

# BatchEnsemble-based Perception Uncertainty Quantification for Autonomous Vehicles

Zhengyi Li

*State Key Laboratory of Automotive  
Chassis Integration and Bionics  
Jilin University  
Changchun, China  
lizy21@mails.jlu.edu.cn*

Hongyu Hu

*State Key Laboratory of Automotive  
Chassis Integration and Bionics  
Jilin University  
Changchun, China  
huhongyu@jlu.edu.cn*

Yang Xing

*Faculty of Engineering and Applied Sciences  
Cranfield University  
MK43 0AL Bedford, U.K.  
Yang.X@cranfield.ac.uk*

Chen Lv

*School of Mechanical and Aerospace Engineering  
Nanyang Technological University  
Singapore  
lyuchen@ntu.edu.sg*

**Abstract**—Quantifying uncertainty significantly enhances the reliability of perception in autonomous vehicles and provides more comprehensive environmental information for downstream modules. However, most existing perception methods lack the capacity to effectively estimate the associated uncertainty. To address this gap, we propose a BatchEnsemble-based network for uncertainty quantification in 3D object detection using point cloud data. Specifically, a BatchEnsemble-based convolutional layer is designed to reduce the memory overhead associated with ensemble-based paradigms. Building upon this, a series of probabilistic object detection networks are constructed by directly modeling object attributes using multivariate Gaussian distributions, thereby enabling the parallel extraction of both object features and their associated variances. Subsequently, an uncertainty-aware fusion strategy is introduced to integrate and filter multiple detection results based on an uncertainty quantification metric—namely, the Uncertainty Index—thereby yielding more reliable and comprehensive outputs. The proposed method is validated on the KITTI dataset. Experimental results demonstrate its competitive accuracy performance and effectiveness across various scenarios, including objects of differing detection difficulty, identification of false-positive results, and under adverse conditions such as snowy weather and sensor degradation.

**Index Terms**—uncertainty quantification, 3D object detection, perception uncertainty, autonomous driving

## I. INTRODUCTION

Autonomous driving technology has witnessed substantial advancements in recent years [1], [2]. The environmental perception module, similar to a driver's vision, is responsible for interpreting the surrounding environment to inform downstream modules such as decision-making, planning, and control. Modern perception systems, predominantly driven by

deep neural networks, have demonstrated impressive performance across various benchmarks [3], [4]. Nevertheless, the performance achieved under controlled datasets often fails to generalize to the unpredictable and adverse conditions encountered in real-world deployments. As a result, existing perception models often yield uncertain or erroneous predictions under adverse conditions—such as snowy weather or sensor degradation—without conveying any warning of the underlying uncertainty [5]. These perception inaccuracies can generally be categorized into two types: existence bias, where the model detects non-existent objects, and attribute bias, where the model mispredicts object properties (e.g., size, velocity, or orientation). Ideally, the perception module should quantify the uncertainty associated with both types of bias, thereby enabling downstream modules to account for potential errors in their reasoning and, if necessary, alert the driver to take over control of the vehicle. However, most existing perception methods lack the capability to estimate the uncertainty of their outputs. Consequently, downstream modules are forced to act upon raw predictions—including false positives and those with high uncertainty. This limitation not only increases the computational and reasoning burden on subsequent modules but also undermines the overall robustness and safety of the autonomous driving system. Therefore, quantifying the uncertainty of the perception system and identifying the boundaries of model capability is of significant importance [6].

Recent research has increasingly focused on uncertainty quantification (UQ) as a means of providing calibrated confidence estimates alongside predictions. UQ distinguishes between two primary forms of uncertainty: epistemic uncertainty and aleatoric uncertainty. Epistemic uncertainty reflects the model's inherent limitations and arises due to factors such as insufficient training data, suboptimal training strategies, or constraints in model architecture. Two predominant ap-

This work was supported in part by the National Natural Science Foundation of China under Grant 52272417, in part by the Science and Technology Development Project of Jilin Province under Grant 20230301008ZD, and in part by the Foundation of State Key Laboratory of Automotive Simulation and Control under Grant 20210214. (*corresponding author: Hongyu Hu.*)

proaches for estimating epistemic uncertainty are Monte Carlo (MC) Dropout and deep ensembles. MC Dropout involves applying dropout during both training and inference, effectively simulating a distribution over model parameters. This approach is often viewed as a pseudo-ensemble method, as each forward pass samples a slightly different network due to the stochastic omission of neurons [7]. Deep ensembles, by contrast, consist of multiple models with identical architectures but independently trained parameters. Their aggregated outputs provide a robust estimate of both the final prediction and the associated epistemic uncertainty [8]. Aleatoric uncertainty, on the other hand, arises from irreducible noise in the input data, such as sensor degradation or adverse weather conditions. To quantify aleatoric uncertainty, many studies adopt a Bayesian framework, wherein a prior distribution is placed over the network parameters and updated via observed data to obtain a posterior distribution [9]. Additionally, several direct modeling approaches have been proposed in the object detection, which involve assuming a parametric distribution over the model's outputs and training the network to predict the parameters of this distribution directly [10]. However, direct modeling approaches often require non-trivial architectural modifications and complete retraining, which can negatively affect perception performance. Although ensemble-based methods can alleviate such performance degradation, they generally incur significant computational overhead, making them impractical for deployment.

