

Uncertainty Quantification of Collaborative Detection for Self-Driving

Sanbao Su Yiming Li Sihong He Songyang Han Chen Feng Caiwen Ding Fei Miao

Abstract—Sharing information between connected and autonomous vehicles (CAVs) fundamentally improves the performance of collaborative object detection for self-driving. However, CAVs still have uncertainties on object detection due to practical challenges, which will affect the later modules in self-driving such as planning and control. Hence, uncertainty quantification is crucial for safety-critical systems such as CAVs. Our work is the first to estimate the uncertainty of collaborative object detection. We propose a novel uncertainty quantification method, called Double-M Quantification, which tailors a moving block bootstrap (MBB) algorithm with direct modeling of the multivariate Gaussian distribution of each corner of the bounding box. Our method captures both the epistemic uncertainty and aleatoric uncertainty with one inference pass based on the offline Double-M training process. And it can be used with different collaborative object detectors. Through experiments on the comprehensive collaborative perception dataset, we show that our Double-M method achieves more than $4\times$ improvement on uncertainty score and more than 3% accuracy improvement, compared with the state-of-the-art uncertainty quantification methods. Our code is public on <https://coperception.github.io/double-m-quantification/>.

I. INTRODUCTION

Multi-agent collaborative object detection has been proposed to leverage the viewpoints of other agents to improve the detection accuracy compared with the individual viewpoint [1]. Recent research has shown the effectiveness of early, late, and intermediate fusion of collaborative detection, which respectively transmits raw data, output bounding boxes, and intermediate features [2], [3], [4], [5], [6], and the improved collaborative object detection results will benefit the self-driving decisions of connected and autonomous vehicles (CAVs) [7]. However, CAVs may still have uncertainties on object detection due to out-of-distribution objects, sensor measurement noise, or poor weather [8], [9], [10]. Even a slightly false detection can lead the autonomous vehicle’s driving policy to a completely different action [11], [12]. For example, miss-detected paint on the road surface can confuse the lane-following policy and cause potential accidents [13]. Therefore, it is crucial to quantify the uncertainty of object detection for safety-critical systems such as CAVs.

Various uncertainty quantification methods have been proposed for object detection [10], [14], [15], [16]. Uncertainty resources can be decomposed into aleatoric and epistemic uncertainty [17], [18]. Direct modeling methods [19], [20],

This work was supported by NSF 1932250, 1952096, 2047354, and 2121391 grants. Sanbao Su, Sihong He, Songyang Han, Caiwen Ding, and Fei Miao are with the Department of Computer Science and Engineering, University of Connecticut, Storrs Mansfield, CT, USA 06268. Email: {sanbao.su, sihong.he, songyang.han, caiwen.ding, fei.miao}@uconn.edu. Yiming Li and Chen Feng are with Tandon School of Engineering, New York University, Brooklyn, NY, USA 11201. Email: {yimingli, cfeng}@nyu.edu.

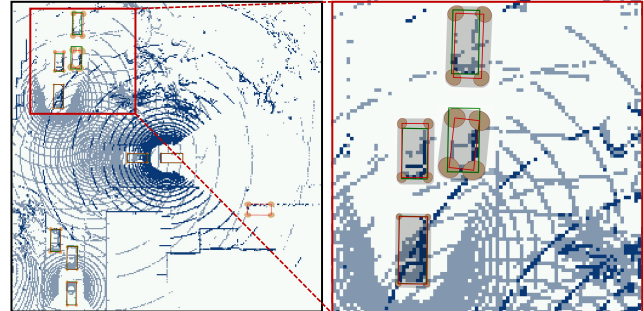


Fig. 1. **Left** figure shows detection results of intermediate collaboration in bird’s eye view (BEV), and **right** figure zooms on a specific part to show the robust range of two detections. Red boxes are predictions, and green boxes are ground truth. The orange ellipse denotes the covariance of each corner. The shadow convex hull shows the uncertainty set of the detected object. The shadow convex hull covers the green bounding box in most cases, which helps the later modules in self-driving tasks, such as trajectory prediction with uncertainty propagation [25] and robust planning and control [26], [27]. With our Double-M Quantification method, detected objects with low accuracy tend to have large uncertainties.

[21] focus on the aleatoric uncertainty (or data uncertainty) which represents inherent measurement noises from the sensor. Monte-Carlo dropout [22], [23] and deep ensemble [24] methods focus on the epistemic uncertainty (or model uncertainty) which reflects the degree of uncertainty that a model describes an observed dataset with its parameters. However, none of the above methods have investigated the uncertainty quantification of collaborative object detection.

In this paper, we propose a novel uncertainty quantification method for collaborative object detection, called Double-M Quantification (direct-Modeling Moving-block bootstrap Quantification), that only needs one inference pass to capture both the epistemic and aleatoric uncertainties. The constructed uncertainty set for each detected object by our method helps the later modules in self-driving tasks, such as trajectory prediction with uncertainty propagation [25] and robust planning and control [26], [27]. From Fig. 1, we can see that with our uncertainty quantification method, detected objects with low accuracy tend to have large uncertainties, and the constructed uncertainty set covers the ground-truth bounding box in most cases. Compared with the state-of-the-arts [20], [21], our Double-M Quantification method achieves up to $4\times$ improvement on uncertainty score and up to 3.04% accuracy improvement on the comprehensive collaborative perception dataset, V2X-SIM [1].

