approach provides stable parameterized and geometric short-term predictions that can be used by Model Predictive Control (MPC) controllers as discussed in [6]. Convolutional neural networks (CNNs) are used as the backbone to directly handle images captured by cameras, thus the model could assimilate the contextual cues as well as the influence of other objects in the scene. An inference procedure of the approach is depicted in Fig. 2.

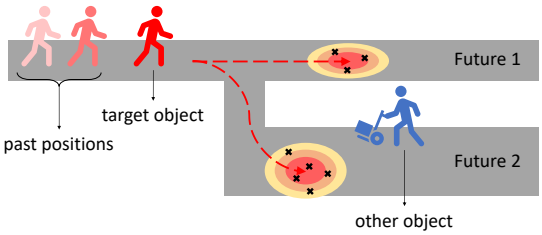

Fig. 1. The concept of multimodal motion prediction by sampling. In this figure, the person in red is our target object while the blue entity is a moving obstacle. Given the past positions of the target object, combining the environmental context, the idea is to generate samples (black crosses) of its possible future positions and estimate the uncertainty of the prediction.

A. Motion Prediction

Motion has different interpretations. Our work aims at future position prediction, thus in this paper motion indicates movement in 2D space. The development of motion prediction problems is introduced in [9]. Earlier motion prediction methods model motion as physical processes first and make predictions according to the models, such as the Social Force [10] and the Reciprocal Velocity Obstacle model [11].

More recent studies treat motions as patterns. From classic machine learning approaches, such as Gaussian Processes [12], to deep learning methods, researchers place attention on discovering hidden states of target objects and their interactions with external factors. Social LSTM [13] and Social Attention [14] explore the social influence on human motion using Recurrent Neural Networks. However, the static environment is neglected, which is an important source of epistemic uncertainty. Therefore, social-based models suit

*We gratefully acknowledge financial support from Chalmers AI Research Centre (CHAIR) and AB Volvo (Project ViMCoR/AiMCoR) and the support from Per-Lage Götvall at Volvo Group Truck Operation.

The authors are with Division of Systems and Control, Electrical Engineering, Chalmers University of Technology, Sweden {zhze, deane, yiannis, knut}@chalmers.se

better for human motion predictions in open areas. To include the surrounding information, SS-LSTM [15] encodes scenes by CNNs and feeds the latent information to LSTM encoders. In [16], a multimodal trajectory forecasting approach based on semantic segmentation networks is introduced, where two semantic segmentation networks are used to predict the waypoints first and the trajectory accordingly. In [17], convolutional recurrent neural networks are used to encode the history information divided into grid cells and decode the features into multimodal trajectory predictions.

B. Multiple Choice Learning

Multimodality means probability distributions with more than one mode. Multimodal estimation is important for many real-world applications involving multiple solutions. Multiple Choice Learning (MCL) [18] is one manner to estimate multimodality. A pipeline of combining MCL with deep learning methods is proposed in [19] as the meta-loss.

In [20], the meta-loss is improved with the idea of generating multiple hypotheses and estimating modes in the form of mixture models such as Gaussian Mixture Models (GMMs) by training Mixture Density Fitting (MDF) networks. A two-stage CNN-based model is used, which first generates several hypotheses of the future position and then uses the MDF to refine the estimation into mixture models. This is necessary since the hypotheses generated in the first stage are not accurate. The MDF screens every hypothesis and keeps the accurate ones. As discussed in [6], the irregular contour of GMMs cannot be easily added as constraints for the trajectory planning of mobile robots. A solution is to treat each mixture component separately as an elliptical obstacle. However, as shown in Section V, a GMM generated by the MDF model may have redundant and abnormal components. Instead, we propose another CGF method to solve this problem.

The main contributions of this paper are:

- A multimodal motion prediction approach on image data combining deep learning multiple hypothesis estimation with unsupervised clustering and fitting;
- A multiple hypothesis loss to obtain more accurate estimations compared to the loss proposed in [20];
- Benchmark of the proposed loss function against previous approaches on a simulated and a real-world dataset.