To address these challenges, this paper proposes an uncertainty quantification method based on BatchEnsemble—a lightweight ensemble paradigm [11]. The proposed approach alleviates the memory overhead typically associated with uncertainty estimation, providing a more efficient and deployable solution. Specifically, we design a BatchEnsemble-based Convolutional Layer (BECL) to reduce the computational cost of convolutional layers in conventional ensemble networks. Based on this design, multiple probabilistic object detection models are constructed. Subsequently, we employ an uncertainty-aware fusion strategy that integrates the multiple sets of predictions generated by the ensemble members using a novel uncertainty metric—Uncertainty Index. Our method provides downstream modules with enhanced and uncertainty-aware perception information, ultimately improving the robustness and reliability of autonomous decision-making.

The main contributions of this paper can be summarized as follows:

- We propose a BatchEnsemble-based uncertainty quantification method for 3D object detection in point clouds. The method constructs probabilistic detection models based on multivariate Gaussian distributions and incorporates an uncertainty-aware fusion strategy alongside a dedicated uncertainty quantification metric to produce comprehensive perception results with associated confidence estimates.
- We propose a BatchEnsemble-based convolutional layer, which optimizes the number of parameters in ensemble networks and significantly reduces memory overhead.

- The proposed method has been validated on the KITTI dataset. Experimental results demonstrate competitive detection accuracy and effectiveness across diverse scenarios.

## II. RELATED WORKS

This study is closely related to 3D object detection in point clouds and uncertainty quantification. The mainstream methods about these topics are briefly introduced.

### A. 3D Object Detection in Point-Cloud

3D object detection in point clouds, which aims to accurately identify and localize specific objects within the vehicle's surrounding environment, is a fundamental perception task in autonomous driving. In recent years, deep learning-based methods have significantly outperformed traditional methods that rely on handcrafted features, achieving *state-of-the-art* performance across various benchmark datasets [3], [12]. PointNet pioneered the direct processing of raw point clouds, offering high efficiency and flexibility [13]. Building on this, VoxelNet introduced an end-to-end voxel-based framework that improved detection accuracy [14]. PointPillars further enhanced efficiency by encoding point clouds into pillar-based features and eliminating costly 3D convolutions, enabling fast and accurate detection suitable for real-time applications [15]. CenterPoint formulates detection as a center-based keypoint estimation problem, leveraging point cloud data to identify object centers and regress their properties with high accuracy and efficiency [16]. The Point Transformer leverages self-attention mechanisms to model both local and global dependencies among points, thereby facilitating more effective feature learning for 3D vision tasks [17]. PillarNeXt improves feature extraction and utilizes an efficient network architecture to achieve higher accuracy and speed in 3D object detection [18]. In contrast to the aforementioned approaches, this paper proposes a novel paradigm for uncertainty quantification within the environmental perception process.

### B. Uncertainty Quantification

Several studies on uncertainty quantification have been proposed with the widespread application of deep learning [5], [19], [20]. The Bayesian formalism remains the dominant approach, and various approximations have been proposed to mitigate its computational challenges in neural networks [21], [22]. Several methods requiring multiple inferences, such as MC-Dropout and deep ensembles, have proven effective in capturing epistemic uncertainty [7], [8], [23]. To mitigate the computational cost associated with ensemble methods, Packed-Ensembles utilizes group convolutions to parallelize ensemble models into a single shared backbone, enhancing both training and inference efficiency [24]. In addition, deterministic networks have been developed to model outputs as probability distributions rather than single-point estimates to capture aleatoric uncertainty. For instance, Gaussian-based bounding box regression loss has been proposed for object detection, allowing for simultaneous prediction of the

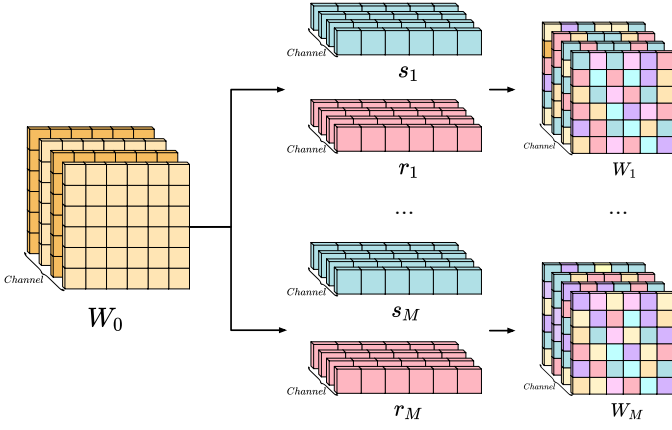


Fig. 1. Illustration of the BatchEnsemble-based convolutional operation. The shared kernel weights  $W_0$  are combined with multiple sets of ensemble-specific modulation vectors to generate diverse effective weights  $\{W_1, W_2, \dots, W_M\}$ , enabling efficient ensemble learning with minimal memory overhead.

bounding boxes and associated variances [25]. Furthermore, uncertainty from interactions has also been investigated. [26] and [27] introduce the concept of collaborative uncertainty, modeling interactive environment uncertainties using Gaussian and Laplace distributions in deterministic networks.

### III. METHOD

#### A. BatchEnsemble-based Convolution Layer

In this section, we propose BatchEnsemble-based Convolutional Layer (BECL), a novel convolutional layer based on the BatchEnsemble methodology. As shown in Fig. 1, let  $W_0$  denote the shared weight tensor of a standard convolutional layer. Assume that the input and output channel dimensions are  $c_{in}$  and  $c_{out}$ , respectively, and the kernel size is  $(k_1, k_2)$ . Therefore, the weight tensor is defined as  $W_0 \in \mathbb{R}^{c_{in} \times c_{out} \times k_1 \times k_2}$ .