The main contributions of this work are as follows:

- 1) To the best of our knowledge, our proposed Double-M Quantification is the first attempt to estimate the uncertainty of collaborative object detection. Our method tailors a moving block bootstrap algorithm to estimate

both the epistemic and aleatoric uncertainties in one inference pass.

- 2) We design a novel representation format of the bounding box uncertainty in the direct modeling component to estimate the aleatoric uncertainty. We consider each corner of the bounding box as one independent multi-variant Gaussian distribution and the covariance matrix for each corner is estimated from one output header, while the existing literature mainly assumes a univariate Gaussian distribution for each dimension of each corner or a high dimensional Gaussian distribution for all corners.
- 3) We validate the advantages of the proposed methodology based on V2X-SIM [1] and show that our Double-M Quantification method reduces the uncertainty and improves the accuracy. The results also validate that sharing intermediate feature information between CAVs is beneficial for the system in both improving accuracy and reducing uncertainty.

II. RELATED WORK

a) Collaborative Object Detection: Collaborative object detection, in which multiple agents collectively perceive a scene via communication, is able to address several dilemmas in individual object detection [1], [28]. Compared to individual detection, multi-agent collaboration introduces more viewpoints to solve the long-range data sparsity and severe occlusions. The pioneer collaborative detectors employ early collaboration which shares raw data [3] or late collaboration which shares output bounding boxes [4]. To further improve the performance-bandwidth trade-off, recent research proposes intermediate collaboration which shares intermediate feature representations from a neural network. Various intermediate collaboration strategies have been developed such as neural message passing [29], knowledge distillation [2], and attention [30], [31]. However, existing works only focus on improving the performance of collaborative detection, no existing work investigates uncertainty quantification of collaborative object detection.

b) Uncertainty Quantification on Object Detection: Different types of uncertainty quantification methods for object detection have been proposed. For epistemic uncertainty, the Monte-Carlo dropout method utilizes the dropout-based neural network training to perform approximated inference in Bayesian neural networks [23]. The deep ensembles method estimates probability distribution by an ensemble of networks with the same architecture and different parameters [24], [32], [33]. Both methods require multiple runs of inference, which makes them infeasible for real-time critical tasks with high computational costs such as collaborative objection detection. Moreover, they do not consider time series properties in the dataset, which is one important characteristic of the autonomous driving dataset. In contrast, our uncertainty quantification method overcomes these problems by tailoring a moving block bootstrap [34] (MBB, an effective algorithm for time series analysis) algorithm and quantifies the uncertainty of collaborative object detection in one inference pass.

The direct modeling (DM) method is designed to estimate aleatoric uncertainty. The main steps of DM are [10]: a) select one object detector; b) set a certainty probability distribution on outputs of the detector and design the corresponding loss function; c) add extra regression layers to predict the covariance; d) train the modified detector. The work [20] proposes the DM method for image object detection, which assumes that the distribution of each bounding box variable is a single-variate Gaussian distribution and introduces one additional layer to estimate the variance of the bounding box. [21] proposes the DM method with a high dimensional multivariate Gaussian distribution. DM methods for point cloud object detection [19], [35] have been proposed. Methods with both DM and MC dropout to estimate aleatoric and epistemic uncertainties in object detection have also been investigated [18], [36], [37].

All the above works only focus on individual object detection. How to quantify the aleatoric and epistemic uncertainties in collaborative object detection remains challenging. In our work, we tailor an MBB-based algorithm process to estimate both aleatoric and epistemic uncertainties of collaborative object detection, with an independent multivariate Gaussian distribution assumption for each corner of the bounding box to represent the uncertainty.

III. UNCERTAINTY QUANTIFICATION APPROACH

In this section, we first define the problem of uncertainty quantification for collaborative object detection. Then we describe the overview structure of our novel Double-M Quantification (direct-Modeling Moving-block bootstrap Quantification) method as in Fig. 2, followed by the detailed algorithm process. Finally, we define our loss function of the neural network model. One major novelty is the first to tailor a moving block bootstrapping [34] (MBB) algorithm to address the uncertainty quantification challenge of collaborative object detection, and estimate the epistemic and aleatoric uncertainty with one inference pass in addition to the offline training process. The algorithm does not rely on a specific neural network model or structure and can be used with different collaborative object detectors such as DiscoNet [1]. The corresponding loss function considers both the prediction accuracy and covariance as metrics.

A. Problem Description

We use X to represent the point cloud data. For the deterministic collaborative object detection, the agent $\alpha \in \{1, \dots, N\}$ creates a local binary Bird's-Eye-View (BEV) map $\mathcal{M} \in \{0, 1\}^{W_m \times L_m \times H_m}$ from X , where N is the number of agents, and W_m , L_m , and H_m indicate the width, length, and height of the local BEV map respectively [38]. As shown in Fig. 2 (b), we consider homogeneous collaborative object detection, in which multiple agents share the same detector f_θ which consists of an encoder denoted by \mathcal{E} , and an aggregator as well as a decoder that is collectively represented by \mathcal{D} , where θ denotes the parameters of f . \mathcal{E} compresses the BEV map into an intermediate feature map $\mathbf{F}_\alpha \in \mathbb{R}^{\frac{W_m}{K_m} \times \frac{L_m}{K_m} \times F_m} = \mathcal{E}(\mathcal{M})$, where K_m