II. Preliminaries

We introduce the definitions of hypothesis and modes for this context, as well as the multiple hypotheses meta-loss.

A. Hypotheses and Modes

Given a question, multiple candidate answers might exist. For instance, Fig. 1 shows two possibilities given the question of the future position of the target object. Therefore, making multiple hypotheses is required. Given n collected answers $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$, the goal is to generate K hypotheses $\{\boldsymbol{h}^k\}_{k=1}^K$ covering as many answers as possible, which in our case are target positions. Since different target positions may show the same tendency, each one can be regarded as a sample from a potential mode. Modes can be interpreted as particular ways in which something is done or takes place. The goal is adjusted to estimate all modes from hypotheses.

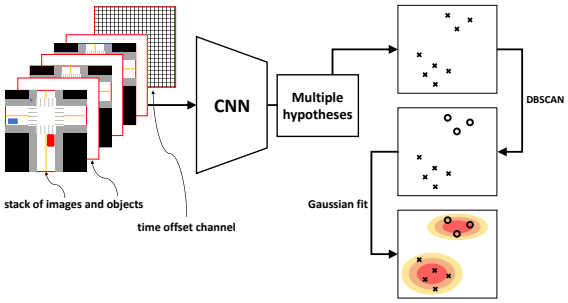

Fig. 2. The inference procedure of the proposed approach. The input is a stack of object masks and images captured by cameras. If multiple time offsets are required for predictions, an extra time offset channel is attached. Multiple hypotheses of the future position of the target object are produced by deep neural networks and further processed by DBSCAN and Gaussian fitting to estimate the final multimodal motion prediction.

To further clarify the connotation of modes, a transformation from hypotheses to modes is very crucial. The simplest way to define modes is to use thresholds. Nonetheless, when there is no universal and explicit threshold, some criteria to distinguish modes are necessitated. Formally, define $\mathcal{M}$ as the set containing all samples of a mode. For $\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{Y}$ and a definition of the distance $d(\boldsymbol{y}_1, \boldsymbol{y}_2)$ between them, they belong to the same mode if and only if their distance is no larger than a given threshold ϵ:

$$\boldsymbol{y}_1, \boldsymbol{y}_2 \in \mathcal{M} \iff d(\boldsymbol{y}_1, \boldsymbol{y}_2) \leq \epsilon \tag{1}$$

or there exists another sample $\boldsymbol{y} \in \mathcal{Y}$ such that $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ can be proved to belong to the same mode with $\boldsymbol{y}$:

$$\exists \boldsymbol{y} \in \mathcal{Y},\ \boldsymbol{y}, \boldsymbol{y}_1 \in \mathcal{M} \text{ and } \boldsymbol{y}, \boldsymbol{y}_2 \in \mathcal{M}' \implies \mathcal{M} = \mathcal{M}' \tag{2}$$

Two target positions $\boldsymbol{y}_1, \boldsymbol{y}_2$ fulfilling either Eq. (1) or (2) are regarded to belong to the same mode $\mathcal{M}$.

B. Multiple Hypothesis Meta-Loss

Assuming a hypothesis generator $f : \mathcal{X} \to \mathcal{H}$, mapping the input $\boldsymbol{x} \in \mathcal{X}$ to the guesses $\boldsymbol{h} \in \mathcal{H}$ of the target position:

$$\{\boldsymbol{h}^k\}_{k=1}^K = f(\boldsymbol{x}) = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(K)}, y^{(K)})\} \tag{3}$$

Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. Given a target position $\boldsymbol{y}$ and a loss function $l(\cdot)$, there are K losses overall $l^{(k)} = l(\boldsymbol{y}, \boldsymbol{h}^k),\ k \in [K]$. The meta-loss [19] for training such models can be simplified as the Winner-Takes-All (WTA) loss [20] as in Eq. (4), where $\delta(A)$ is the general Dirac delta function which is 1 if A is true and 0 otherwise. The basic idea is to only update the hypothesis with the smallest loss.

$$\mathcal{L}_{\text{WTA}} = \sum_{k=1}^{K} w_k l(\boldsymbol{y}, \boldsymbol{h}^k), \quad w_k = \delta(k = \operatorname*{argmin}_i l^{(i)}) \tag{4}$$

There are two issues of the WTA loss in practice. At least one hypothesis will be attracted by at least two targets if there are fewer hypotheses than targets in the neighbor area. Such a hypothesis will have a local minimum in the equilibrium

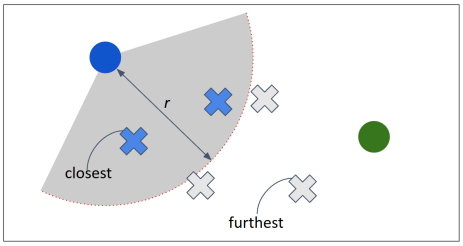

Fig. 3. The concept of the AWTA loss. The two circles are the ground truth while the five crosses are hypotheses. After determining the range r, all hypotheses within the range r will be updated.

point of all targets that attract it. The second problem is that each target only attracts one hypothesis. Many hypotheses will be discarded, which cannot be tracked.

A relaxed WTA [19] is introduced to solve the second problem. Instead of only updating the winner, the remaining ones are also updated with a small relaxation factor. This ensures that no hypothesis is abandoned, but all relaxed hypotheses are trapped by the equilibrium. An Evolving WTA (EWTA) [20] is proposed to solve the first issue. It changes the idea of winner-takes-all to top-winners-take-all, i.e. the top k_{top} winners will be updated. During the training, k_{top} acts as a preset hyperparameter. The strategy is to update all hypotheses at the beginning and keep halving k_{top} once in a while until $k_{top} = 1$. In this way, more hypotheses are updated comparing to the original or relaxed WTA. However, it still leaves some hypotheses at equilibria, which need to be removed by other methods such as MDFs [20].

The EWTA loss lacks the control on every hypothesis especially when $k_{top} = 1$. Thus, a more adaptive way to estimate each hypothesis is needed. Conversely to the original WTA fashion, one solution is to decide if a hypothesis belongs to a mode or not by looking at its distance or similarity to the hypothesis with the smallest loss.

III. Problem Formulation

Given an input $\boldsymbol{x}_i \in \mathcal{X},\ i \in [N]$, and one corresponding target position $\boldsymbol{y}_i \in \mathcal{Y}$, which is a sample from M_i potential modes, K hypotheses $\{\boldsymbol{h}_i^k\}_{k=1}^K \in \mathcal{H}$ are generated to capture all modes. In our case, $\boldsymbol{x}_i$ is a stack of images, which is a 3D matrix. The target position $\boldsymbol{y}_i$ and hypotheses $\boldsymbol{h}_i^k$ are pairs of coordinates indicating the future positions of the target object after a certain amount of time T_{offset}. By classifying these hypotheses according to some specific criterion, $\tilde{M}_i$ clusters are formed as modes. For simplicity, the subscript i is omitted if no ambiguity. The inference procedure is shown in Fig. 2. A mode $\mathcal{M}$ is determined by a cluster of all targets in a certain area and is represented by a Gaussian distribution.

IV. Adaptive and Swarm WTA Losses

In this section, we formulate the Adaptive WTA (AWTA) and the Swarm WTA (SWTA) approaches, and compute explicit multimodal distributions based on them.

A. Adaptive WTA Loss

Motion prediction exhibits multiple uncertainties with an unknown number of modes and dispersal samples. Furthermore, the uncertainties can be time-variant or scene-variant. To solve this problem, we propose an adaptive version of the WTA loss, named the Adaptive WTA (AWTA) loss.