In contrast to the standard convolutional layer, BECL optimizes the storage of convolutional kernels by representing each ensemble member's weight as a tuple of trainable vectors  $r \in \mathbb{R}^{c_{in} \times c_{out} \times k_1 \times 1}$  and  $s \in \mathbb{R}^{c_{in} \times c_{out} \times 1 \times k_2}$ , instead of allocating a full set of parameters per member. A new convolutional kernel for each ensemble member can then be generated by performing a Hadamard product between the shared base kernel and the corresponding  $r$  and  $s$  vectors, as follows:

$$W' = W_0 \odot rs \quad (1)$$

where  $W'$  denotes the weight of the BECL, and  $\odot$  represents the Hadamard product.

Compared to conventional convolution operations, the proposed method reduces the memory overhead of convolutional layers by a factor of  $1 - \left(\frac{1}{k_1} + \frac{1}{k_2}\right)$ . For example, when  $k_1 = k_2 = 3$ , the memory consumption can be reduced by approximately 33%. Compared with deep ensembles, the ensemble network constructed based on BECL only requires storing the feature modulation vectors  $r$  and  $s$ , significantly reducing memory usage across other layers.

#### B. BatchEnsemble-based Neural Network

We designed a BatchEnsemble-based network, which consists of a shared probabilistic objection detection network and  $M$  sub-networks. The shared model adopts a fully BECL network based on PointPillars as the backbone encoder, along with three fully convolutional decoders. First, the point cloud is converted into a sparse pseudo-image and encoded into a high-level representation of the environment. Subsequently, the classification decoder identifies the categories of objects such as pedestrians, cyclists, and vehicles. Simultaneously, the regression decoder determined the precise attributes of the 3D bounding boxes of the detected objects, which are essential for the accurate spatial awareness and autonomous vehicles. In addition to the two branches mentioned above, the variance-inference decoder estimates the variance associated with the attributes of the 3D bounding boxes. These variances allow for the estimation of the uncertainty associated. During both the training and inference stages, each sub-network only retains only the feature vectors of its convolutional kernels, denoted as  $R_m = \{r_1, r_2, \dots, r_L\}$  and  $S_m = \{s_1, s_2, \dots, s_L\}$ , where  $m \in [0, M]$  indicates the index of the sub-network and  $L$  indicates the number of BECL in each sub-network. The case  $m = 0$  corresponds to the shared base network, in which both  $r_0$  and  $s_0$  are tensors filled with ones, serving as identity scaling factors for the shared convolutional weights. The convolutional weights of each sub-network are generated using Equation. 1, ensuring that all sub-networks share the same architecture but differ in parameters. This enables the construction of an efficient deep ensemble framework with significantly reduced memory and computational overhead.

We use  $X$  to indicate the point cloud data as input. Each  $X$  contains  $N$  objects  $\mathcal{O} = \{o_1, o_2, \dots, o_N\}$ , where each object is characterized by a category  $c_n$  and bounding box  $\mathcal{B}_n = \{x_n, y_n, z_n, w_n, l_n, h_n, r_n\}$ , where the coordinates  $(x_n, y_n, z_n)$  indicate the location of the object  $o_n$ , and  $(w_n, l_n, h_n, r_n)$  describe the dimension and orientation of their bounding box. Therefore, the set of ground-truth  $Y$  in  $X$  can be described as  $Y = \{\{c_n, \mathcal{B}_n\}_{n=1}^N\}$ . For the  $m$ -th sub-network, the prediction results are denoted as  $\hat{Y}_m = \{\{\hat{c}_n, \hat{\mathcal{B}}_n, \hat{\Sigma}_n\}_{n=1}^{\hat{N}_m}\}$ , where  $\hat{N}_m$  indicates the number of predicted objects.  $\hat{c}_n$  denotes the category of the predicted object  $o_n$ , and  $\hat{\mathcal{B}}_n$  is the predicted bounding box consisting of  $K$  attributes, similar to  $\mathcal{B}_n$ .  $\hat{\Sigma}_n$  is a  $K \times K$  covariance matrix.

The transformation process for  $m$ -th sub-network can be described as follows:

$$\theta_m = f(R_m, S_m) \quad (2)$$

$$\mathbb{F}_m = f_{enc}(X; \theta_m) \quad (3)$$

$$\hat{c}_m = D_{cls}(\mathbb{F}_m; \theta_m) \quad (4)$$

$$\hat{\mathcal{B}}_m = D_{reg}(\mathbb{F}_m; \theta_m) \quad (5)$$

$$\hat{\Sigma}_m = D_{var}(\mathbb{F}_m; \theta_m) \quad (6)$$

where  $f(\cdot)$  indicates the transformation from parameter set  $\theta_0$  to  $\theta_m$ .  $f_{enc}(\cdot; \theta_m)$  indicates the point cloud encoding process over the parameter set  $\theta_m$ .  $\mathbb{F}_m$  indicates the higher-