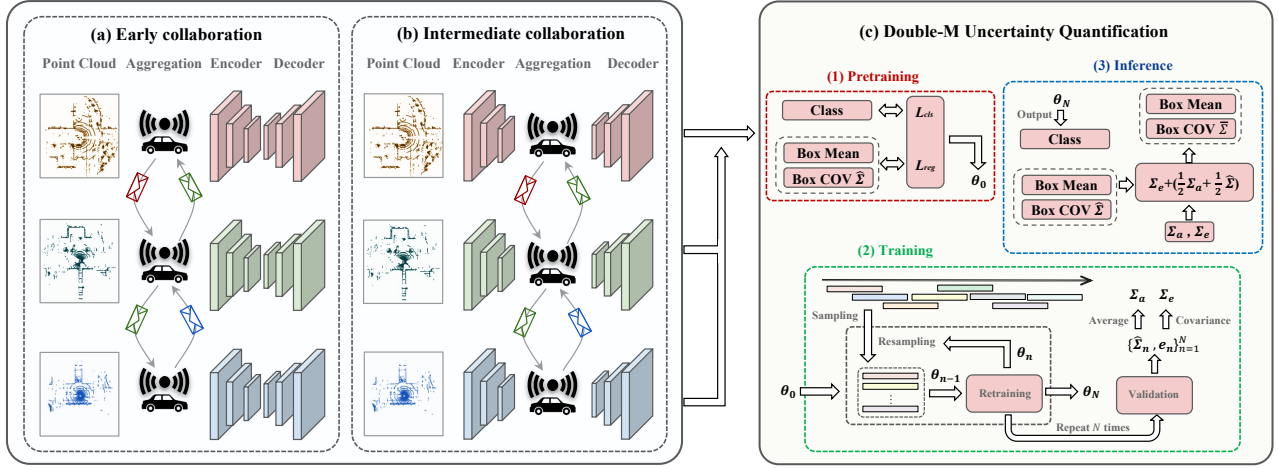


Fig. 2. Overview of our Double-M Quantification method on collaborative object detection. (a) **Early collaboration** shares raw point cloud with other agents, and (b) **intermediate collaboration** shares intermediate feature representations with other agents. (c) **Double-M Quantification method** estimates the multi-variate Gaussian distribution of each corner. Our Double-M Quantification method can be used on different collaborative object detection. During the training stage, Double-M Quantification tailors a moving block bootstrapping algorithm to get the final model parameter, Σ_a as the average aleatoric uncertainty of the validation dataset and Σ_e as the covariance of all residual vectors for epistemic uncertainty. During the inference stage, combine Σ_a , Σ_e and the predicted covariance matrix $\hat{\Sigma}$ from the object detector to compute the covariance matrix $\hat{\Sigma} = \Sigma_e + (\frac{1}{2}\Sigma_a + \frac{1}{2}\hat{\Sigma})$ of the distribution.

is the spatial downsampling scale, and F_m is the feature dimension. Afterwards, F_α is sent to other agents, and the agent α will use \mathcal{D} to generate a set of bounding boxes $\hat{Y} = \mathcal{D}(F_\alpha, \{F_\beta\}_{\beta \neq \alpha})$. We also consider early collaboration which shares raw point cloud as shown in Fig. 2 (a), we omit its notations for simplicity because the only thing that sets it apart from the intermediate collaboration is the timing of sharing information.

In each point cloud data X , there are J objects. For each object $j \in \{1, \dots, J\}$, we propose to predict I corners of the bounding box. Each corner $i \in \{1, \dots, I\}$ is represented by a D -dimensional vector in the BEV map. The set of ground-truth bounding boxes Y is represented as $Y = \{c_j, \{y_{ij}\}_{i=1}^I\}_{j=1}^J$ where c is the classification label and $y_{ij} \in \mathbb{R}^D, \forall (i, j)$. The set of predicted bounding box \hat{Y} is represented as $\hat{Y} = \{\hat{p}_j, \{\hat{y}_{ij}, \hat{\Sigma}_{ij}\}_{i=1}^I\}_{j=1}^J$, where \hat{p} is the predicted classification probability, and each corner of the bounding box is modeled as a multivariate Gaussian distribution with $\hat{y}_{ij} \in \mathbb{R}^D$ as the predicted mean and $\hat{\Sigma}_{ij}$ as the $D \times D$ covariance matrix coordinate of corner i of object j . Here we assume the probability distribution of each corner of the bounding box is independent. During training, the neural network parameters of the encoder \mathcal{E} , aggregator and decoder \mathcal{D} are jointly learned by minimizing the detection loss $\mathcal{L}_{det}(Y, \hat{Y})$ that includes a classification loss and a regression loss that considers both prediction accuracy and uncertainty.

B. Solution Overview

We design a novel uncertainty quantification method called direct-Modeling Moving-block bootstrap Quantification (Double-M Quantification) to estimate epistemic and aleatoric uncertainties by tailoring an MBB algorithm with the DM method. The overview of Double-M Quantification on collaborative object detection is shown in Fig. 2. During the training stage, we train the object detector on resampled

moving blocks. After N bootstraps, we get the object detector $f_{\hat{\theta}}$ where $\hat{\theta}$ is the final model parameter, compute Σ_a as the average aleatoric uncertainty of the validation dataset, and compute Σ_e as the covariance of all residual vectors for epistemic uncertainty. During the inference stage, with input point cloud X , we combine Σ_a , Σ_e and the predicted covariance matrix $\hat{\Sigma}_{ij}$ from $f_{\hat{\theta}}(X)$ to compute the covariance matrix $\hat{\Sigma}_{ij} = \Sigma_e + (\frac{1}{2}\Sigma_a + \frac{1}{2}\hat{\Sigma}_{ij})$ of the multivariate Gaussian distribution. In the following subsection III-C, we propose our novel design, the main contribution of this work, the Double-M Quantification algorithm. Then, in subsection III-D, we introduce the loss function for our method.