The initial phase is the same as in the EWTA loss. The top k_{top} winners are updated and k_{top} keeps decreasing during training. The difference is that when $k_{top} = 1$, an adaptive range is used to update the hypothesis. Any hypotheses with losses smaller than the range will be updated. The range r is defined as

$$r = \min_{k\in[K]}(l^{(k)}) + \alpha \cdot \left(\max_{k\in[K]}(l^{(k)}) - \min_{k\in[K]}(l^{(k)})\right) \tag{5}$$

where $\alpha \in [0, 1)$ is hyperparameter used to control how close the range is to the smallest loss. If $\alpha = 0$, then it is the same as the WTA loss and the whole progress is the same as the EWTA loss. An illustration of the AWTA loss is in Fig. 3. The formulation of the AWTA loss is defined as:

$$\mathcal{L}_{\text{AWTA}} = \sum_{k=1}^{K} w'_k l(\boldsymbol{y}, \boldsymbol{h}^k), \quad w'_k = \delta(l^{(k)} \le r) \tag{6}$$

The AWTA loss has a clustering effect on the hypotheses since all hypotheses inside the adaptive range will be drawn to the same point. This effect gives a straightforward prediction of the mean position of each component in the ground truth distribution.

B. Swarm WTA Loss

The clustering effect of the AWTA loss gives a better estimation of mean positions, but the aleatoric uncertainty is lost during this process. This problem affects the effectiveness of the result, particularly when the aleatoric uncertainty is substantial. To conquer this deficiency, we modify the AWTA loss by altering the updating rule in (6):

$$\mathcal{L}_{\text{SWTA}} = \sum_{k=1}^{K} w'_k \min_i l(\boldsymbol{y}, \boldsymbol{h}^i) \tag{7}$$

where w'_k is the same as (6). We call this Swarm WTA (SWTA) since the update extent of every hypothesis in the adaptive range depends on the one with the smallest loss, which exhibits a local swarm behavior. The SWTA loss is used after the AWTA loss to cancel the clustering effect and estimate the variance of the ground truth distribution. A comparison amongst EWTA, AWTA, and SWTA is shown in Fig. 4 which shows that the equilibrium point in the EWTA case is not present in AWTA and SWTA cases. Compared to the AWTA case, SWTA losses cancel the clustering effect and give better approximations of the variance.

C. Unsupervised Mode Split

Using the resultant hypotheses, an unsupervised clustering algorithm is applied to explicitly display multiple modes. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [8] is efficient and fits our definition of modes. In the DBSCAN, we use the Euclidean distance as the metric and select ϵ in Eq. (1), and a minimal number of hypotheses N_{cluster} required for a cluster according to the scenario. Once all modes have been split, Gaussian distributions can easily fit for each mode by calculating the mean value and the