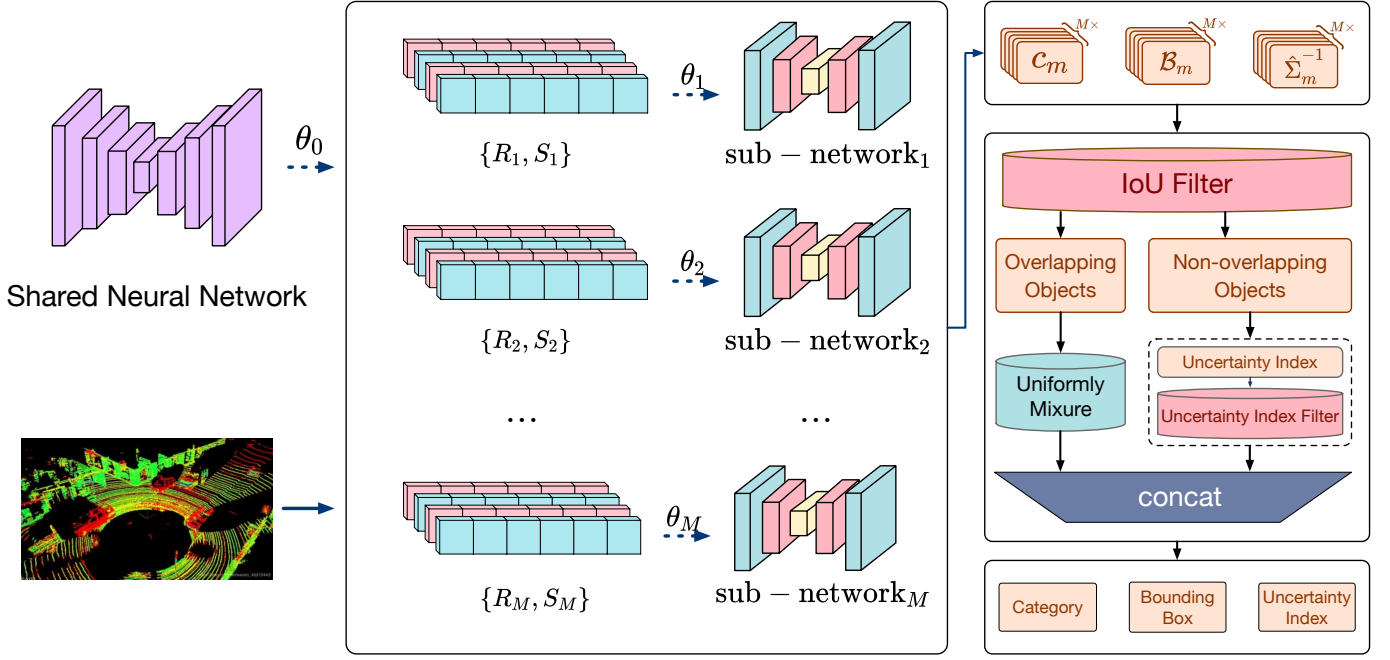


Fig. 2. Illustration of the proposed method. Dashed lines indicate the construction of model parameters, while solid lines represent the data flow within the model. Multiple probabilistic object detection sub-networks are constructed by combining shared weights  $\theta_0$  with sub-network-specific modulation vectors. The resulting sets of perception outputs are then passed to the uncertainty-aware fusion module, where the uncertainty index is computed and the final detection results along with their corresponding uncertainties are generated.

order feature map of the environment.  $D_{cls}(\cdot, \theta_m)$ ,  $D_{reg}(\cdot, \theta_m)$  and  $D_{var}(\cdot, \theta_m)$  indicate the classification decoder, regression decoder, and variance-inference decoder process over the parameter set  $\theta_m$ , respectively. Additionally, a *softplus* layer was added to the final layer of the variance-inference decoder to ensure that the output was strictly positive.

Evaluating uncertainty remains a challenging task due to the absence of ground-truth uncertainty labels [8]. Therefore, we adopt a direct modeling approach, wherein the object bounding boxes are assumed to follow a multivariate Gaussian distribution. The model is trained to learn the optimal parameters  $\hat{\theta}_m$  by minimizing the discrepancy between predicted objects and their ground-truth. Instead of predicting only deterministic bounding box attributes, the network estimates a set of parameters characterizing the full probability distribution for each object. The mean vector of the multivariate Gaussian represents the attributes of predicted object, while the covariance matrix  $\hat{\Sigma}_n$  quantifies the uncertainty associated with each attribute. To ensure both computational efficiency and numerical stability in estimating  $\hat{\Sigma}_n$ , we apply the Cholesky decomposition, which guarantees a symmetric positive-definite covariance matrix. This process is formally expressed in Equation (7).

$$P_{\hat{\theta}_m} = \frac{1}{\sqrt{2\pi|\hat{\Sigma}_n|}} \exp \left( -\frac{(\mathcal{B}_n - \hat{\mathcal{B}}_n)^T \hat{\Sigma}_n^{-1} (\mathcal{B}_n - \hat{\mathcal{B}}_n)}{2} \right) \quad (7)$$

where  $\hat{\theta}_m$  is the learnable parameters of  $m$ -th sub-network.  $n \in \{1, 2, \dots, \tilde{N}_m\}$  indicates the index of objects.  $|\cdot|$  indicates the determinant of matrix.  $\hat{\Sigma}_n^{-1}$  indicates the inverse of matrix

$\hat{\Sigma}_n$ . Note that we assume that each attribute of the bounding box is independent and  $\hat{\Sigma}_n$  is a diagonal matrix.

The ground truth of the bounding box can similarly be modeled as a multivariate Gaussian distribution, wherein  $\sigma \rightarrow 0$  effectively becomes a Dirac delta function [25]. It can be described as:

$$P_G = \delta(\hat{\mathcal{B}}_n - \mathcal{B}_n) \quad (8)$$

where  $P_G$  denotes the probability distribution of the ground truth.  $\delta(\cdot)$  denotes the Dirac delta function.

For each sub-network, we need to find the optimal parameter  $\theta_m$  that minimizes the discrepancy between  $P_{\hat{\theta}_m}$  and  $P_G$  over dataset  $D$ , as shown in Equation. (9).

$$\hat{\theta}_m = \operatorname{argmin} \left\{ \sum_{x \sim D} \mathbb{D}_{KL}(P_{\theta_m}(x), P_G(x)) \right\} \quad (9)$$

where  $\mathbb{D}_{KL}(\cdot, \cdot)$  indicates the Kullback-Leibler(KL) divergence between the two distributions.