C. Double-M Quantification

Monte-Carlo dropout [23] and deep ensembles [24] have been proposed to estimate the epistemic uncertainty. However, none of them consider the time-series features in the dataset while the temporal features are important for CAVs. We design a novel uncertainty quantification method called Double-M Quantification to estimate epistemic and aleatoric uncertainties when considering the temporal features in the dataset. In particular, our design tailors a moving block bootstrapping [34] process on time-series data that captures the autocorrelation within the data by sampling data from the constructed data blocks in the training process.

We show the training stage of Double-M Quantification method in Algorithm 1. We first initialize θ , the parameters of a collaborative object detector and pretrain the model using the training data set. Then we construct the constant-length time-series block set \mathbf{B} from the time-series training dataset \mathbf{D}_K which contains K frames. Notice that the block set \mathbf{B} keeps the temporal characteristic (see Line 2) by maintaining the order of frames within the same block. Then in every iteration, we retrain the model using a sampled dataset that contains M blocks sampled with replacement and uniform random probability from the block set \mathbf{B} (see Lines 4-5). The sampled dataset still contains about K frames since

Algorithm 1: Double-M Quantification - Training

Data: The training dataset $\mathbf{D}_K = \{(X_k, Y_k)\}_{k=1}^K$ with K frames, the validation dataset $\mathbf{V}_{K'} = \{(X_k, Y_k)\}_{k=1}^{K'}$ with K' frames, the block length l , number of bootstraps N , the loss function \mathcal{L}_{det} in Eq (3), the collaborative object detector f_θ

Result: $\theta, \Sigma_e, \Sigma_a$

- 1 $\theta_0 \leftarrow \operatorname{argmin} \mathcal{L}_{det}(\mathbf{D}_K, \theta)$
- 2 Construct the overlapping moving-block collection $\mathbf{B} = \{\mathcal{B}_b\}_{b=1}^{K-l+1}$ from the time-series training dataset \mathbf{D}_K with each block $\mathcal{B}_b = \{(X_b, Y_b), \dots, (X_{b+l-1}, Y_{b+l-1})\}, b = 1, 2, \dots, K-l+1$.
- 3 **for** n from 1 to N **do**
- 4 Sample $M = \lfloor K/l \rfloor$ blocks with replacement from \mathbf{B} and get a sampled dataset $\mathbf{S}_n = \{\mathcal{B}_{b_m}\}_{m=1}^M$ where b_m are iid uniform random variables on $\{1, 2, \dots, K-l+1\}$.
- 5 Update $\theta_n \leftarrow \operatorname{argmin} \mathcal{L}_{det}(\mathbf{S}_n, \theta_{n-1})$
- 6 $f_{\theta_n}(\mathbf{V}_{K'}) = \{ \{ \hat{p}_{jk}, \{ \hat{y}_{ijk}, \hat{\Sigma}_{ijk} \}_{i=1}^I \}_{j=1}^J \}_{k=1}^{K'}$
- 7 Compute the residual vector $e_{ijk} = y_{ijk} - \hat{y}_{ijk} \forall i \in [1, I], j \in [1, J], k \in [1, K']$
- 8 **end**
- 9 $\hat{\theta} \leftarrow \theta_N$
- 10 Compute Σ_a as the average aleatoric uncertainty of the validation dataset
- 11 Compute Σ_e as the covariance of all residual vectors for epistemic uncertainty

$M = \lfloor K/l \rfloor$. The floor function $\lfloor \cdot \rfloor$ is to make sure M is an integer. At the final step in each training iteration n , we test the retained model f_{θ_n} on the validation dataset $\mathbf{V}_{K'}$ (see Line 6) and compute the residual vector as the difference between the ground-truth vector y_{ijk} and the predicted mean vector $\hat{y}_{ijk}, \forall i \in [1, I], j \in [1, J], k \in [1, K']$ (see Line 7). After N iterations, we get the final model parameters θ_N as $\hat{\theta}$ to predict the covariance by model $f_{\hat{\theta}}$. Other than the final trained model, we estimate both the aleatoric and epistemic uncertainties by using the residuals and predicted covariance matrices of the validation dataset. We first estimate the aleatoric uncertainty by computing Σ_a , the mean of all predicted covariance matrices. To estimate the epistemic uncertainty, we then compute the covariance matrix of all residual vectors, which is denoted by Σ_e . On the one hand, our Double-M Quantification method provides the bagging aleatoric uncertainty estimates through the aggregation over multiple models from the N iterations on the validation dataset. On the other hand, it approximates the distribution of errors from the residuals so that we can quantify the epistemic uncertainty.