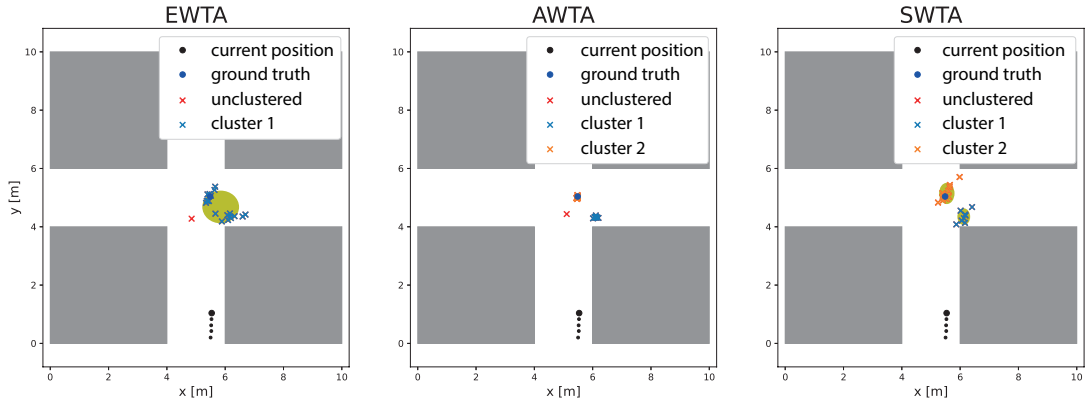

Fig. 4. These figures show a comparison among different WTA losses with CGF on the SID. The simulation is a target vehicle that approaches a crossing. Given its past positions (small black circles), the objective is to predict its possible future positions. The crosses are clustered and unclustered predictions. The yellow ellipses are the approximated Gaussian distributions based on the clustered hypotheses. From the left to the right, the first figure shows the EWTA approach, where some hypotheses are trapped in an equilibrium point. The second and third figures show the hypotheses based on AWTA and SWTA respectively and in both cases the equilibrium point is not present. A simulation of the predictions can be seen at `https://youtu.be/vTDNlaUTbxg`.

variance of all hypotheses belonging to that mode. This CGF method generates parameterized and geometric results without redundant modes, as shown in Fig. 5.

V. Evaluation

A. Experimental Setup

To generate multiple hypotheses, we construct a regression neural network as in [20], with $K \cdot D_o$ outputs, where K is the preset number of hypotheses and D_o is the dimension of target positions. A light ResNet34 [21] is adopted as the backbone together with the multiple hypothesis regression head. It mainly contains four blocks with 3, 4, 6, 3 convolutional layers respectively. Layers have 16/32/64/128 filters in the first/second/third/fourth block. The neural network is trained with different WTA losses. The input of the neural network is, for the current time t and past time h, a stack of images consisting of environment frames $\boldsymbol{I}$ and object location masks $\boldsymbol{L}$, $\boldsymbol{x} = (\boldsymbol{I}_{t-h}, \ldots, \boldsymbol{I}_t, \boldsymbol{L}_{t-h}, \ldots, \boldsymbol{L}_t)$. The neural network generates K hypotheses to compare with the label $\boldsymbol{y}$ and backpropagate the WTA loss. In the AWTA loss, α is set to 0.05. The source code is available online [1].

B. Datasets

The proposed approach is evaluated on a synthetic and a real-world dataset. The synthetic dataset, called Single-object Interaction Dataset (SID), simulates a crossing scene as illustrated in Fig. 4. There is one object approaching the crossing with multiple possible actions. The sampling time is 0.2 sec. The prediction time offset for single future position predictions is $T_{\text{offset}} = 4$ sec and for trajectory forecasting is $T_{\text{offset}} = 0{\sim}2$ sec, i.e. 10 time steps in the future.

The model is also tested on a real-world dataset, the Stanford Drone Dataset (SDD) [22]. The dataset is resampled to 3 FPS. The time offset is $T_{\text{offset}} = 5s$. A crossing scene in the SDD is selected for training and testing as shown in Fig. 7. Three videos under the "hyang" scene of the SDD are used to generate 165/47 trajectories for training/testing.

[1] `https://github.com/Woodenonez/MultimodalMotionPred_SamplingWTACGF_Pytorch`

C. Metrics

To evaluate different models, four metrics are selected to examine different aspects of prediction results. These metrics verify if at least one mode from a prediction is accurate and if the prediction covers the multimodality properly.

- **Oracle error** [18]. At least one predicted mode should be close to the ground truth. This can be measured by the oracle loss that selects the best hypothesis or mode to the ground truth and calculates the loss.
- **Negative log-likelihood** (NLL). The NLL loss is commonly used to train neural networks [20]. It takes the negative logarithm of the likelihood of the ground truth with respect to the estimated distribution.
- **Mahalanobis distance** (MD). It measures the distance between a point and a distribution. Two variants, the Oracle MD (OMD) and the Weighted MD (WMD) [6] are used. The OMD measures the MD from the best-predicted mode to the ground truth. The WMD measures the weighted summation of MDs of each mode.