We adopt the focal loss for the classification task and the softmax-based cross-entropy loss for orientation prediction [15], [28]. For bounding box regression and variance estimation, we adopt the KL divergence as the loss function, denoted

by  $\mathcal{L}_{KL}$ . The complete loss function is formulated as follows:

$$\mathcal{L} = \beta_{cls}\mathcal{L}_{cls} + \beta_{KL}\mathcal{L}_{KL} + \beta_{dir}\mathcal{L}_{dir} \quad (10)$$

$$\mathcal{L}_{KL} = \frac{1}{\hat{N}} \times \sum_{n=1}^{\hat{N}} \mathbb{D}_{KL}(P_{\hat{\theta}}(x) || P_G(x)) \quad (11)$$

$$\mathbb{D}_{KL}(P_{\hat{\theta}} || P_G) = \frac{(\mathcal{B}_n - \hat{\mathcal{B}}_n)^T \hat{\Sigma}_n^{-1} (\mathcal{B}_n - \hat{\mathcal{B}}_n) + \log |\hat{\Sigma}_n|}{2} \quad (12)$$

where  $\beta_{cls}$ ,  $\beta_{KL}$  and  $\beta_{dir}$  are the coefficients of the classification, regression, and direction loss, respectively. All the sub-networks share these coefficients.

During the training phase, the proposed method consists of two stages. First, the shared network is trained by setting the vectors  $R_0$  and  $S_0$  as all-one tensors set, resulting in the parameter set  $\theta_0$ . In the second stage, each sub-networks are trained based on the pretrained  $\theta_0$ , all network parameters are frozen except for  $R_m$  and  $S_m$ . The network is fine-tuned using different initialization parameters and the shuffled dataset to obtain a set of ensemble parameters  $\{\theta_1, \theta_2, \dots, \theta_M\}$ .

### C. Uncertainty-aware Fusion Strategy

It is well-known that neural networks with identical architectures but different parameter can yield diverse outputs even when provided with the same input. Consequently, naively aggregating these outputs to produce a final prediction can result in erroneous and redundant results, which degrade the overall detection accuracy and impose substantial computational burdens on downstream modules. Therefore, an effective fusion strategy is required to aggregate predictions from multiple networks in a manner that preserves reliability while minimizing noise and redundancy. In this paper, we employ an uncertainty-aware fusion strategy that consolidates and filters object detection results based on their associated uncertainties. Given  $M$  sets of predictions  $\{\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_M\}$ —each comprising category labels, bounding boxes, and variance estimates generated by the sub-networks described above—the final object detection outputs and corresponding uncertainty metrics are obtained through the following steps:

First, we iterate through the detection outputs. For each object category, the predictions are grouped into overlapping and non-overlapping sets based on the Intersection over Union (IoU) metric. If the IoU between predictions from different ensemble members exceeds a predefined threshold, they are classified as a group of overlapping results. In contrast, predictions that do not meet the threshold with any other result are treated as non-overlapping results. For the overlapping results, we treat the results as a uniformly weighted mixture strategy and combine the predictions, which further improves the detection accuracy. This is described by Equation. (13).

$$\bar{\mathcal{B}} = \hat{M}^{-1} \sum_{m=1}^{\hat{M}} \hat{\mathcal{B}}_m \quad (13)$$

where  $\hat{M}$  is the number of models in overlapping results.

Therefore, the variance of the detected object can be calculated as follows:

$$\bar{\Sigma} = \hat{M}^{-1} \sum_{m=1}^{\hat{M}} (\hat{\Sigma}_m + \hat{\mathcal{B}}_m^2) - \bar{\mathcal{B}}^2 \quad (14)$$

In the sections above, the variances of object attributes can be obtained. However, relying solely on the variance is insufficient for a comprehensive assessment of an object's uncertainty. Variance fails to account for false-positive predictions, and different object categories may exhibit varying sensitivities to the scale of attribute values. To address these limitations, we adopt the Uncertainty Index (UI), a novel metric designed to uniformly quantify uncertainty in 3D object detection results. This metric leverages the Gaussian error function—a well-established concept in probability theory and statistics—to evaluate the dispersion of predicted distributions. The Gaussian error function effectively measures how data points deviate from the mean, making it a suitable tool for capturing the uncertainty inherent in the predictions of 3D object detection models.

Additionally, for the uncertainty index of ensemble networks, the degree of consensus among multiple models regarding object detection serves as an important indicator of an object's existence reliability. When an object is consistently detected by a majority of networks, it implies higher confidence and should correspond to a lower uncertainty index. In contrast, if an object is detected by only a single network, the associated uncertainty should be higher, suggesting that the object may be a false positive. To accurately quantify this aspect of epistemic uncertainty, it is essential to incorporate the number of ensemble members that successfully detect a given object into the construction of the *UI*. To this end, we introduce the concept of hard voting into the uncertainty index formulation. This mechanism ensures that the final index reflects both the statistical dispersion and the consensus of the model, providing a more comprehensive measure of uncertainty. The formulation is described in Equation (15).

$$UI = H_{\hat{M}} \times \lambda \times \frac{2}{K \times \sqrt{\pi}} \sum_{k=1}^K \left( \int_0^{\sqrt{\hat{\Sigma}_{k,k}}} e^{-\eta^2} d\eta \right) \quad (15)$$

where *UI* denotes the uncertainty index of a prediction. *UI* is set within the range  $[0, 1]$ , a larger *UI* indicates higher uncertainty and lower confidence in the result; conversely, a smaller *UI* indicates lower uncertainty and higher confidence.  $\lambda$  is a proportionality coefficient that depends on the object category.  $H$  is a vector reflecting the impact of the number of models on the index. In this paper, we set  $H = \{1.2, 1.0, 0.8, 0.6, 0.4\}$ .