The Inference stage of our Double-M Quantification method is shown in Algorithm 2. For the i th corner of the j th bounding box, we use \hat{y}_{ij} as the mean of the multivariate Gaussian distribution. we estimate the covariance matrix $\bar{\Sigma}_{ij}$ by using (i) the predicted covariance matrix $\hat{\Sigma}_{ij}$ from the

Algorithm 2: Double-M Quantification - Inference

Data: Σ_e, Σ_a , input point cloud X , the trained collaborative object detector $f_{\hat{\theta}}$

Result: \bar{Y}

- 1 $f_{\hat{\theta}}(X) = \bar{Y} = \{ \{ \hat{p}_j, \{ \hat{y}_{ij}, \hat{\Sigma}_{ij} \}_{i=1}^I \}_{j=1}^J \}$
- 2 **for** j from 1 to J **do**
- 3 **for** i from 1 to I **do**
- 4 $\bar{\Sigma}_{ij} = \Sigma_e + (\frac{1}{2}\Sigma_a + \frac{1}{2}\hat{\Sigma}_{ij})$
- 5 **end**
- 6 **end**
- 7 $\bar{Y} = \{ \{ \hat{p}_j, \{ \hat{y}_{ij}, \bar{\Sigma}_{ij} \}_{i=1}^I \}_{j=1}^J \}$

extra regression header and (ii) the estimated aleatoric and epistemic uncertainties Σ_e, Σ_a obtained from the training stage. $\bar{\Sigma}_{ij}$ is calculated as the following (see Lines 2–6):

$$\bar{\Sigma}_{ij} = \Sigma_e + \left(\frac{1}{2}\Sigma_a + \frac{1}{2}\hat{\Sigma}_{ij} \right). \quad (1)$$

D. Loss Function

In our Double-M Quantification method, we define the regression loss function of the object detector as the KL-Divergence between the predicted distribution and the ground-truth distribution.

Here we assume all corners are independent, and the distribution of each corner is a multivariate Gaussian distribution:

$$P_\theta(\bar{y}_i | \hat{y}_i, \hat{\Sigma}_i) = \frac{1}{\sqrt{2\pi |\hat{\Sigma}_i|}} \exp \left(-\frac{(\bar{y}_i - \hat{y}_i)^T \hat{\Sigma}_i^{-1} (\bar{y}_i - \hat{y}_i)}{2} \right),$$

where \bar{y}_i is one possible vector for the i -th corner, $\hat{\Sigma}_i$ is a symmetric positive definite $D \times D$ covariance matrix predicted for the i -th corner. In the implementation, we utilize the Cholesky decomposition [39] to calculate a symmetric positive definite covariance matrix. We compare this distribution with other distributions in Section IV and demonstrate our selected distribution is the best.

We assume the distribution of each corner for the ground-truth bounding box as a Dirac delta function [20]:

$$P_G(\bar{y}_i | y_i) = \delta(\bar{y}_i - y_i).$$

Then, we define the regression loss function for the i -th corner as the Kullback–Leibler (KL) divergence between $P_\theta(\bar{y}_i | \hat{y}_i, \hat{\Sigma}_i)$ and $P_G(\bar{y}_i | y_i)$ [40]:

$$\begin{aligned} \mathcal{L}_{KL}^i(y_i, \hat{y}_i, \hat{\Sigma}_i) &= \frac{1}{2} (y_i - \hat{y}_i)^T \hat{\Sigma}_i^{-1} (y_i - \hat{y}_i) + \\ &\quad \frac{1}{2} \log |\hat{\Sigma}_i| + \frac{\log(2\pi)}{2} - H(P_G(\bar{y}_i)), \end{aligned} \quad (2)$$

where $H(P_G(\bar{y}_i))$ is the entropy of $P_G(\bar{y}_i)$. The last two terms $\frac{\log(2\pi)}{2}$ and $H(P_G(\bar{y}_i))$ could be ignored in the loss function, for they are independent of the model parameters θ . The first term encourages increasing the covariance of the Gaussian distribution as the predicted mean vector diverges from the ground-truth vector. The second regularization term penalizes high covariance to reduce uncertainty.

We add one extra regression header to predict all covariance matrix $\hat{\Sigma}_i$ ($i \in 1, \dots, I$) with a similar structure as the regression header for \hat{y}_i , based on the origin collaborative object detector.

With a given training dataset, the collaborative object detector f_θ is trained to predict \hat{Y} , the classification probability and the regression distribution of all J objects. The classification loss is $\mathcal{L}_{cls}(c, \hat{p})$ as [1]. The loss function of the object detector is:

$$\mathcal{L}_{det} = \sum_{j=1}^J (\mathcal{L}_{cls}(c_j, \hat{p}_j) + \sum_{i=1}^I \mathcal{L}_{KL}^i(y_{ij}, \hat{y}_{ij}, \hat{\Sigma}_{ij})). \quad (3)$$

IV. EXPERIMENT

To demonstrate our uncertainty quantification method for collaborative detection, we evaluated it on the V2X-Sim dataset [1] which contains 80 scenes for training, 10 scenes for validation, and 10 scenes for testing. It is generated by CARLA simulation [41]. Each scene contains a 20-second traffic flow at a certain intersection with a 5Hz record frequency, which means each scene contains 100 time-series frames. In each scene, 2-5 vehicles are selected as the connected vehicles and 3D point clouds are collected from LiDAR sensors on them.

For object detection, we consider Lower-bound, DiscoNet, and Upper-bound for benchmark as follows:

- 1) Lower-bound (LB) [1]: The individual object detector without collaboration which only uses the point cloud data from one individual LiDAR.
- 2) DiscoNet (DN) [2]: The intermediate collaborative object detector which utilizes a directed graph with matrix-valued edge weight to adaptively aggregate features of different agents by repressing noisy spatial regions while enhancing informative regions. It has shown a good performance-bandwidth trade-off by sharing a compact and context-aware scene representation.
- 3) Upper-bound (UB) [1]: The early collaborative object detector uses raw point cloud data from all connected vehicles, as shown in Fig. 2(a). It usually has excellent performance with lossless information, yet consumes high communication bandwidth.