From the definitions of these metrics, the oracle error evaluates the best-predicted mode pointwise while the OMD evaluates the best-predicted mode with respect to distributions. The NLL loss and WMD analyze the performance of all predictions with all modes. NLL losses are more general for any probability distributions but might tend to infinity, while the WMDs are less numerically vulnerable but only suitable for the evaluation of multimodal distributions.

D. Evaluation

Mixture Density Networks (MDNs) [23] are included as a baseline. MDNs are a type of neural networks producing mixture densities, but are known to be unstable for high-dimensional inputs [20], [6]. On the SID, the performance of the Kalman Filters (KFs) with constant velocity models, MDNs, WTA with CGF, and WTA with MDF are compared.

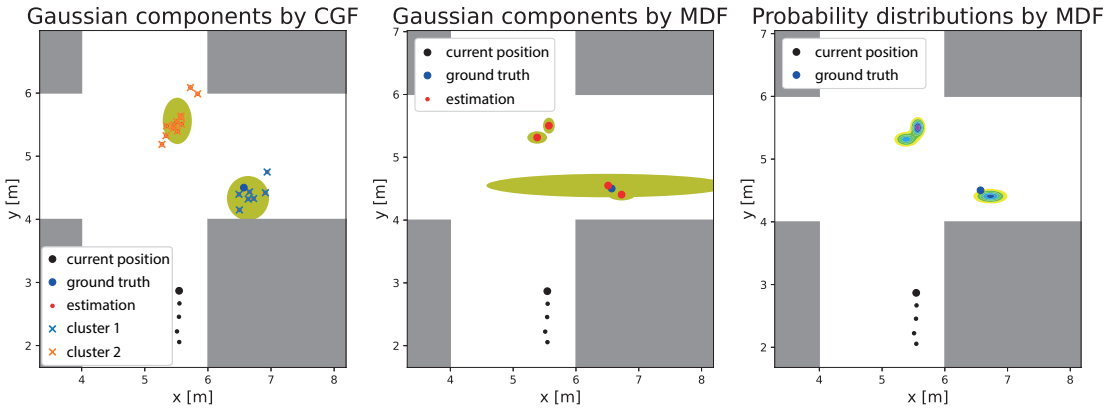

Fig. 5. These figures show the comparison between the CGF and MDF methods. The yellow ellipses are predicted Gaussian components. The first figure shows the prediction result by the CGF method. Since the predicted Gaussian components are directly generated from the hypotheses, there are no redundant or masked abnormal ones. In comparison, even though the MDF method makes accurate predictions on the probability distribution as shown in the last figure, it might contain abnormal components as shown in the middle.

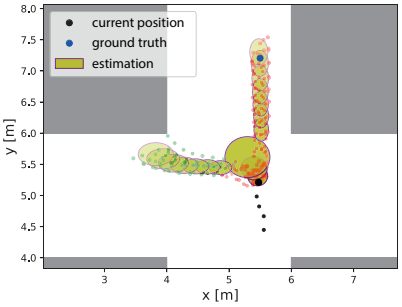

Fig. 6. The trajectory forecasting result with the time offset from 1 to 10. Red and green dots are hypotheses belonging to two different modes. Yellow ellipses are estimated Gaussian distributions, and lighter colors mean further predictions. Here only shows the ground truth at time step 10. A simulation of the predictions can be seen at `https://youtu.be/s-sDAfs5IO8`.

On the SDD, the performance of the Kalman Filters (KFs) with constant velocity models and WTA with CGF are evaluated. MDF methods are not included in the evaluation on the SDD since they do not suit our application. The results are shown in Table I and II. In Table I, since KF and MDN produce single-mode predictions, their WMDs are the same as OMDs. MDF methods in general have slightly better results, but they cannot be used practically for the irregular contour of its predictions. Therefore, the comparison is made among KF, MDN, and CGF methods. Meanwhile, MDF methods are listed to show that the CGF approaches can achieve comparably good results.