For the non-overlapping results, we compute *UI* for each object and establish a category-specific threshold. If the *UI* exceeds the threshold, the object is retained in the final result  $\bar{Y}$ ; otherwise, it is regarded as a false-positive detection. Finally, the proposed method outputs both the environmental perception results and their corresponding uncertainty estimates. These comprehensive results can be further utilized

TABLE I  
PERFORMANCE COMPARISON AMONG DIFFERENT METHODS ON KITTI 3-D VALIDATION SET

| Method          | uncertainty output | mAP            | Pedestrian     |                |                | Cyclist        |                |                | Car            |                |                |
|-----------------|--------------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
|                 |                    |                | Easy           | Mod.           | Hard           | Easy           | Mod.           | Hard           | Easy           | Mod.           | Hard           |
| PointPillars    | ✗                  | 63.9082        | 48.5506        | 44.6239        | <b>41.3760</b> | 81.7063        | 62.9372        | <b>60.6820</b> | 85.7048        | 76.1309        | 73.4627        |
| Shared Detector | ✓                  | 62.6148        | 46.8686        | 42.5368        | 38.2561        | 78.8958        | 60.9305        | 58.0010        | 87.4758        | 76.5609        | 74.0080        |
| <b>Ours</b>     | ✓                  | <b>64.9037</b> | <b>48.8068</b> | <b>44.9527</b> | 41.0497        | <b>83.2674</b> | <b>63.3034</b> | 60.3677        | <b>89.1844</b> | <b>78.1883</b> | <b>75.0131</b> |

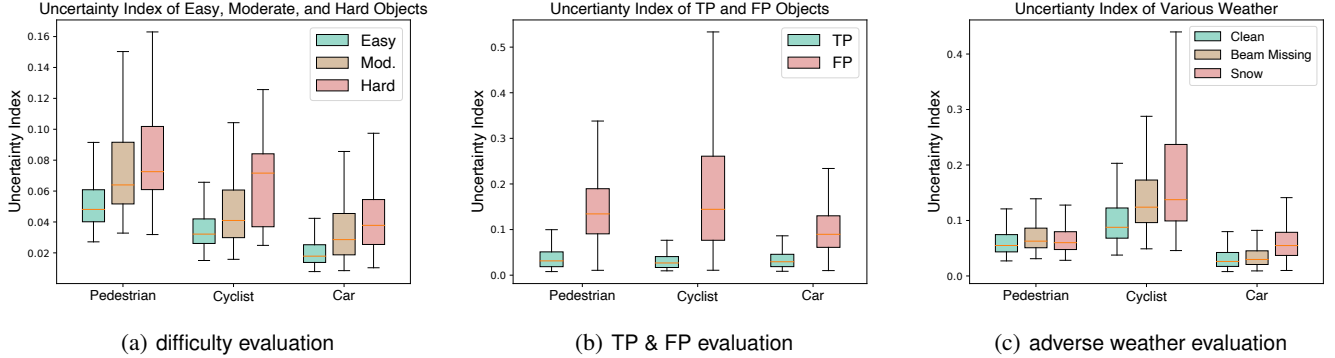


Fig. 3. Statistical analysis results of the Uncertainty Index. (a) illustrates the *UI* for easy, moderate, and hard detection difficulty levels. The results indicate that the perception model exhibits higher uncertainty when detecting challenging objects, such as smaller or more occluded targets like pedestrians and cyclists. For each object category, the differences in uncertainty across difficulty levels are statistically significant. (b) illustrates the *UI* for true-positive and false-positive results. The results indicate that the perception model exhibits higher uncertainty for false-positive results compared to true-positive results. (c) illustrates the *UI* under various weather conditions. The results indicate that the model exhibits higher uncertainty in adverse conditions, and there is a statistically significant difference between these conditions.

by the vehicle’s decision-making and planning modules to enhance autonomous driving safety and reliability.

#### IV. EXPERIMENT

In this study, all experiments were conducted using an Intel Core(R) i9–10920X CPU @ 3.50 GHz (Turbo 4.60 GHz), an NVIDIA GeForce(R) RTX 3090 24-GB GPU, under the Ubuntu 16.04 LTS operating system. All programming tasks were executed in Python 3.8, using PyTorch 1.8.1.

##### A. Implementation Details

In the implementation, we adopted fully BECL as both the encoder and decoder based on PointPillars [15]. Similar to [8], the number of ensemble members was set to  $M = 5$ . For the shared network, the feature vectors  $R_0$  and  $S_0$  were initialized as all-one tensors with their gradients frozen, and the network was trained for 150 epochs to obtain the shared parameters  $\theta_0$ . Then, based on the pre-trained shared parameters  $\theta_0$ , each ensemble member was fine-tuned by updating only its member-specific feature vectors  $R_m$  and  $S_m$  for an additional 50 epochs. The Adam optimizer was adopted and the learning rate was dynamically adjusted using the OneCycleLR policy across both phases. For the Uncertainty-aware Fusion Strategy, the IoU threshold was set to 0.5. The thresholds of the uncertainty index filter were 0.8 for pedestrians, 0.7 for cyclists, and 0.5 for cars. This differentiation optimizes the fusion process to enhance detection accuracy and reliability across various object types.