LB is the traditional individual object detector. DN and UB are collaborative object detectors. The basic implementation of all three benchmarks is from the public code of [1]. It uses FaFNet [42], a classic anchor-based model containing a convolutional encoder, a convolutional decoder, and two output headers for classification and regression, as the backbone of all detectors.

We compare three uncertainty quantification methods, which are Direct Modeling (DM), Moving Block Bootstrap (MBB), and Double-M Quantification on accuracy and uncertainty. Compared with Double-M Quantification, the MBB method does not have the predicted covariance matrix during training and inference. For DM and Double-M Quantification, we add one output header for the covariance matrix and the regression loss is the Kullback–Leibler (KL) loss in Eq (2). For MBB, the regression loss is the smooth L_1 loss. For all, the classification loss is the focal cross-entropy loss [43].

TABLE I

DETECTION PERFORMANCE OF DIFFERENT UQ METHODS ON LB, DN, AND UB. OUR DOUBLE-M QUANTIFICATION METHOD IMPROVES UP TO 3.04% AVERAGE PRECISION.

UQ Method	AP @IoU=0.5 \uparrow			AP @IoU=0.7 \uparrow		
	LB	DN	UB	LB	DN	UB
None	46.50	66.60	69.76	41.29	60.84	64.78
DM [10]	43.31	66.36	69.28	39.24	60.91	65.25
MBB [34]	46.64	67.10	70.41	40.77	60.98	64.48
Double-M (Ours)	43.83	67.20	70.44	39.74	62.69	66.37

TABLE II

NLL COMPARISON OF DIFFERENT UQ METHODS ON LB, DN, AND UB. OUR DOUBLE-M QUANTIFICATION METHOD ACHIEVES UP TO 4 \times IMPROVEMENT ON NLL.

UQ Method	NLL @IoU=0.5 \downarrow			NLL @IoU=0.7 \downarrow		
	LB	DN	UB	LB	DN	UB
DM [10]	13.222	10.015	14.721	9.009	7.896	13.001
MBB [34]	28.130	13.794	22.958	19.996	9.710	18.077
Double-M (Ours)	6.871	5.084	7.974	4.889	3.851	6.696

A. Accuracy Evaluation

We use Average Precision (AP) at Intersection-over-Union (IoU) thresholds of 0.5 and 0.7 as the accuracy measurement, which is widely used in object detection [10]. Table I shows the AP results of different uncertainty quantification (UQ) methods on the test dataset, where “None” means the object detector is a deterministic one without any UQ method. Our Double-M Quantification achieves better accuracy than others, especially for collaborative object detection. Compared with deterministic detectors without the UQ method, it increases up to 3.04% AP, which means our proposed Double-M Quantification method improves the accuracy of collaborative object detection. Meanwhile, DiscoNet and Upper-bound achieve much higher accuracy than Lower-bound for sharing information between CAVs.

B. Uncertainty Evaluation

We use Negative Log Likelihood (NLL) at IoU thresholds of 0.5 and 0.7 as the uncertainty measurement [10], [15]. NLL is a widely used uncertainty score to measure the quality of predicted probability distribution on a test dataset [19], [20], [24], [25]. For the test dataset with K frames, it is computed as:

$$NLL = -\frac{1}{I \times J \times K} \sum_{k=1}^K \sum_{j=1}^J \sum_{i=1}^I \log P(y_{ijk} | \hat{y}_{ijk}, \hat{\Sigma}_{ijk}),$$

where \hat{y}_{ijk} , $\hat{\Sigma}_{ijk}$ are the mean and covariance of the multivariate Gaussian distribution. For MBB, $\hat{\Sigma}_{ijk}$ is Σ_e . For DM, it is the predicted covariance $\hat{\Sigma}_{ijk}$ from the additional regression header. For Double-M Quantification, it is computed with Eq (1).

Table II shows the NLL results of different uncertainty quantification methods on the test dataset. From Table II, we can see our proposed Double-M Quantification always achieves much smaller NLL than other uncertainty quantification methods, with up to 4 \times improvement, which means our Double-M Quantification method performs best on uncertainty quantification. Fig. 3 shows the qualitative results of our Double-M Quantification method on different scenes. For different types of object detectors, DiscoNet always achieves the smallest NLL. From Table II and Fig. 3,