In the evaluation on the SID, one can see that the MDF method outperforms the CGF method in terms of the estimation of the probability distribution. However, the components of the estimated mixture density from the MDF are not observable. Some components might be redundant and even abnormal. An example is illustrated in Fig. 5. Another example, which is not shown, is that there might be multiple components overlapping with each other. On the real-world SDD, the clustering effect of the AWTA and the decentralizing effect of the SWTA are shown in Fig. 7. The evaluation also suggests that the proposed losses are better than the Kalman filter and the EWTA loss in terms of the NLL metric. However, multimodality is not shown in the test. This might be due to the fact that there is larger randomness in the SDD compared to the simulation, and the WTA loss is sensitive to large noise and outliers.

TABLE I

EVALUATION RESULTS ON THE SID.

Method \ Dataset	SID Test			
	Oracle	OMD	NLL	WMD
KF	5.248	3.554	41.20	-
MDN	3.120	1.338	2.843	-
EWTA+CGF	0.127	0.566	1.107	5.098
AWTA+CGF	**0.075**	0.875	Inf	8.262
SWTA+CGF	0.088	**0.510**	**0.449**	**4.670**
EWTA+MDF	0.073	0.465	0.141	4.833
SWTA+MDF	0.065	0.527	0.427	5.102

TABLE II

EVALUATION RESULTS ON THE CROSSING SCENE IN THE SDD.

Method \ Dataset	SDD Test
	NLL
KF	8.507
EWTA+CGF	8.139
AWTA+CGF	9.939
SWTA+CGF	**7.828**

E. Trajectory Forecasting

The objective of this work is to provide motion prediction of objects to ATRs. These mobile robots can accomplish intelligent behaviors such as collision avoidance and interaction with workers using the predicted positions. In [6], it is discussed how to combine motion prediction results with MPC controllers. Instead of predicting the moving obstacle's position of one future time instant, the controller needs to know the positions of successive time steps. This is known as trajectory forecasting. In order to make predictions on future positions at different time instants, one more channel is added in the inputs as the time offset channel as in Fig. 2. During training, training data with different prediction time offsets T_{offset} is shuffled together and the last channel of the stacked input images indicates the time offset. For example,

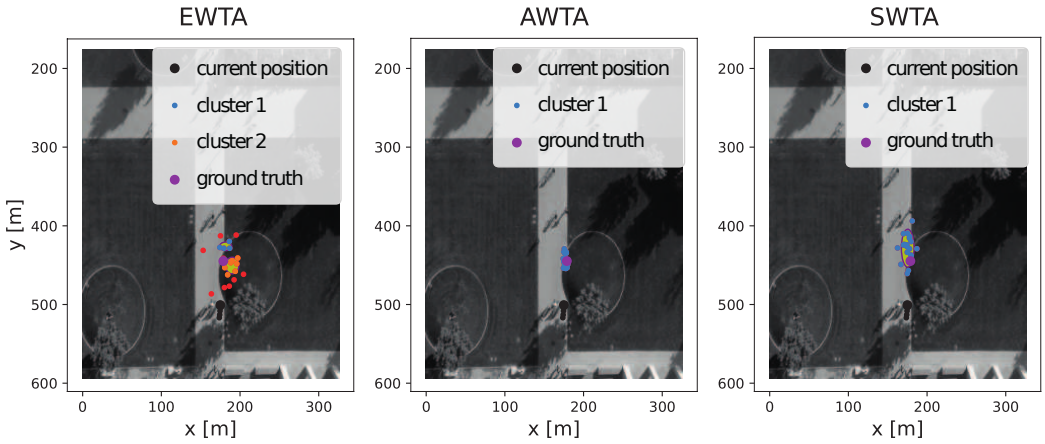

Fig. 7. These figures show the comparison among different WTA losses with CGF on the SDD test data. It is evident that the predicted hypotheses from the EWTA case are more dispersive than the AWTA and SWTA cases. Therefore, it is hard to cluster the predicted hypotheses from the EWTA.

if the prediction time offset is 10, the last channel of the input is a frame with 10s. The effect is shown in Fig. 6.