To validate the proposed method, experiments were conducted using the KITTI dataset, which is a widely recognized benchmark for autonomous driving [29]. KITTI consists of 7481 annotated training samples and 7518 testing samples. Each training sample includes comprehensive annotations. The dataset categorizes samples into three levels of difficulty—easy, moderate, and hard—based on criteria such as occlusion, visibility, and bounding box size. In addition, we utilized KITTI-C, a variant of the original KITTI dataset designed to evaluate the robustness under adverse conditions [30]. For the experiments, we further divided the dataset training samples into 3712 samples for training and 3769 samples for validation.

We conducted performance comparisons among several models. As shown in Table. I, PointPillars serves as our baseline model, which does not incorporate uncertainty quantification. The Shared Detector refers to the model constructed based solely on the shared parameters  $\theta_0$ , without applying ensemble integration. Experimental results demonstrate that the proposed method outperforms both the original baseline and the Shared Detector in terms of average precision, highlighting the effectiveness of our uncertainty-aware framework.

##### B. Uncertainty Quantification

Several experiments were conducted to validate the capacity of the uncertainty quantification method. Because of the absence of ground-truth uncertainty labels, directly assessing the proposed uncertainty index is not feasible. Therefore, a

series of statistical analyses were conducted on the proposed method across the dataset to evaluate its effectiveness.

Firstly, the *UI* across objects with varying detection difficulty levels was evaluated using the KITTI dataset. As shown in Fig. 3a, our method assesses the *UI* associated with objects of different difficulty. Experimental results indicate that objects located at greater distances, those occluded by surrounding things, or those corresponding to false-positive detections tend to exhibit higher *UI*. In contrast, objects that are closer to the ego vehicle and fully visible generally yield lower *UI*, indicating more confident predictions by the perception system. Specifically, for pedestrians, the median *UI* for moderately difficult objects was 32.95% higher than that for easy objects, while the median *UI* for hard objects was 50.78% higher than for easy ones. For cyclists, the median *UI* of moderately difficult cases exceeded that of easy cases by 27.46%, and hard cases exceeded easy cases by 122.83%. For cars, moderate-difficulty objects had a median *UI* 59.92% higher than that of easy objects, and hard objects exhibited a 111.59% increase over easy ones. In addition, the experimental results reveal that the *UI* associated with detecting larger objects, such as cars, is significantly lower than that for smaller objects, such as pedestrians. This observation highlights a fundamental limitation of the LiDAR-based object detection method. Furthermore, the Kruskal–Wallis (KW) test was conducted to statistically assess the differences in *UI* values across difficulty levels within the same category. The results indicate that for all categories, the differences in *UI* across difficulty levels are highly significant, with  $p$ -values less than 0.001. These findings support the effectiveness of the proposed method in quantifying uncertainty across varying levels of detection difficulty.

Secondly, we conducted experiments to evaluate the effectiveness of the proposed method in identifying false-positive detections. As illustrated in Fig. 3b, the results show that false-positive results exhibit significantly higher uncertainty compared to true-positive results. This suggests that the proposed approach can effectively distinguish false-positive detections, thereby substantially reducing the burden on the decision-making and planning modules of autonomous vehicles. Quantitatively, the median *UI* of false-positive results was 4.28, 5.38, and 3.04 times higher than that of true-positive results for pedestrians, cyclists, and cars, respectively. Furthermore, we conducted the Mann–Whitney U test—a non-parametric method suitable for comparing two independent groups without assuming normality—across all object categories. The results show that the *UI* of false-positive results is significantly higher than that of true-positive results ( $p < 0.001$ ), indicating that the proposed method effectively distinguishes between true and false detections based on uncertainty.

Finally, we evaluated the performance of the proposed method in quantifying uncertainty under adverse conditions using the KITTI-C dataset, which includes two representative scenarios: snowy weather and LiDAR beam-missing faults, as shown in Fig. 3c. Compared with normal conditions (i.e., clear weather and fully functional sensors), the *UI* showed a

noticeable increase under both types of disturbances. Specifically, under snowy conditions, the median *UI* increased by 9.32% for pedestrians, 56.79% for cyclists, and 109.57% for cars. In the presence of LiDAR beam-missing faults, the median *UI* increased by 14.28% for pedestrians, 41.39% for cyclists, and 13.71% for cars. To statistically validate these differences, we employed the Mann–Whitney U test, which revealed significant disparities between abnormal and normal conditions across all object categories ( $p < 0.001$ ). These results demonstrate the effectiveness of the proposed method in capturing and quantifying uncertainty under adverse environmental and sensor-degradation scenarios, thereby further confirming its potential to enhance perception reliability in autonomous driving applications. This allows downstream modules to respond more appropriately to uncertain perception outputs, improving overall system robustness.

## V. CONCLUSION

In this study, we propose a BatchEnsemble-based Network to quantify uncertainty for 3D object detection in point clouds. Specifically, a BatchEnsemble-based convolutional layer is designed, upon which a probabilistic object detection model is constructed. This model maintains a single set of shared parameters along with multiple sets of feature vectors, achieving ensemble-like performance in a memory-efficient manner. Furthermore, an uncertainty-aware fusion strategy is employed to integrate and filter the outputs based on the uncertainty index, producing more reliable perception results along with corresponding uncertainty estimates. We conduct extensive experiments on the KITTI and KITTI-C datasets. The results demonstrate that our method outperforms the original baseline in terms of detection accuracy while providing effective uncertainty quantification. For uncertainty quantification, our method yields significantly higher *UI* values for small, occluded, and distant objects, as well as for false-positive detections. In addition, the proposed approach yields higher *UI* values under adverse weather conditions and sensor degradation. The statistical significance of these results is validated through the Kruskal–Wallis test and the Mann–Whitney U test. These experimental findings confirm the effectiveness of the proposed method. The network architecture adopted in this study is relatively simple, and primarily aimed at validating the effectiveness of constructing convolutional neural networks based on BECL in uncertainty quantification. In future work, we plan to explore more comprehensive and high-performance models to achieve a better balance among uncertainty quantification, perception accuracy, and other key metrics.