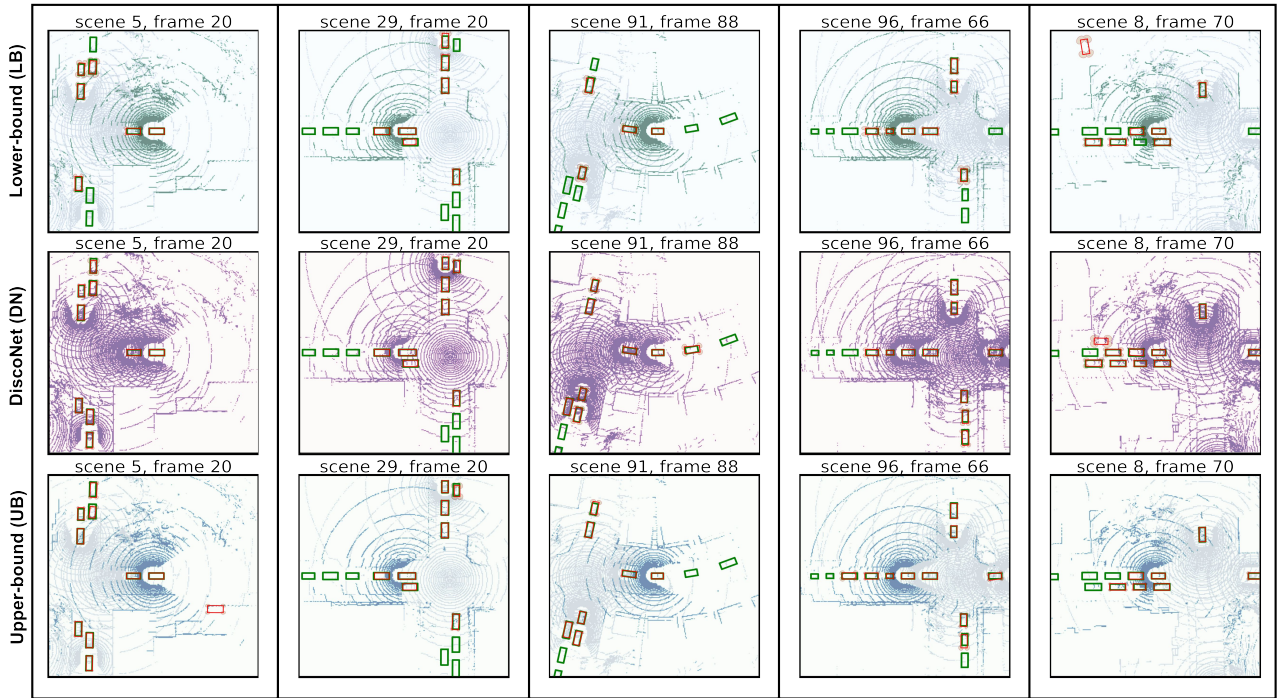


Fig. 3. Visualization of our Double-M Quantification results on different scenes of V2X-Sim [1]. The results of LB, DN, and UB are respectively shown in the first, second, and third rows. Red boxes are predictions, and green boxes are ground truth. The orange ellipse denotes the covariance of each corner. We can see our Double-M Quantification predicts large orange ellipses when the difference between the red bounding box and the corresponding green bounding box is huge, which means our method is efficient. For example, in the subfigure of DiscoNet on scene 29 frame 20, for the top-right object O_1 , the difference between the red and green bounding boxes is huge so that Double-M Quantification predicts large orange ellipses. And for the three objects on the left side of O_1 , the difference between their red and green bounding boxes is little so that Double-M Quantification predicts small orange ellipses.

we can see sharing intermediate feature information between CAVs could improve the uncertainty of object detection.

C. Ablation Study on Uncertainty Distribution

We compare different probability distributions, which is the key step of the DM method and the Double-M Quantification method, on accuracy. In particular, we consider the following probability Gaussian distribution:

- 1) Independent Multivariate Gaussian (IMG): Our uncertainty representation of independent Gaussian distribution. All corners are independent, and the distribution of each corner is a multivariate Gaussian distribution.
- 2) Independent Single-variate Gaussian (ISG) [20]: Single-variate Gaussian distribution. All corners are independent, and all dimensions of one corner are also independent. We use a single-variate Gaussian distribution for each dimension of each corner.
- 3) Dependent Multivariate Gaussian (DMG) [21]: High dimensional Gaussian distribution. All corners are dependent, and the distribution of all corners is a multivariate Gaussian distribution.

Table III shows the AP results of DM method and Double-M Quantification method for the Upper-bound detector, under different probability distributions. From the table, we can see IMG achieves the best accuracy with up to 4.14% improvement, which means our uncertainty representation of independent Gaussian distribution for each corner outperforms single-variate Gaussian distribution and high

TABLE III

THE ACCURACY COMPARISON OF UPPER-BOUND UNDER DIFFERENT PROBABILITY DISTRIBUTIONS. OUR IMG DISTRIBUTION IMPROVES UP TO 4.14% AVERAGE PRECISION.

	AP @IoU=0.5 \uparrow		AP @IoU=0.7 \uparrow	
Distribution	DM	Double-M	DM	Double-M
IMG (Ours)	69.28	70.44	65.25	66.37
ISG [20]	68.47	68.95	64.68	65.30
DMG [21]	68.23	67.74	64.86	63.73

dimensional Gaussian distribution formats. The reason is that our design considers high dependence of all dimensions in one corner and low dependence of all corners.

V. CONCLUSION

This work proposes the first attempt to estimate the uncertainty of collaborative object detection. We propose one novel uncertainty quantification method, called Double-M Quantification, to predict both the epistemic and aleatoric uncertainty with one inference pass. The key novelties are the tailored moving block bootstrap training process, and the loss function design that estimates one independent multivariate Gaussian distribution for each corner of the bounding box. We validate our uncertainty quantification method on different collaborative object detectors. Experiments demonstrate that our method achieves better uncertainty estimation and accuracy. In the future, we will apply our method to more collaborative perception datasets, and enhance the performance of trajectory prediction with uncertainty quantification.