VI. Conclusions

In this work, a new method for multimodal motion prediction based on a sampling and clustering approach is proposed. We improve a multiple hypothesis estimation WTA loss, propose the AWTA and SWTA losses, and apply the DBSCAN clustering method with Gaussian fitting to generate multimodal predictions. The importance of making parameterized and geometric predictions is addressed so they can be utilized by model predictive controllers of mobile robots for collision avoidance. The evaluation on the simulated and real-world datasets shows that the proposed approach results in improved future position predictions compared to previously published loss functions.

Currently, a light neural network architecture is used. More experiments on the generalization of the proposed method should be made with more complicated networks. Future work focuses on improving the stability of loss functions so that it could tolerate large randomness and outliers.

References

[1] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, N. SINGH, and J. Schneider, "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[2] 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 *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[3] B. Zhou, X. Tang, and X. Wang, "Learning collective crowd behaviors with dynamic pedestrian-agents," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 50–68, 2015.

[4] S. Schaefer, K. Leung, B. Ivanovic, and M. Pavone, "Leveraging neural network gradients within trajectory optimization for proactive human-robot interactions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[5] X. Zhang, M. Scholz, S. Reitelshöfer, and J. Franke, "An autonomous robotic system for intralogistics assisted by distributed smart camera network for navigation," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2018.

[6] Z. Zhang, E. Dean, Y. Karayiannidis, and K. Åkesson, "Motion prediction based on multiple futures for dynamic obstacle avoidance of mobile robots," in *IEEE International Conference on Automation Science and Engineering (CASE)*, 2021.

[7] A. D. Kiureghian and O. Ditlevsen, "Aleatory or epistemic? does it matter?" *Structural Safety*, vol. 31, no. 2, pp. 105–112, 2009.

[8] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *The Second International Conference on Knowledge Discovery and Data Mining (KDD)*, vol. 96, no. 34, 1996.

[9] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, "Human motion trajectory prediction: a survey," *The International Journal of Robotics Research*, vol. 39, no. 8, pp. 895–935, 2020.

[10] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[11] J. Van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008.

[12] M. K. C. Tay and C. Laugier, "Modelling smooth paths using gaussian processes," in *Field and Service Robotics*. Springer, 2008.

[13] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[14] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[15] H. Xue, D. Q. Huynh, and M. Reynolds, "SS-LSTM: A hierarchical LSTM model for pedestrian trajectory prediction," in *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2018.

[16] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[17] J. Liang, L. Jiang, K. Murphy, T. Yu, and A. Hauptmann, "The garden of forking paths: Towards multi-future trajectory prediction," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[18] A. Guzman-Rivera, D. Batra, and P. Kohli, "Multiple choice learning: Learning to produce multiple structured outputs," *Advances in neural information processing systems*, vol. 25, 2012.

[19] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager, "Learning in an uncertain world: Representing ambiguity through multiple hypotheses," in *IEEE International Conference on Computer Vision (ICCV)*, 2017.

[20] O. Makansi, E. Ilg, O. Cicek, and T. Brox, "Overcoming limitations of mixture density networks: A sampling and fitting framework for multimodal future prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE conference on computer vision and pattern recognition (CVPR)*, 2016.

[22] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European Conference on Computer Vision (ECCV)*, 2016.

[23] C. M. Bishop, "Mixture density networks," *Technical report, Neural Computing Research Group, Aston University*, 1994.