## REFERENCES

- [1] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixão, F. Mutz, L. de Paula Veronese, T. Oliveira-Santos, and A. F. De Souza, “Self-driving cars: A survey,” *Expert Systems with Applications*, vol. 165, p. 113816, 2021.
- [2] H. Hu, Q. Wang, Z. Zhang, Z. Li, and Z. Gao, “Holistic transformer: A joint neural network for trajectory prediction and decision-making of autonomous vehicles,” *Pattern Recognition*, vol. 141, p. 109592, 2023.

- [3] H. Hu, T. Zhao, Q. Wang, F. Gao, L. He, and Z. Gao, "Monocular 3-d vehicle detection using a cascade network for autonomous driving," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–13, 2021.
- [4] L. Yang, B. Kang, Z. Huang, X. Xu, J. Feng, and H. Zhao, "Depth anything: Unleashing the power of large-scale unlabeled data," *arXiv preprint arXiv:2401.10891*, 2024.
- [5] W. He and Z. Jiang, "A survey on uncertainty quantification methods for deep neural networks: An uncertainty source perspective," *arXiv preprint arXiv:2302.13425*, 2023.
- [6] J. Liu, H. Wang, L. Peng, Z. Cao, D. Yang, and J. Li, "Pnnuad: Perception neural networks uncertainty aware decision-making for autonomous vehicle," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 24 355–24 368, 2022.
- [7] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059.
- [8] 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.
- [9] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural network," in *Proceedings of the 32nd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 1613–1622.
- [10] S. Su, Y. Li, S. He, S. Han, C. Feng, C. Ding, and F. Miao, "Uncertainty quantification of collaborative detection for self-driving," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5588–5594.
- [11] Y. Wen, D. Tran, and J. Ba, "Batchensemble: an alternative approach to efficient ensemble and lifelong learning," in *International Conference on Learning Representations*, 2020.
- [12] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, "Bevfusion: A simple and robust lidar-camera fusion framework," in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 10 421–10 434.
- [13] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.
- [14] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
- [15] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [16] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3d object detection and tracking," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 11 784–11 793.
- [17] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.
- [18] J. Li, C. Luo, and X. Yang, "Pillarnext: Rethinking network designs for 3d object detection in lidar point clouds," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.
- [19] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher *et al.*, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. Suppl 1, pp. 1513–1589, 2023.
- [20] D. Krueger, C.-W. Huang, R. Islam, R. Turner, A. Lacoste, and A. Courville, "Bayesian hypernetworks," *arXiv preprint arXiv:1710.04759*, 2017.
- [21] A. Mobiny, P. Yuan, S. K. Moulik, N. Garg, C. C. Wu, and H. Van Nguyen, "Dropconnect is effective in modeling uncertainty of bayesian deep networks," *Scientific reports*, vol. 11, no. 1, p. 5458, 2021.
- [22] A. Wu, S. Nowozin, E. Meeds, R. E. Turner, J. M. Hernandez-Lobato, and A. L. Gaunt, "Deterministic variational inference for robust bayesian neural networks," in *International Conference on Learning Representations*, 2019.
- [23] X. Tang, K. Yang, H. Wang, J. Wu, Y. Qin, W. Yu, and D. Cao, "Prediction-uncertainty-aware decision-making for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 4, pp. 849–862, 2022.
- [24] O. Laurent, A. Lafage, E. Tartaglione, G. Daniel, J. marc Martinez, A. Bursuc, and G. Franchi, "Packed ensembles for efficient uncertainty estimation," in *The Eleventh International Conference on Learning Representations*, 2023.
- [25] 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 (CVPR)*, June 2019.
- [26] B. Tang, Y. Zhong, U. Neumann, G. Wang, S. Chen, and Y. Zhang, "Collaborative uncertainty in multi-agent trajectory forecasting," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 6328–6340.
- [27] B. Tang, Y. Zhong, C. Xu, W.-T. Wu, U. Neumann, Y. Zhang, S. Chen, and Y. Wang, "Collaborative uncertainty benefits multi-agent multi-modal trajectory forecasting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 11, pp. 13 297–13 313, 2023.
- [28] 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 (ICCV)*, Oct 2017.
- [29] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [30] L. Kong, Y. Liu, X. Li, R. Chen, W. Zhang, J. Ren, L. Pan, K. Chen, and Z. Liu, "Robo3d: Towards robust and reliable 3d perception against corruptions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2023, pp. 19 994–20 006.



**DS**SPACE

<https://dspace.org/>

## **BatchEnsemble-based perception uncertainty quantification for autonomous vehicles**

**Li, Zhengyi; Hu, Hongyu; Xing, Yang; Lv, Chen**

**2025-11-18**

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/itsc60802.2025.11423601>

Li Z, Hu H, Xing Y, Lv C. (2025) BatchEnsemble-based perception uncertainty quantification for autonomous vehicles. In: Proceedings of the 2025 IEEE 28th International Conference on Intelligent Transportation Systems (ITSC), 18-21 Nov 2025, Gold Coast, Australia, pp. 122-129