REFERENCES

- [1] Y. Li, D. Ma, Z. An, Z. Wang, Y. Zhong, S. Chen, and C. Feng, "V2x-sim: Multi-agent collaborative perception dataset and benchmark for autonomous driving," *IEEE Robotics and Automation Letters*, 2022.
- [2] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," *Advances in Neural Information Processing Systems*, pp. 29 541–29 552, 2021.
- [3] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, 2019, pp. 514–524.
- [4] E. Arnold, M. Dianati, R. de Temple, and S. Fallah, "Cooperative perception for 3d object detection in driving scenarios using infrastructure sensors," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [5] W. Chen, R. Xu, H. Xiang, L. Liu, and J. Ma, "Model-agnostic multi-agent perception framework," *arXiv preprint arXiv:2203.13168*, 2022.
- [6] Y. Li, J. Zhang, D. Ma, Y. Wang, and C. Feng, "Multi-robot scene completion: Towards task-agnostic collaborative perception," in *6th Annual Conference on Robot Learning*, 2022.
- [7] S. Han, H. Wang, S. Su, Y. Shi, and F. Miao, "Stable and efficient shapley value-based reward reallocation for multi-agent reinforcement learning of autonomous vehicles," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8765–8771.
- [8] Z. Liu, Y. Cai, H. Wang, L. Chen, H. Gao, Y. Jia, and Y. Li, "Robust target recognition and tracking of self-driving cars with radar and camera information fusion under severe weather conditions," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [9] D. Kothandaraman, R. Chandra, and D. Manocha, "Ss-sfda: Self-supervised source-free domain adaptation for road segmentation in hazardous environments," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3049–3059.
- [10] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [11] S. Huang, N. Papernot, I. Goodfellow, Y. Duan, and P. Abbeel, "Adversarial attacks on neural network policies," *The International Conference on Learning Representations*, 2017.
- [12] Y.-C. Lin, Z.-W. Hong, Y.-H. Liao, M.-L. Shih, M.-Y. Liu, and M. Sun, "Tactics of adversarial attack on deep reinforcement learning agents," *Proceedings of the International Joint Conference on Artificial Intelligence*, 2017.
- [13] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.
- [14] D. Hall, F. Dayoub, J. Skinner, H. Zhang, D. Miller, P. Corke, G. Carneiro, A. Angelova, and N. Sünderhauf, "Probabilistic object detection: Definition and evaluation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1031–1040.
- [15] A. Harakeh and S. L. Waslander, "Estimating and evaluating regression predictive uncertainty in deep object detectors," in *International Conference on Learning Representations*, 2021.
- [16] H.-k. Chiu, J. Li, R. Ambruş, and J. Bohg, "Probabilistic 3d multi-modal, multi-object tracking for autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 14 227–14 233.
- [17] E. Hüllermeier and W. Waegeman, "Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods," *Machine Learning*, vol. 110, no. 3, pp. 457–506, 2021.
- [18] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" *Advances in neural information processing systems*, vol. 30, 2017.
- [19] G. P. Meyer and N. Thakurdesai, "Learning an uncertainty-aware object detector for autonomous driving," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 521–10 527.
- [20] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2888–2897.
- [21] Y. He and J. Wang, "Deep mixture density network for probabilistic object detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 550–10 555.
- [22] Y. Gal, J. Hron, and A. Kendall, "Concrete dropout," 2017.
- [23] D. Miller, L. Nicholson, F. Dayoub, and N. Sünderhauf, "Dropout sampling for robust object detection in open-set conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3243–3249.
- [24] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *Advances in neural information processing systems*, vol. 30, 2017.
- [25] I. Boris, L. Yifeng, S. Shubham, C. Punarjay, and P. Marco, "Propagating state uncertainty through trajectory forecasting," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.
- [26] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
- [27] W. Xu, J. Pan, J. Wei, and J. M. Dolan, "Motion planning under uncertainty for on-road autonomous driving," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 2507–2512.
- [28] X. Cai, W. Jiang, R. Xu, W. Zhao, J. Ma, S. Liu, and Y. Li, "Analyzing infrastructure lidar placement with realistic lidar," *arXiv preprint arXiv:2211.15975*, 2022.
- [29] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 605–621.
- [30] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.
- [31] R. Xu, J. Li, X. Dong, H. Yu, and J. Ma, "Bridging the domain gap for multi-agent perception," *arXiv preprint arXiv:2210.08451*, 2022.
- [32] Z. Lyu, N. Gutierrez, A. Rajguru, and W. J. Beksí, "Probabilistic object detection via deep ensembles," in *European Conference on Computer Vision*. Springer, 2020, pp. 67–75.
- [33] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift," *Advances in neural information processing systems*, vol. 32, 2019.
- [34] S. N. Lahiri, "Theoretical comparisons of block bootstrap methods," *Annals of Statistics*, pp. 386–404, 1999.
- [35] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 677–12 686.
- [36] D. Feng, L. Rosenbaum, F. Timm, and K. Dietmayer, "Leveraging heteroscedastic aleatoric uncertainties for robust real-time lidar 3d object detection," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1280–1287.
- [37] D. Feng, L. Rosenbaum, and K. Dietmayer, "Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection," in *2018 21st international conference on intelligent transportation systems (ITSC)*. IEEE, 2018, pp. 3266–3273.
- [38] P. Wu, S. Chen, and D. N. Metaxas, "Motionnet: Joint perception and motion prediction for autonomous driving based on bird's eye view maps," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 385–11 395.
- [39] A. Kumar, T. K. Marks, W. Mou, Y. Wang, M. Jones, A. Cherian, T. Koike-Akino, X. Liu, and C. Feng, "Luvli face alignment: Estimating landmarks' location, uncertainty, and visibility likelihood," in *Proceedings of the IEEE/CVPR Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8236–8246.
- [40] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.
- [41] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.
- [42] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.
- [43] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.